Operator's Commands

Version 3 Release 5
Contents

Figures .................................... vii

Tables .................................... ix

Accessibility ................................. xi
Using assistive technologies .............. xi
Keyboard navigation of the user interface ... xi

Dotted decimal syntax diagrams .......... xiii

How to send your comments to IBM ....... xv
If you have a technical problem .......... xv

About This Book ................................ xvii
Who Should Use This Book ............... xvii
Where to Find More Information .......... xviii
The System Automation for z/OS Library ... xviii
Related Product Information ............... xviii
Summary of Changes for SC34-2720-00 .... xviii
New Information .............................. xix
Changed Information ....................... xix
Deleted Information ....................... xix

Part 1. Introduction ......................... 1

Chapter 1. Introduction ..................... 3
Overview of Commands ..................... 3
Understanding Terms ...................... 3
Resource ................................ 3
Format of Syntax Diagrams ............... 5

Part 2. SA z/OS System Operations
Commands ................................ 7

Chapter 2. System Operations
Commands ................................ 9
Using System Operations Commands ....... 9
General Information ....................... 9
Overview of Commands that Operate Sysplexwide 9
Additional Parameters for System Operations
  Commands ............................ 10
  Varying the Format of the Command Output . 13
ACF ..................................... 15
AOCHelp .................................. 22
AOCTrace .................................. 23
ASF ..................................... 28
ASFUSER .................................. 30
DISPACF .................................. 32
DISPAOPS .................................. 34
DISPAPG .................................. 36
DISPASF .................................. 38
DISPAUTO .................................. 40
DISPERRS .................................. 41
DISPEVT .................................. 43
DISPEVTS .................................. 44
DISPFLGS .................................. 47
DISPGW .................................. 50
DISPINFO .................................. 52
DISPMSG .................................. 54
DISPMTR .................................. 55
DISPSCDH .................................. 58
DISPSFLT .................................. 60
DISPSLT .................................. 63
DISPSYS .................................. 67
DISPTREE .................................. 69
DISPTRG .................................. 70
DRAINJES .................................. 72
EXPLAIN .................................. 73
INGAMS .................................. 74
INGAUTO .................................. 88
INGCF .................................... 92
INGCF DRAIN ................................ 99
INGCF ENABLE ................................ 103
INGCF MAINT ................................ 106
INGCF PATH ................................ 107
INGCF STRUCTURE .......................... 108
INGCFG ................................... 110
INGCFL ................................... 110
INGCICS ................................... 112
INGDB2 ................................... 116
INGDLA ................................... 119
INGEVENT .................................. 121
INGFILT ................................... 122
INGGROUP .................................. 129
INGHIST ................................... 137
INGHWSRV ................................ 139
INGIMS ................................... 140
INGINFO ................................... 146
INGLKUP ................................... 148
INGLKP ................................... 158
INGMDFY ................................... 163
INGMOVE .................................... 166
INGMSG .................................... 171
INGNTFY .................................... 174
INGPAC .................................... 177
INGPW .................................... 180
INGPLEX .................................... 184
INGPLEX CDS ................................ 189
INGPLEX SYStem ............................ 195
INGPLEX Console ........................... 197
INGPLEX IPL ................................ 200
INGPLEX SDUMP ............................ 201
INGPLEX SVCdump .......................... 204
INGPLEX SLIp ................................ 207
INGRELS .................................... 208
INGREQ .................................... 211
INGERPT .................................... 221
INGERUN .................................... 224

© Copyright IBM Corp. 1996, 2014
Figures

1. Resource Selection Panel ........ 4
2. Resource Selection Panel: Non-Sysplexwide 11
3. Resource Selection Panel 2: Sysplexwide 12
4. AOCHelp Command Dialog Panel .... 23
5. AOCTRAce Main Command Dialog Panel 26
6. AOCTRAce Command Dialog Panel for a Specific REXX Script .......... 27
7. Display of Automation Control File Settings for Subsystem (DISPACF SUBSYSTEM) .... 34
8. Automation Operators Panel ........ 36
9. DISPAPG Panel .................. 38
10. Display of Automation Status File Information for TSO (DISPASF TSO) ........ 40
11. DISPAuto Command Dialog Panel .... 41
12. DISPERRS Command Dialog Panel .... 42
13. DISPEVT Command Dialog Panel .... 44
14. DISPEVTS Command Dialog Panel .... 46
15. DISPEVTS Command Dialog Panel: Setting an Event .................. 47
16. DISPFLGS Command Dialog Panel .... 49
17. DISPGW Command Dialog Panel 1 .... 51
18. DISPINFO Command Dialog Panel .... 53
19. Authorized Message Receivers (DISPMGS) Panel .......................... 55
20. DISPMTTR Initial Command Dialog Panel 57
21. DISPMTTR Command Dialog Panel Showing Details for a Monitor .......... 58
22. DISPSCHD Command Dialog Panel .... 60
23. DISPSFLT Command Dialog Panel .... 62
24. Display of Status of All Automated Resources (DISPSTAT) ................. 66
25. DISPSYS Command Dialog Panel .... 68
26. DISPTREE Command Dialog Panel .... 70
27. DISPTRG Command Dialog Panel .... 71
28. EXPLAIN Command Dialog Panel .... 74
29. INGAMS Command Dialog Panel .... 79
30. INGAMS Command Dialog Panel to Control Secondary Automation Manager Functions . 81
31. INGAMS Command Dialog Panel to Show Details .......................... 83
32. INGAMS Command Dialog Panel to Refresh the Configuration ............... 85
33. INGAMS Command Dialog Panel to Manage Diagnostic Functions ............ 86
34. INGAMS Work Item Statistics Display .... 87
35. INGAMS Work Item History Display .................. 88
36. INGAUTO Command Dialog Panel ........ 92
37. INGCFC Selection Panel ........... 99
38. ENABLE Command Dialog Panel: After Populating .......................... 105
39. ENABLE Command Dialog Panel: After Issuing Command “INGCF E cf_name” ........ 106
40. MAINT Command Output ............. 107
41. PATH Command Dialog Panel .......... 107
42. STRUCTURE Command Dialog Panel .... 108
43. INGCICS REQ=CMD Command Dialog Panel 114
44. INGCICS REQ=BROADCAST Command Dialog Panel .................. 115
45. INGCICS REQ=INFO Command Dialog Panel .... 116
46. SA z/OS Discovery Library Adapter Report 121
47. INGFILT Command Dialog Panel .... 127
48. INGGROUP Command Dialog Panel .... 134
49. INGGROUP Command Dialog Selection Panel .......................... 135
50. INGGROUP Command Dialog Panel Showing Members’ Detail ............... 136
51. INGHIST Command Dialog Panel .... 139
52. INGIMS Command Dialog Panel .... 143
53. INGIMS REQ=BROADCAST Command Dialog Panel .................. 144
54. IMS Dependent Regions Panel ....... 146
55. INGINFO Command Dialog Panel .... 147
56. INGLIST Command Dialog Panel .... 153
57. INGLIST Command Dialog Panel To Update Status .......................... 157
58. INGLKUP Command Dialog Panel .... 162
59. INGLKUP Command Dialog Panel for the ANALYSE Option ............... 162
60. INGMDFY Command Dialog Panel .... 165
61. INGMOVE Command Dialog Panel .... 169
62. INGMOVE Confirmation Panel ....... 171
63. INGMSGS Command Dialog Panel .... 173
64. INGNTFY Command Dialog Panel .... 176
65. INGPAC Pacing Gate(s) Display Panel .... 179
66. INGPAC Pacing Gate Detail Display Panel 180
67. INGPLEX Selection Panel .......... 187
68. INGPLEX CDS Command Dialog Panel .... 190
69. Confirmation Panel for Switching from the Current Primary CDS to the Alternate CDS .... 191
70. INGPLEX CDS Command Dialog Panel after the Switch ............... 192
71. CFRM Couple Data Set Information Panel before Policy Switch ............ 193
72. Confirmation Panel for Policy Switch 193
73. Channel Path Information for CFRM Couple Data Sets ............... 194
74. INGPLEX SYSTEM Command Dialog Panel 1 .... 195
75. INGPLEX SYSTEM Command Dialog Panel 2 .... 197
76. INGPLEX CONS Command Dialog Panel .... 198
77. INGPLEX IPL Main Panel ............. 200
78. INGPLEX Dump Options Panel ........ 202
79. INGPLEX SDUMP Panel ....... 202
80. INGPLEX SDUMP Modification Panel .... 203
81. INGPLEX SVCDUMP Target System Selection Panel ............... 204
82. INGPLEX SVCDUMP Address Space Selection Panel ............... 205
83. INGPLEX SVCDUMP Address Space Detail Panel .... 205
84. INGPLEX SVCDUMP Dump Option Panel .... 206
85. INGPLEX SVC Line Mode Sample .... 207
86. INGPLEX SLIP Main Panel .......... 208
Tables

1. System Automation for z/OS Library xvii 5. Device and Path Status Before and After the
2. Overview of Commands 3 REMOVE SWITCH and RESTORE SWITCH
3. ISQVARS Keywords Sorted by Keyword
   Name . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ...
Accessibility

Publications for this product are offered in Adobe Portable Document Format (PDF) and should be compliant with accessibility standards. If you experience difficulties when using PDF files, please refer to the Knowledge Center at: [http://www-01.ibm.com/support/knowledgecenter/SSWRCJ_3.5.0/](http://www-01.ibm.com/support/knowledgecenter/SSWRCJ_3.5.0/)

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS® enable users to:

- Use assistive technologies such as screen readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using such products to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to z/OS TSO/E Primer, z/OS TSO/E User’s Guide, and z/OS ISPF User’s Guide Vol 1 for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.
Dotted decimal syntax diagrams

Syntax diagrams are provided in dotted decimal format for users accessing the Information Center using a screen reader. In dotted decimal format, each syntax element is written on a separate line. If two or more syntax elements are always present together (or always absent together), they can appear on the same line, because they can be considered as a single compound syntax element.

Each line starts with a dotted decimal number; for example, 3 or 3.1 or 3.1.1. To hear these numbers correctly, make sure that your screen reader is set to read out punctuation. All the syntax elements that have the same dotted decimal number (for example, all the syntax elements that have the number 3.1) are mutually exclusive alternatives. If you hear the lines 3.1 USERID and 3.1 SYSTEMID, you know that your syntax can include either USERID or SYSTEMID, but not both.

The dotted decimal numbering level denotes the level of nesting. For example, if a syntax element with dotted decimal number 3 is followed by a series of syntax elements with dotted decimal number 3.1, all the syntax elements numbered 3.1 are subordinate to the syntax element numbered 3.

Certain words and symbols are used next to the dotted decimal numbers to add information about the syntax elements. Occasionally, these words and symbols might occur at the beginning of the element itself. For ease of identification, if the word or symbol is a part of the syntax element, it is preceded by the backslash (\) character. The * symbol can be used next to a dotted decimal number to indicate that the syntax element repeats. For example, syntax element *FILE with dotted decimal number 3 is given the format 3 \* FILE. Format 3* FILE indicates that syntax element FILE repeats. Format 3* * FILE indicates that syntax element * FILE repeats.

Characters such as commas, which are used to separate a string of syntax elements, are shown in the syntax just before the items they separate. These characters can appear on the same line as each item, or on a separate line with the same dotted decimal number as the relevant items. The line can also show another symbol giving information about the syntax elements. For example, the lines 5.1*, 5.1 LASTRUN, and 5.1 DELETE mean that if you use more than one of the LASTRUN and DELETE syntax elements, the elements must be separated by a comma. If no separator is given, assume that you use a blank to separate each syntax element.

If a syntax element is preceded by the % symbol, this indicates a reference that is defined elsewhere. The string following the % symbol is the name of a syntax fragment rather than a literal. For example, the line 2.1 %OP1 means that you should refer to separate syntax fragment OP1.

The following words and symbols are used next to the dotted decimal numbers:

- ? means an optional syntax element. A dotted decimal number followed by the ? symbol indicates that all the syntax elements with a corresponding dotted decimal number, and any subordinate syntax elements, are optional. If there is only one syntax element with a dotted decimal number, the ? symbol is displayed on the same line as the syntax element, (for example 5? NOTIFY). If there is more than one syntax element with a dotted decimal number, the ? symbol is displayed on a line by itself, followed by the syntax elements that are
optional. For example, if you hear the lines 5 ?, 5 NOTIFY, and 5 UPDATE, you
know that syntax elements NOTIFY and UPDATE are optional; that is, you can
choose one or none of them. The ? symbol is equivalent to a bypass line in a
railroad diagram.

- ! means a default syntax element. A dotted decimal number followed by the !
symbol and a syntax element indicates that the syntax element is the default
option for all syntax elements that share the same dotted decimal number. Only
one of the syntax elements that share the same dotted decimal number can
specify a ! symbol. For example, if you hear the lines 2? FILE, 2.1! (KEEP), and
2.1 (DELETE), you know that (KEEP) is the default option for the FILE keyword.
In this example, if you include the FILE keyword but do not specify an option,
default option KEEP will be applied. A default option also applies to the next
higher dotted decimal number. In this example, if the FILE keyword is omitted,
default FILE(KEEP) is used. However, if you hear the lines 2? FILE, 2.1, 2.1.1!
(KEEP), and 2.1.1 (DELETE), the default option KEEP only applies to the next
higher dotted decimal number, 2.1 (which does not have an associated
keyword), and does not apply to 2? FILE. Nothing is used if the keyword FILE
is omitted.

- * means a syntax element that can be repeated 0 or more times. A dotted
decimal number followed by the * symbol indicates that this syntax element can
be used zero or more times; that is, it is optional and can be repeated. For
example, if you hear the line 5.1* data area, you know that you can include one
data area, more than one data area, or no data area. If you hear the lines 3*, 3
HOST, and 3 STATE, you know that you can include HOST, STATE, both
together, or nothing.

Notes:

1. If a dotted decimal number has an asterisk (*) next to it and there is only one
   item with that dotted decimal number, you can repeat that same item more
   than once.

2. If a dotted decimal number has an asterisk next to it and several items have
   that dotted decimal number, you can use more than one item from the list,
   but you cannot use the items more than once each. In the previous example,
   you could write HOST STATE, but you could not write HOST HOST.

3. The * symbol is equivalent to a loop-back line in a railroad syntax diagram.

- + means a syntax element that must be included one or more times. A dotted
decimal number followed by the + symbol indicates that this syntax element
must be included one or more times; that is, it must be included at least once
and can be repeated. For example, if you hear the line 6.1+ data area, you must
include at least one data area. If you hear the lines 2+, 2 HOST, and 2 STATE,
you know that you must include HOST, STATE, or both. Similar to the * symbol,
the + symbol can only repeat a particular item if it is the only item with that
dotted decimal number. The + symbol, like the * symbol, is equivalent to a
loop-back line in a railroad syntax diagram.
How to send your comments to IBM

We appreciate your input on this publication. Feel free to comment on the clarity, accuracy, and completeness of the information or give us any other feedback that you might have.

Use one of the following methods to send us your comments:
1. Send an email to s390id@de.ibm.com
2. Mail the comments to the following address:
   IBM Deutschland Research & Development GmbH
   Department 3282
   Schoenaicher Str. 220
   D-71032 Boeblingen
   Federal Republic of Germany
3. Fax the comments to us as follows:
   From Germany: 07031-16-3456
   From all other countries: +(49)-7031-16-3456

   Include the following information:
   • Your name and address
   • Your email address
   • Your telephone or fax number
   • The publication title and order number:
     IBM Tivoli System Automation for z/OS V3R50 Operator’s Commands
     SC34-2720-00
   • The topic and page number related to your comment
   • The text of your comment.

   When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

   IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you submit.

If you have a technical problem

Do not use the feedback methods listed above. Instead, do one of the following:
• Contact your IBM service representative
• Call IBM technical support
• Visit the IBM zSeries support web page at www.ibm.com/systems/z/support/.
About This Book

This document provides detailed information and reference material for operating IBM® Tivoli® System Automation for z/OS (SA z/OS).

Throughout this publication references to MVS™ refer either to MVS/ESA, or to the MVS element of z/OS.

Who Should Use This Book

This information is intended primarily for operators and system programmers. It may also be useful for others, for example, help desk personnel and customer engineers.

Where to Find More Information

The System Automation for z/OS Library

Table 1 shows the information units in the System Automation for z/OS library:

<table>
<thead>
<tr>
<th>Title</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM Tivoli System Automation for z/OS Planning and Installation</td>
<td>SC34-2716</td>
</tr>
<tr>
<td>IBM Tivoli System Automation for z/OS Customizing and Programming</td>
<td>SC34-2715</td>
</tr>
<tr>
<td>IBM Tivoli System Automation for z/OS Defining Automation Policy</td>
<td>SC34-2717</td>
</tr>
<tr>
<td>IBM Tivoli System Automation for z/OS User’s Guide</td>
<td>SC34-2718</td>
</tr>
<tr>
<td>IBM Tivoli System Automation for z/OS Messages and Codes</td>
<td>SC34-2719</td>
</tr>
<tr>
<td>IBM Tivoli System Automation for z/OS Operator’s Commands</td>
<td>SC34-2720</td>
</tr>
<tr>
<td>IBM Tivoli System Automation for z/OS Programmer’s Reference and Operator’s Guide</td>
<td>SC34-2748</td>
</tr>
<tr>
<td>IBM Tivoli System Automation for z/OS TWS Automation Programmer’s and Operator’s Reference Guide</td>
<td>SC34-2749</td>
</tr>
<tr>
<td>IBM Tivoli System Automation for z/OS End-to-End Automation Adapter</td>
<td>SC34-2750</td>
</tr>
<tr>
<td>IBM Tivoli System Automation for z/OS Monitoring Agent Configuration and User’s Guide</td>
<td>SC34-2751</td>
</tr>
</tbody>
</table>

SA z/OS Home Page:

For the latest news on SA z/OS, visit the SA z/OS home page at:

Related Product Information

For information that supports Tivoli System Automation for z/OS, visit the z/OS library in IBM Knowledge Center [www.ibm.com/support/knowledgecenter/SSLTBW/welcome].
Summary of Changes for SC34-2720-00

This document contains information previously presented in System Automation for z/OS V3R4.0 Operator's Commands, SC34-2649-00.

You may notice changes in the style and structure of some content in this document—for example, headings that use uppercase for the first letter of initial words only, and procedures that have a different look and format. The changes are ongoing improvements to the consistency and retrievability of information in our documents.

This document contains terminology, maintenance, and editorial changes.

New Information

The following information and commands have been introduced:

**System Operations**

**INGPAC**

The “INGPAC” on page 177 command is added for managing pacing gates.

**INGPW**

The “INGPW” on page 180 command is added for managing passwords.

**DISPINFO**

The UP Status Delay field is added to the DISPINFO command output. See “DISPINFO” on page 52.

**INGAMS**

The XSTATUS parameter is added to the INGAMS command to display information denoted by the PLEXID specification for XCF groups. See “INGAMS” on page 74.

**HMCSWITCH**

A new Ensemble command is added for switching between a Primary and an alternative Hardware Management Console (HMC). See “HMCSWITCH” on page 516.

**INGFILT/INGLIST**

The PGNAME parameter is added for INGFILT and INGLIST to display information about Pacing Gates for resources. See INGFILT and “INGLIST” on page 148.

**Processor Operations**

**ISQSNDH**

A new processor operations ISQSNDH command is added for sending high priority commands. See “ISQSNDH” on page 372.

**ISQVARS Keywords**

Two new keywords lparscp (LPAR scope) and pollopt (Path poll option) are added to the list of keywords available with the processor operations command “ISQVARS” on page 377.

**ISQVARS Keywords for SNMPv3**

Three new keywords snmpv3 (Use SNMPv3 protocol), v3pwd (SNMPv3 Password), and v3User (SNMPv3 Username) are added for the SNMPv3 settings when using the processor operations command “ISQVARS” on page 377.
**Changed Information**

Information has been updated for the following commands:

**System Operations**

**DISPAPG**

The **DISPAPG** command panel is updated.

**INGFILT/INGLIST**

The Application Type and SubType parameters are replaced by the Category and Subcategory parameters in **INGFILT** and **INGLIST** on page 148 to cover IBM-defined and user-defined application categories.

**INGMDFY**

The scenarios for using these commands are redefined. See **“INGMDFY”** on page 163.

**Processor Operations**

**ISQXDST**

The SA z/OS PATH details screen (ISQEPNVC) is updated. See “Display Path Details” on page 415 for new SNMPv3 security fields and parameters.

**Additional value ZASURE for ISQVARS variables**

Value ZASURE is added for variables `ostype` (Target operating system type) and `tmode` (Target system mode) in use with the **ISQVARS** on page 377 command.

**Deleted Information**

The following information has been deleted:

**UOW and INIT settings for the Override parameter in INGREQ/INGRUN**

The above settings are retired for the **INGREQ** on page 211 and **INGRUN** on page 224 system operations commands.
Part 1. Introduction

This part gives an overview of System Automation for z/OS commands — how to enter them, their format, and the various types of commands.
Chapter 1. Introduction

Overview of Commands

Table 2 gives a brief overview of the System Automation for z/OS commands. This overview describes the various types of commands, their functions and where they can be entered. For detailed information about how to issue commands, see IBM Tivoli System Automation for z/OS User’s Guide.

Table 2. Overview of Commands

<table>
<thead>
<tr>
<th>Type of command</th>
<th>Function</th>
<th>Where entered</th>
</tr>
</thead>
<tbody>
<tr>
<td>System operations commands</td>
<td>Control and maintain resources in the enterprise from a single point of control</td>
<td>NetView console, or NMC</td>
</tr>
<tr>
<td>I/O operations commands</td>
<td>Control input/output devices</td>
<td>TSO/ISPF, API, operator console</td>
</tr>
<tr>
<td>Processor operations commands</td>
<td>Common commands for automation</td>
<td>API, NetView console, or NMC</td>
</tr>
<tr>
<td></td>
<td>Note: Precede with ISQCCMD command</td>
<td></td>
</tr>
<tr>
<td>Control hardware processors</td>
<td></td>
<td>NetView console or NMC</td>
</tr>
<tr>
<td>Ensemble commands for System z® zEnterprise® BladeCenter® Extension (zBX) automation</td>
<td>API, NetView console Note: Precede with ISQECMD command</td>
<td></td>
</tr>
</tbody>
</table>

Understanding Terms

This section contains details of various terms that are used throughout this book.

Resource

In SA z/OS the term resource denotes any entity that can be automated. The automation manager administers resources. A resource is identified by its name. The resource name must be unique within the automation manager’s name space. The format of the resource name is name/type/system. The three components of the resource name can be the following:

Name  Specifies the name of the resource, for example the subsystem name for an application (APL) or the automation name for an application group (APG).

Type  Specifies the type of the resource. The type parameter is optional. It can have the following values:

- APL  Indicates that the resource is a subsystem.
- APG  Indicates that the resource is an application group.
- EVT  Indicates that the resource is an event.
- MTR  Indicates that the resource is a monitor.
- SVP  Indicates that the resource is a service period (schedule).
- SYG  Indicates the group of all resources of a system. SYG is created automatically. This is the application group that contains all resources of a particular system.
- SYS  Indicates that the resource is a system.
**Understanding Terms**

**System**
Specifies the system/image name where the resource is defined. The system parameter is not used for sysplex resources, for example, sysplex application groups, events or service periods.

If the resource that you specified is not unique within the domain of the automation manager, a selection panel is displayed where you can select what you want to work with, as shown in Figure 1.

![Figure 1. Resource Selection Panel](image)

**Specifying Resources**
There are various ways of specifying automation manager resources in System Automation for z/OS. The scope of the automation manager is a sysplex or SA z/OS subplex (that a group that shares the same XCF ID).

**Using Components of Resource Names:**
Components are the parts that make up the name of a resource using automation manager notation.

Here are a few examples of how to specify resource names using components:

- **TSO** All resources that have the name TSO.
- **TSO/APG** All resources that have the name TSO, and the type APG.
- **TSO/APL** All resources that have the name TSO, of the type APL, on all systems in the sysplex.

**Using Wildcards:**

You can also use an asterisk (*) to substitute one or more components of a fully-qualified resource name. Components are optional and may be replaced by an asterisk (*). For example, a specification for TSO expands with TSO/*. This allows you to specify a wide range of resource types quickly.

Here are a few examples of how to specify resources using wildcards:
Understanding Terms

*/APL/SYS1
All resources of type APL, on system SYS1.

TSO*/SYS1
All resources that have the name TSO, of any type, on system SYS1.

*/SYS/
All resources of type SYS, on any system in the sysplex.

*/*/
All resources of any type on any system in the sysplex.

You can specify wildcards for a component of the fully-qualified resource name as a leading or trailing character. The following shows a few examples:

TSO*/APL/KEY*
All resources starting with TSO, of type APL, on the systems whose names start with KEY.

*TSO/AP*/
All resources whose names end with TSO, of any type starting with AP, on any system in the sysplex.

You can use the percentage sign (%) as a placeholder for one character. This means that any character in that position of the resource name matches. The following shows a few examples:

%TSO*
All resources whose names contain TSO starting in character two.

TSO/S%S/*
All resources whose names start with TSO and where the first and last character of the type is S, on any system in the sysplex.

Format of Syntax Diagrams

The description of each command and routine includes the format of the command in a syntax diagram. The diagram shows the operands for the commands. Use blanks to separate the operands, unless otherwise stated or diagrammed.

To construct a command from the diagram, follow the diagram from left to right, choosing the path that suits your needs. Following is a sample syntax diagram that explains how to construct a command. This sample command is for illustration only.

Notes:
1 Start here. ➤➤ indicates the start of the diagram.
2 You can type ASAMPLE or abbreviate it to AS. The uppercase characters are the abbreviation. Operands on the main line are required.
3 Choose one of the options. The default is always above the main line. In this case, ALL is the default. If the option includes punctuation marks, include them too: =(), .
4 Choose E, Q, or neither. Operands below the main line are optional.

5 Repeat job_number any number of times. Variables are shown in italics. Replace them with a real name or value. Note that if you specify more than one job_number you must put them in brackets.

6 End here. ➔� indicates the end of the command.

If a command continues to the next line, you see ➔ and ➔. |and| indicates a fragment for a specific condition or option.

Examples:

====> asample none q DAF00821 DAF00832 ELD00824
====> as some DLR01445
This part describes SA z/OS system operations commands — their format, and specifics of how to use them.

Chapter 2. System Operations Commands

Using System Operations Commands

General Information

You can issue any system operations command by typing the command and its parameters on the command line of any NetView or system operations panel. You can also issue system operations commands by entering the command via NMC. In addition, System Automation for z/OS provides a menu of command dialogs that allows you to select a command dialog panel for a specific system operations command. For further information on how to issue system operations commands, see IBM Tivoli System Automation for z/OS User’s Guide.

Overview of Commands that Operate Sysplexwide

The following system operations commands operate sysplexwide (which means that they also operate across an SA z/OS subplex):

- DISPEVT/DISPEVTS
- DISPMT
- INGAMS
- INGCF/INGCFL
- INGCFG
- INGCICS
- INGEVENT
- INGHIST
- INGIMS
- INGINFO
- INGLIST
- INGMOVE
- INGTWS
- INGPLEX
- INGRELS
- INGREQ
- INGRUN
- INGSCHED
- INGSET
- INGSTR
- INGTRIG
- INGVOTE

You can issue sysplexwide commands from any system within the sysplex, regardless of where the resource resides.

If no target is specified, the command that was issued will find the affected resources in the sysplex.

Specifying the target system parameter is only required when routing the command from the focal-point system to a system in another sysplex or to a single remote system.

For further information about sysplexwide commands, see IBM Tivoli System Automation for z/OS User’s Guide.
Additional Parameters for System Operations Commands

The following parameters are available for a number of system operations commands.

**TARGET Parameter**

**Purpose**

The TARGET parameter lets you specify the system where the command is to be processed. You can direct the command to:

- A particular system in the sysplex, or enterprise
- A subset of systems in the sysplex, or enterprise
- A sysplex
- All systems currently active in the local sysplex

**Format**

```
Command name TARGET= Destination Dest_list *ALL
```

**Destination:**

```
-system_name
-domain_ID
-sysplex_name
```

**Dest_list:**

```
(Destination)
```

**Values**

- `system_name`
  Specifies the name of the z/OS system

- `domain_ID`
  Specifies the NetView domain identifier

- `sysplex_name`
  Specifies the name of the sysplex

- `*ALL`
  SA z/OS specifies that the command should be routed to all SA z/OS systems that are currently active in the local sysplex.

  Note that not all of the commands that support the TARGET parameter also support the *ALL value. Refer to the syntax diagrams to find out whether *ALL is supported.

**Processing**

The search sequence to locate the target is as follows:

1. System name within the local sysplex
2. Domain ID within the local sysplex
3. Local sysplex name

If it is a human operator, the search continues in the following sequence:

- Domain ID within the enterprise
- System name within the enterprise
- Sysplex name within the enterprise

If no value is specified, and the command does not refer to a defined resource, it is processed on the local system.

**Non-Sysplexwide Commands:**

If you do not specify the TARGET parameter, and the SA z/OS command refers to a defined resource, SA z/OS processing is as follows:

- If the resource is only defined on one active system, the command is routed to that system.
- When processing the command in line mode, SA z/OS checks whether the resource is defined on the local system. If it is, the command is processed locally. Otherwise, an error message is issued.
- If the resource is defined on more than one active system, and the command is not running on an autotask, SA z/OS displays a selection panel such as in Figure 2 where you can select the resource or resources that the command is to be routed to.

If you enter `dispacf mvesa` a panel similar to Figure 2 is displayed.

![Figure 2. Resource Selection Panel: Non-Sysplexwide](image)

If the target is within the local sysplex, the communication method is via XCF facilities. Otherwise, the NetView RMTCMD command is used.

**Sysplexwide Commands:**

Sysplexwide commands are processed locally.

If a target is specified and the resource is not defined on the specified system, SA z/OS issues an error message.
Additional Parameters for System Operations Commands

If you enter ingvote stdt000* a panel similar to Figure 3 is displayed.

<table>
<thead>
<tr>
<th>Command name</th>
<th>OUTMODE</th>
<th>TYPE</th>
<th>SYSTEM</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD000AN00</td>
<td>LINE</td>
<td>APL</td>
<td>ACAC</td>
<td>Parent for all STD* appl 05/16/00 06:25</td>
</tr>
<tr>
<td>STD000AN00</td>
<td>LINE</td>
<td>APL</td>
<td>ACAB</td>
<td>Parent for all STD* appl 05/16/00 06:25</td>
</tr>
<tr>
<td>STD000AN00</td>
<td>LINE</td>
<td>APL</td>
<td>ACC</td>
<td>Parent for all STD* appl 05/16/00 06:25</td>
</tr>
<tr>
<td>STD000AN00</td>
<td>LINE</td>
<td>APL</td>
<td>ACD</td>
<td>Parent for all STD* appl 05/16/00 06:25</td>
</tr>
<tr>
<td>STD000AN1A</td>
<td>LINE</td>
<td>APL</td>
<td>ACAC</td>
<td>Child of AN10 (child tree --- AN1B)</td>
</tr>
<tr>
<td>STD000AN1A</td>
<td>LINE</td>
<td>APL</td>
<td>ACAB</td>
<td>Child of AN10 (child tree --- AN1B)</td>
</tr>
<tr>
<td>STD000AN1A</td>
<td>LINE</td>
<td>APL</td>
<td>ACC</td>
<td>Child of AN10 (child tree --- AN1B)</td>
</tr>
<tr>
<td>STD000AN1A</td>
<td>LINE</td>
<td>APL</td>
<td>ACD</td>
<td>Child of AN10 (child tree --- AN1B)</td>
</tr>
<tr>
<td>STD000AN1B</td>
<td>LINE</td>
<td>APL</td>
<td>ACAC</td>
<td></td>
</tr>
<tr>
<td>STD000AN1B</td>
<td>LINE</td>
<td>APL</td>
<td>ACAB</td>
<td></td>
</tr>
<tr>
<td>STD000AN1B</td>
<td>LINE</td>
<td>APL</td>
<td>ACC</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3. Resource Selection Panel 2: Sysplexwide

OUTMODE Parameter

Purpose

The OUTMODE parameter lets you specify the output mode of a command.

Format

```
Command name OUTMODE=
```

Values

LINE

If you specify LINE, the output is displayed in line mode, independent of the task type. Further characteristics are:

- No color attributes are set for data that is shown in line mode.
- The sequence of the fields may be different in line mode than in fullscreen.
- Not all fields from the fullscreen display may be shown in line mode.
- Line mode output is shown in a multiline message.
- System operations commands can be issued within a NetView PIPE by using the OUTMODE=LINE parameter, unless noted otherwise in the command description.
- Line mode output is not processed by the NetView automation table and is not written to the netlog. To write output from a command such as DISPSTAT to the netlog, use a PIPE command, for example:
  ```
  PIPE NETV DISPSTAT OUTMODE=LINE | LOGTO NETLOG
  ```
- Line mode output cannot be processed by a TRAP and WAIT.
Additional Parameters for System Operations Commands

- System operations commands that support the OUTMODE=LINE option can be used in user-written command lists. Note, however, that the format of the output may change in later releases.
- If you work with OUTMODE=LINE no prompt panel is displayed.

**AUTO**
Specifications that when the task that the command list runs on is unattended, the output of the command is written to the NetView log. Otherwise the output is written to the console.

**NETLOG**
Specifies that the output of the command is written to the NetView log.

**Note:** If no value is specified, the decision whether to display the command output with a fullscreen panel or in line mode is based on the NetView task type that the command is running on.

**OUTDSN Parameter**

**Purpose**
This parameter lets you specify the name of the data set that is to contain the output of the command.

**Format**
```
Command name OUTDSN=dsname
```

**Values**

*dsname*
The data set name. You can specify a sequential data set or a member of a partitioned data set.

The record format should be variable-blocked and recommended is VB 1024.

**Restrictions**

The data set must already exist.

The OUTDSN parameter forces OUTMODE=LINE.

**Note:** If OUTDSN specifies a data set that is already allocated to NetView (for example DSI LIST), the exclusive ENQ that is issued when OUTDSN is used will not be removed until NetView is stopped. Thus, it is recommended that you do not use pre-allocated data sets for OUTDSN.

**Varying the Format of the Command Output**

Most of the commands in fullscreen mode support the SORT and FIND subcommands.

**Sorting a List**
The SORT subcommand lets you change the order that data is displayed in.

The syntax of the SORT subcommand is as follows:
Additional Parameters for System Operations Commands

You can specify the following:

**sort_order**

The sort order. It can be:

- **A** The list is displayed in ascending order. This is the default.
- **D** The list is displayed in descending order.

**coln** The column to be sorted on. You can specify more than one column, separated by a comma or a blank.

For example, specifying `SORT D,3 1 4` sorts the display in descending sequence and in the order column 3, 1, and 4.

For commands that support horizontal scrolling (for example, DISPSTAT or INGLIST), the column title (header) can be used instead of column number. The title can be abbreviated. First match is taken. For those commands you can also enter `SORT` without parameters which leads you to the column tailoring panel. There you can assign sort orders and sort keys to the columns. You can also hide columns or change the column sequence and whether a column is to be prefixed (show on every panel) or not. Refer to the online help on the column tailoring panel for further capabilities of `SORT`.

### Searching for Strings

The **FIND** subcommand searches the display data for a specified string. It can be abbreviated to `F`. The **RFIND** (repeat find) command is used to find the next occurrence of the string in the display data. It can be abbreviated to `RF`.

The syntax of the **FIND** subcommand is as follows:

`FIND search_string direction`

**direction**:

- **N**
- **P**

You can specify the following parameters with the **FIND** and **RFIND** commands:

**search_string**

The string that is to be searched for. The search starts at the first line shown in the display.

**direction**

The direction that is used for the search. It can be:
Purpose

The ACF command loads, displays, and modifies automation control file entries. You can use ACF to refresh data of a particular system if it does not affect automation manager configuration data.

For modification and display actions to work, the automation control file must be loaded into storage. Once loaded, the displays and modifications affect an in-storage version of the automation control file, allowing you to make temporary changes. To make permanent changes, change the automation policy using the customization dialogs, generate the automation control file member, then reload the new version using INGAMS. This ensures that the configuration matches the automation manager and the automation agents.

Recommendations:
1. Use the INGAMS command rather than the ACF command to load or refresh an automation control file. This ensures that the configuration matches the automation manager and the automation agents.
2. Changes to automation policy that are made using the SA z/OS command dialogs or the ACF command are temporary. They modify the current in-storage version of the automation control file directly. They do not modify the automation control file that is stored on disk. To change an automation policy setting permanently, make sure you also change the automation policy (using the customization dialogs) that is stored on disk.
3. If the customization dialogs are used to rebuild the automation control file on disk, the changed data on disk will replace the data in storage at ACF REFRESH.
4. Use scope-checking to limit operator use of ACF to load and display operations.

The following syntax diagrams show how to use the ACF command to perform the different functions ACF supports. Do not combine syntax from the separate diagrams in the same ACF call.

Note: The ACF command is freeform:
- Commas are optional
- More than one space can separate keywords
- Keywords can be specified in any sequence
- Any parameters specified must follow the keyword that they apply to

To replace or add information in the automation control file use the following syntax:
Syntax

```
>> ACF
   function options
   display request options
   delete request options
   add/replace request options
```

**function options:**

```
-COLD
-REFRESH
-ATLOAD
-STATUS
-CHECK
   TOKEN=current_token
   DSN=dataset_name
   TOKEN=token
   DSN=dataset_name
   TOKEN=*                      
   DSN=dataset_name
```

**display options:**

```
(1)
-REQ=DISP
   ENTRY=*                      
   ENTRY=entry
   TYPE=*                      
   TYPE=(type)
```

**delete options:**

```
-REQ=DEL
   ENTRY=entry
   TYPE=type
```

**add/replace options:**

```
-REQ=REPL
   ENTRY=entry
   TYPE=type
   parms=value
```

**Notes:**

1. The specification of either **REQ=DISP** or **ENTRY=entry** is required.

**Parameters**

**COLD**

Reloads the currently loaded automation control file data from disk. Automation is disabled while the automation control file load is performed. It deletes all global variables that were associated with the Entry/type pairs defined in the policy database. It also deletes all exceptional messages and disabled timers. Statistical data provided by the INGRPT command is also deleted.
**ACF**

**Note:** It is not recommended to use ACF COLD. Especially for cleanups, use the INGCLEAN command. For further details, refer to the INGCLEAN command reference in *IBM Tivoli System Automation for z/OS Programmer’s Reference*.

**REFRESH**
Updates the currently loaded automation control file data. Automation is not disabled while the automation control file load is performed. Only data that has been changed will be loaded, everything else remains the same as before. This is the safe way to update the automation control file. Refresh will not load data of a subsystem that is currently in the process of being started or stopped by SA z/OS. SA z/OS will automatically retry the refresh five minutes later for the data that cannot currently be processed.

**Note:** ACF REFRESH will not delete global variables that were associated with Entry/Type pairs, that were deleted from the policy database before the last build was done, if those Entry/Types were not associated with a subsystem, a monitor resource, or an application group. However, if a keyword/value is removed from the Entry/Type, this is a change to the Entry/Type, and consequently an ACF REFRESH will rebuild the Entry/Type with the result that the deleted keyword/value pair is removed from the global variables. All deleted Entry/Types and their associated global variables will be reset during the next SA z/OS NetView restart or during ACF COLD processing or by using the INGCLEAN command. For further details, refer to the INGCLEAN command reference in *IBM Tivoli System Automation for z/OS Programmer’s Reference*.

**ATLOAD**
Reloads the NetView automation tables that are specified in the System Info policy item and the message revision table, depending on the value of AOFSMARTMAT.

**STATUS**
Displays information about the automation control file that is currently in storage.

**CHECK**
Verifies the ACF for validity and tests the automation tables that are specified in the System Info policy and also the message revision table, depending on the value of AOFSMARTMAT.

**TOKEN**
The configuration token that the ACF should be validated against.

- **current_token**
  The token that is currently in use (displayed with ACF STATUS). This is the default.

- **token**
  The configuration token used as the reference.

- **Specifying an asterisk (*) means that the token validation should be omitted.**

**DSN**
The data set name that contains the ACF data.

- **current_dsn**
  The data set that is currently in use (displayed via ACF STATUS). This is the default.
dataset_name

The configuration data set name containing the ACF data.

REQ

The type of request for automation control file information the ACF command performs. This value can be one of the following:

DISP

Displays information in the automation control file. This value is the default if this parameter is not coded.

DEL

Deletes information in the automation control file. This value must be coded when using ACF to delete automation control file information.

REPL

Replaces or adds information in the automation control file. This value must be coded when using ACF to replace automation control file information. REPL adds the entry specified on the ENTRY parameter if the entry does not already exist in the automation control file.

REQ=REPL will update data in place. That is, only data that is to be replaced needs to be specified in the command. All other existing data will be retained.

ENTRY

The entry field of the automation control file. This value can be up to 32 characters long, without imbedded blanks, commas, or quotes.

If information in the automation control file is displayed (REQ=DISP), and no value is specified in the entry field, ENTRY=* is used.

TYPE

The type field in the automation control file. The following values can be specified:

* Specifying * returns all type fields that are associated with a given entry, for example, all SUBSYSTEM or NTFYOP entries. * is the default value when REQ=DISP (display). REQ=DISP supports the use of * as a wildcard character when specifying type names, with the following restrictions:
  • The wildcard character, *, must be the last character in the type name. If an asterisk appears in any other position in a type name then it will be treated as a literal. If an asterisk appears in any other position in a type name with an asterisk as the last character then no wildcard processing occurs and both asterisks are treated as literals.
  • If you update an entry, you must specify the ENTRY= operand without a wildcard.
  • If no matches are found, a final search is performed with a type name of DEFAULTS.

For other ACF request types (delete and replace), you must specify an actual type name.

type

The name of the type field. REQ=REPL requests allow you to enter only one type.

When ENTRY=SUBSYSTEM, type can be up to 11 bytes long. In all other cases, type can be up to 32 characters long, without imbedded blanks, commas, or quotes.

(type,type,...)

Multiple types may be specified for DISP and DEL requests. Type names should be enclosed in parentheses and separated by commas.
For REQ=DISP, only the first type name found is displayed. For REQ=DEL requests, all the type names will be deleted.

parms=value
The data associated with the specified ENTRY and TYPE fields. This field is valid only with the REQ=REPL option. Specify this field as the parameter value, an equal (=) sign, and the value, without any spaces in between; for example, AUTO=NO.

The value can be any character data. It can have imbedded quotes, commas, and blanks, provided that single quotes or parentheses frame the value.

SA z/OS defines several ENTRY, TYPE, and parms=value fields. A parms=value example is the JOB=jobname parameter in the SUBSYSTEM automation control file entry.

Note: REQ=REPL will update data in place. That is, only modified data is updated. All data will be retained.

Restrictions and Limitations
- If you use the COLD parameter, automation is disabled while the ACF LOAD is in progress. If you use the REFRESH parameter, automation continues.
- An automation control file cannot be loaded by an operator using the ACF command if the automation flag for major resource MVSESA is set to 'N'. If the automation flag for the MVSESA.RELOAD.CONFIRM minor resource is set to 'Y', the operator can reload the ACF.
- ACF must comply with the automation manager configuration, otherwise message AOF618I is issued.
- The ACF command should be used with care to change or delete automation policy settings. All changes are only temporary changes and may get lost during the next configuration refresh. The changes are definitely lost during an ACF COLD or after processing the INGCLEAN command.
- The number of entries in the automation control file is limited only by the amount of storage in the SA z/OS address space or region. If you have a very large configuration, you may have to increase the REGION size in the SA z/OS procedure.
- Not all data can be changed using ACF REQ=REPL, for example, service periods, events, triggers, and dependencies cannot be changed. ACF REQ=DEL and ACF REQ=REPL cannot be used to add or delete an application (subsystem), an application class, an application group or a monitor resource. It is NOT recommended to change data of existing applications, application groups and monitor resources via ACF REQ=REPL to avoid data mismatch between automation agent and automation manager. Use the customization dialogs to define these items and/or change those data.
- Changes to System Defaults and Application Defaults policy items are not propagated to the instances that have inherited this data.

Security Considerations

The ACF command supports resource level security. If turned on, the following profiles in class SYSAUTO are checked:
ACF

<table>
<thead>
<tr>
<th>Profile</th>
<th>Authority</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGT.sysplex\xcfgrp.RES.rename.restype[\ressys]</td>
<td>UPDATE</td>
<td>When function option is REFRESH or ATLOAD</td>
</tr>
<tr>
<td></td>
<td>CONTROL</td>
<td>When function option is COLD and for delete (REQ=DEL) or add/replace requests (REQ=REPL)</td>
</tr>
</tbody>
</table>

Usage other than listed in the table does not cause a resource level security check. For further details, refer to *IBM Tivoli System Automation for z/OS Planning and Installation*.

**Return Codes**

- **0**: Function completed successfully.
- **1**: Control file inactive.
- **3**: ACF token mismatch.
- **4**: Invalid parameters were used in the call.
- **12**: No valid ACF member available.
- **20**: ACF reload not allowed by automation flag settings.
- **24**: Resource level security check failed.
- **100**: Other error.

**Usage**

- ACF performs two flag checks when the automation control file is reloaded. The checks are of the Automation flags for MVSESA.RELOAD.CONFIRM and MVSESA.RELOAD.ACTION. The action flag is checked only if the reload is to continue.

**MVSESA.RELOAD.CONFIRM**

This lets you:
- Turn the flag off and thus disable an automation control file reload by an operator.
- Tell SA z/OS to ask for confirmation by setting the AOFCONFIRM TGLOBAL to YES in a flag exit.
- Use a flag exit to perform additional checks and processing before deciding to either cancel or proceed with the reload.

If the flag is found to be off, the operator cannot reload the automation control file.

**Note:** You need to be aware of this if you turn the MVSESA automation flag off.

If the flag is on, the AOFCONFIRM task global is checked. If it is set to YES, SA z/OS will post a message requesting confirmation of the reload request. If AOFCONFIRM is null, SA z/OS will proceed with the reload.

**MVSESA.RELOAD.ACTION**

This provides an opportunity for you to do your own processing once SA z/OS is committed to reloading or restoring the automation control file. You can use a flag exit to call your own code. Your exit should return 0, indicating that the flag is turned on. If you return a nonzero return code, subsequent exits defined for the flag will not be invoked.
Note that there is no way of preventing the reload from an exit on this flag. See IBM Tivoli System Automation for z/OS Defining Automation Policy and IBM Tivoli System Automation for z/OS Customizing and Programming for further information.

- When you use ACF REQ=DISP to request a certain ENTRY value with one or more specific TYPE values, ACF searches for those types in the order specified in the command. When the first match is found, the information is returned to the requester as a multiline message. If there are no matches, it performs a final search with a type of DEFAULTS for that ENTRY value. If there is still no match, a message is returned to the requester. If the type DEFAULTS is found, that information is returned to the requester.

- If, when ACF is used to display an automation control file entry, a specific TYPE is found, it is treated as a complete entry. Only that specific entry is displayed.

Messages

The following lists the messages that are issued during the operation of ACF.

For the load function (COLD/REFRESH):

AOF042I MEMBER ACFZ999 NOT FOUND
AOF100I 16:05:09 : 'ACF REFRESH' COMMAND ISSUED
AOF618I NO VALID ACF FOUND FOR sysname - detail description
AOF782I AUTOMATION CONTROL FILE PROCESSING COMPLETED

For the status function (STATUS):

AOF005I MEMBER ACFZ992 CURRENTLY BEING USED FOR THE CONTROL FILE
AOF006I BUILT BY OPER1 ON 04/29/12 AT 11:43:03
AOF006I REFRESHED BY OPER1 ON 04/30/12 AT 08:52:20
AOF006I CONFIGURATION TOKEN = 2012049090119000C6F7A2084
AOF006I CONFIGURATION DATASET = OPER1.USER.V34.ACF
AOF002I END OF MULTI-LINE MESSAGE GROUP

For the delete and replace function (REQ=DEL/REPL):

AOF001I REQUEST REPL SUCCESSFUL FOR JES2-$HASP098

Note: In a display where the type_name is asterisk (*), multiple sets of AOF112I and AOF113I messages may be displayed. When the type is omitted or specified as asterisk (*), the DESIRED TYPE is not displayed in the AOF112I message.

For the display function (REQ=DISP):

AOF041I UNABLE TO FIND entry type
AOF111I AUTOMATION CONFIGURATION DISPLAY - ENTRY= entry_name
AOF112I ACTIVE TYPE= act_type, DESIRED TYPE= desired_type ...
AOF113I DATA IS data=value
AOF002I END OF MULTILINE MESSAGE GROUP

For example, the following may occur:

AOF111I AUTOMATION CONFIGURATION DISPLAY - ENTRY= NTFYOP
AOF112I ACTIVE TYPE= NETOP1
AOF113I DATA IS OPER='OPER 1'
AOF113I DATA IS CLASS=(10,40)
AOF112I ACTIVE TYPE= NETOP2
AOF113I DATA IS CLASS=(10)
AOF002I END OF MULTILINE MESSAGE GROUP

Note: Use of the replace parameter (REPL) adds an entry if none exists, resulting in a successful message.
ACF

Generic error messages that can occur:
AOF013I  SPECIFIED OPERAND operand INVALID FOR PARAMETER parameter.
AOF025I  SYNTAX ERROR

Examples

The ACF command to display the Start automation flag for the CICST subsystem
is:
ACF REQ=DISP,ENTRY=START,TYPE=CICST

The response is:
AOF111I AUTOMATION CONFIGURATION DISPLAY - ENTRY= START
AOF112I  ACTIVE TYPE= CICST , DESIRED TYPE= CICST
AOF113I  DATA IS AUTO=Y
AOF113I  DATA IS NOAUTO=(TUESDAY,10:00,12:00)
AOF002I  END OF MULTILINE MESSAGE GROUP

In this example, a Start automation flag exists for the CICST subsystem. The
operator or automation procedure processes the command to display the entry, and
the associated response is returned as a multiline message.

Use the following automation procedure to update ACF data for an entry. It allows
you to modify the automation agent configuration data without affecting other
automation agents or the automation manager.

``` /* ************************************************** **
** Function: **
**  - Read ACF Fragment **
**  - Modify ACF entries **
** **************************************************/

'PIPE (NAME ACFREPL)',
'QSAM (DSN) -dataset-', /* read ACF fragment */
'! NLOC 1.1 /*/', /* skip comments */
'! COLLECT', /* collect to multiline */
'! NETV ACF REQ=REPL', /* call ACF command */
'! CONS' /* issue msgs to console */
```

AOCHELP

Purpose

AOCHELP displays information about SA z/OS commands, and enables you to
execute them.

Syntax

``` AOCHELP commandname ```

Parameters

`commandname`

Specifies the SA z/OS system operations command that you require
information about. The result is the same as if you had specified help
`commandname`.
Restrictions and Limitations

- Some commands may not be executed from the AOCHELP command dialog. Attempts to execute these commands produce SA z/OS message AOF129I.
- Help is available only in fullscreen mode.
- The AOCHELP command can only be used when SA z/OS has initialized.

Usage

If you type aochelp by itself, you see a list of commands to choose from, as shown in Figure 4. From this panel you can ask for help on a command by entering its item number.

Examples

To get help, specify the associated number of the command.

![Figure 4. AOCHELP Command Dialog Panel](image)

To execute the command, specify the number associated with the command followed by an x, for example, 9x.

AOCTRACE

Purpose

The AOCTRACE command turns the SA z/OS system operations debugging feature on or off. AOCTRACE allows you to activate:

- Debugging globally
- Debugging for specific REXX script
- Debugging at a certain level of detail
- Tracing for a particular message
- REXX trace for a specific REXX script
AOCTRACE

Syntax

AOCTRACE

- OFF
- ON

MSG/id

- OFF
- ON

rexx_exec

- OFF
- ON

level Trace

TARGET=

system_name

domain_ID

OUTMODE=

LINE

AUTO

NETLOG

Trace:

trace routine_name

Parameters

MSG/id

The message that is being traced.

rexx_exec

The name of the REXX script that you want to debug.

ON

Turns the debugging feature on.

If you do not specify a REXX script or message, AOCTRACE activates a global debugging flag. This causes each SA z/OS system operations REXX script that is invoked to write a record (AOF700I) to the netlog showing the parameters that it was invoked with.

If you do specify a REXX script or message, only that REXX script or message writes messages to the netlog. The netlog shows when and where the REXX script was called, and what parameters were given.

If you specify a REXX script or message but not a trace, an intermediate level of debugging is activated for the specified REXX script or message. This causes the REXX script to issue a number of messages describing its actions. Note, however, that not all REXX scripts support this facility.

OFF

Turns the debugging feature off.

If a REXX script is specified, this option deactivates both the intermediate and REXX tracing for the specified REXX script.

If a message is specified, this option deactivates message tracing for the specified message.

If a REXX script or message is not specified, this option disables the global trace.

trace

The type of REXX trace that you want for the specified REXX script. Valid values are:
AOCTRACE

A  All
R  Results
I  Intermediates
C  Commands
E  Errors
F  Failures
L  Labels
O  Off
N  Normal

routine_name
The REXX script that is being traced.

level
Specifies the debug level that allows you to limit the amount of debug information that is written to the netlog. The meaning of the level depends on the particular REXX script.

NORM
This is the default. Control flow and important data area is written to the netlog.

MAX
Extensive logging of control flow and data areas and structures.

/AM
Indicates that data received from the automation manager is not traced.

/SLAVE
Indicates that data received from the command slave is not traced.

TARGET
For information on the TARGET parameter, see “TARGET Parameter” on page 10.

OUTMODE
For information on the OUTMODE parameter, see “OUTMODE Parameter” on page 12.

Restrictions and Limitations

The trace operand can only be used on interpreted REXX scripts.

Using the trace on certain REXX scripts can cause execution problems, particularly with REXX scripts that trap messages.

The REXX trace setting of S (Scan) is not available.

Message tracing only works from the command line.

Note: The debugging facility is not active if REXX traces A, R, or I are used.

Examples

If you enter aoctrace on you see a message indicating that the debugging facility has been enabled. While the debugging facility is enabled, message AOF700I is written to the netlog for each procedure being processed.
If no parameter is specified, Figure 5 is displayed.

The panel shows all the REXX scripts (that is, clists) and messages that have debug mode or REXX tracing turned on.

The first entry is the *GLOBAL* entry. This entry is always shown. It shows whether the global debug option is on or off. You cannot modify the setting with a command code. To set the global debugging option specify

- `AOCTRACE ON`
- `AOCTRACE OFF`

*MSG* in the CLIST column indicates a message trace entry. The message ID is shown in the Subroutines/Messages being traced column. As with the Global entry (*GLOBAL*), message trace entries cannot be modified on this panel: you must use the AOCTRACE MSG/id,ON|OFF command entered at the command line to do so.

The following information is shown:

- The CLIST field shows the name of the REXX script.
- The System field shows the name of the system where debugging or tracing is active.
- The Dbg field indicates whether or not debug mode is on. Y indicates that debug mode is turned on.
- The Level field indicates the debug level.
- The T (Trace) field shows the Trace option specified for the REXX script.
- The Subroutines/Messages being traced field shows the list of subroutines that are being traced.

The Cmd field lets you specify the command codes shown on the panel. Type the appropriate letter next to the resource name and press Enter. The following command codes are available:

- **A** Allows you to add debug or trace settings for another REXX script. The panel shown in Figure 6 on page 27 is displayed where you can specify the REXX script name and optionally overtype the other settings.
C Allows you to modify the debug or trace settings of the selected REXX script. The panel shown in Figure 6 is displayed showing the current settings. Here you can overtype the appropriate values.

D Turns off the debug option for the selected REXX script.

R Turns off the debug and trace option for the selected REXX script.

T Turns off the trace option for the selected REXX script.

Use the PF9 key to refresh the data displayed.

Use the PF10 key to turn off the debug and trace option for ALL displayed REXX scripts.

The ADD command can also be entered at the command line to define the debug or trace settings for a REXX script.

Figure 6 is displayed when the command code A or C is specified on the panel in Figure 5 on page 26, or a REXX script name is specified as the only parameter of the AOCTRACE command. You can use this panel to enable and disable the automation debugging facility for a given REXX script.

The following fields can be entered:

Clist name
Specify the name of the clist (that is, REXX script) When turned ON, the REXX script will write debug information to the NetView log. Note that if you have already specified a REXX script with the AOCTRACE command, or entered the command code C next to a REXX script on the main AOCTRACE panel, this field will contain that REXX script name and cannot be changed.

direct mode
Specify ON to turn on the REXX script's debug mode.

debug detail level (NORM, MAX, /AM, /SLAVE)

REXX trace option (A, R, I, C, E, F, L, O N)

Subroutines to be traced: (+ for all)
- or -

Message id ________________________________

Command ===>
PF1=Help PF2=End PF3=Return PF6=Roll PF12=Retrieve

Figure 6. AOCTRACE Command Dialog Panel for a Specific REXX Script

The following fields can be entered:

Clist name
Specify the name of the clist (that is, REXX script) When turned ON, the REXX script will write debug information to the NetView log. Note that if you have already specified a REXX script with the AOCTRACE command, or entered the command code C next to a REXX script on the main AOCTRACE panel, this field will contain that REXX script name and cannot be changed.

debug mode
Specify ON to turn on the REXX script's debug mode.
debug detail level
Specify the debug level for the REXX script. The meaning of the debug level varies from REXX script to REXX script. Standard values are:

NORM
This is the default. Control flow and important data area is written to the netlog.

MAX
Extensive logging of control flow and data areas and structures.

/AM
Indicates that data received from the automation manager is not traced.

/SLAVE
Indicates that data received from the command slave is not traced.

REXX trace option
Specify the trace setting for the REXX script. It can be one of the following:
A  All
R  Results
I  Intermediates
C  Commands
E  Errors
F  Failures
L  Labels
O  Off
N  Normal
_  Default

For more information about trace settings, see TRACE in z/OS TSO/E REXX Reference.

Notes:
1. The REXX script debug setting need not be ON to use REXX script trace.
2. Specifying REXX trace option A, I, or R makes individual REXX script debugging ineffective.

Subroutines to be traced
Specify the list of subroutines to be traced. By default, only the REXX script main routine will be traced. You can specify more than one subroutine name separated by a blank character. Specify "*" to trace all subroutines.

Message id
Specify the message identifier to be traced.

ASF

Purpose
The ASF command is a file manager command that displays records from the automation status file. The automation status file records are maintained in a VSAM data set. ASF interfaces with the VSAM file to obtain control information that is vital to SA z/OS, such as:

• Automation status
• Time and date information for specific events within the lifecycle of an application
Whether an error threshold has been exceeded
Time and date information for error conditions

You can also use the ASF command to reset occurrences of threshold errors by updating records in the automation status file.

**Syntax**

To display single or multiple records use the following syntax:

```
ASF ID=resource
```

To reset occurrences of threshold errors use the following syntax:

```
ASF REQ=REPL ID=resource ERRORDT=''
```

**Parameters**

**REQ**

The type of request for automation status file information that the ASF command performs. This value may be:

**DISP**

Displays information in the automation status file. This value is the default if the REQ parameter is not coded.

**REPL**

Resets a record in the automation status file.

**ID**

The resource ID that is the key to the automation status file record. This ID is the application name for application records. This value can be from 1 through 64 characters long.

**FROM**

The resource ID that is the starting key when displaying multiple automation status file records. This resource is the application name for application records. This value can be from 1 through 64 characters long.

**TO**

The resource ID that is the ending key when displaying multiple automation status file records. If not specified, the value defaults to the FROM parameter value. This resource is the application name for application records. This value can be from 1 through 64 characters long.

If a resource ID is specified, the records that are to be displayed are resolved alphabetically.

**ERRORDT**

Specify `ERRORDT='' THRESHLD=''` with `REQ=REPL` to reset occurrences of threshold errors for a resource.

**THRESHLD**

Specify `ERRORDT='' THRESHLD=''` with `REQ=REPL` to reset occurrences of threshold errors for a resource.
Usage
To modify automation status file fields that are reserved for your data, use the ASFUSER command. See "ASFUSER" for details.

Error conditions
To display status information about subsystem JES2, enter:
ASF REQ=DISP ID=JES2

To reset the occurrences of threshold errors for the application APPC, enter:
ASF ID=APPC REQ=REPL ERRORDT='' THRSHLD=''
ASFUSER

Parameters

REQ
The type of request for automation status file record information the ASFUSER command performs. This value may be one of the following:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISP</td>
<td>Displays a record in the automation status file. This value is the default if the REQ parameter is not coded.</td>
</tr>
<tr>
<td>DEL</td>
<td>Deletes a record in the automation status file.</td>
</tr>
<tr>
<td>REPL</td>
<td>Replaces or adds a record in the automation status file.</td>
</tr>
</tbody>
</table>

If this parameter is specified, other parameters that describe the data to be displayed, deleted, or updated must be specified.

FROM
The resource ID that is the starting key when displaying multiple automation status file records. This value can be from 1 through 64 characters long.

TO
The resource ID that is the ending key when displaying multiple records. If not specified, the value defaults to the same key as the FROM parameter. This value can be from 1 through 64 characters long.

ID
The resource ID that is the key to the automation status file record. This ID is the application name for application records. This ID can be from 1 through 64 characters long.

USER1=\textit{data}\textendash USER40=\textit{data}
These parameters specify data that is stored in each of the 40 fields in the automation status file that are reserved for your information. All these parameters are optional. The specified data can be 1 through 20 characters long. These parameters are only used with ASFUSER replace requests (REQ=REPL). The data must be enclosed in parenthesis or quotes if it contains blanks or a comma.

Restrictions and Limitations
None.

Usage
If your automation procedure performs a replace or update function, only those fields that need replacing must be specified. No change occurs to other automation status file fields.

Examples

Example 1
This example shows a command to create an automation status file record for a resource with a resource ID of DASD.

The ASFUSER command to create the record is:

\texttt{ASFUSER REQ=REPL ID=DASD USER1=3390 USER2=SITE1}
**ASFUSER**

The response to the ASFUSER command is:
AOF001I REQUEST "REPLACE" WAS SUCCESSFUL FOR "DASD"

**Example 2**

This example shows a command to display an automation status file record for DASD.

The ASFUSER command to display the record is:
ASFUSER ID=DASD

The response to the ASFUSER command is:
AOF150I STATISTICS DISPLAY REQUESTED FOR DASD THRU DASD
AOF151I ID=DASD , TYPE= N/A , STATUS= N/A
AOF152I LAST UPDATED BY OPERATOR OPER1
AOF158I USER1 = 3390
AOF158I USER2 = SITE1
AOF002I END OF MULTILINE MESSAGE GROUP

**Example 3**

This example shows a command to update the DASD automation status file record to add a new DASD device type.

The ASFUSER command to add a new field to the existing record is:
ASFUSER REQ=REPL,id=DASD,USER3=3990

The response to the ASFUSER command is:
AOF001I REQUEST "REPLACE" WAS SUCCESSFUL FOR "DASD"

**Note:** Other values in record DASD remain as they were before the ASFUSER command was issued.

---

**DISPACF**

**Purpose**

The DISPACF command displays resource information and automation policy settings for a specific entry or entry-type pair in the automation control file.

**Syntax**

```
>>-DISPACF- entry_name [entry_type] [TARGET= Destination | Dest_list | ALL] [OUTMODE= LINE | AUTO | NETLOG]
```
**Dispacf**

**Destination:**

- `system_name`
- `domain_ID`
- `sysplex_name`

**Dest_list:**

```
| Destination |
```

**Parameters**

`entry_name`

Specifies the name of the entry in the automation control file.

If you specify `*`, all entry names are displayed.

`entry_type`

Specifies the type within the entry in the automation control file. If you specify a particular type-name, only that type-name (and its data fields) for the entry-name is displayed.

If you specify `*`, all of the type-names and their data fields for the entry-name are displayed. This is the default. Wildcard is supported.

`TARGET`

For information on the TARGET parameter, see “[TARGET Parameter” on page 10](#).

`OUTMODE`

For information on the OUTMODE parameter, see “[OUTMODE Parameter” on page 12](#).

**Restrictions and Limitations**

The DISPACF command can only be used when SA z/OS is initialized.

**Examples**

*If you enter `dispacf` subsystem, you will see a fullscreen display similar to Figure 7 on page 34*
This command displays information for all types of the SUBSYSTEM entry, because you accepted the default TYPE=*

If you type dispacf subsystem jes2, you see only information about the JES2 type.

**DISPAOPS**

**Purpose**

The DISPAOPS command displays a list of automation operators that are currently active.

**Syntax**

```
>>DISPAOPS TARGET= Destination Dest_list OUTMODE= OUTMODE=
```

Figure 7. Display of Automation Control File Settings for Subsystem (DISPACF SUBSYSTEM)

This command displays information for all types of the SUBSYSTEM entry, because you accepted the default TYPE=*

If you type dispacf subsystem jes2, you see only information about the JES2 type.
DISPAOPS

Destination:

- system_name
- domain_ID
- sysplex_name

Dest_list:

- (Destination)

Parameters

TARGET
For information on the TARGET parameter, see "TARGET Parameter" on page 10.

OUTMODE
For information on the OUTMODE parameter, see "OUTMODE Parameter" on page 12.

Restrictions and Limitations
The DISPAOPS command can only be used when SA z/OS is initialized.

Examples
Type dispaops on any command line and press the Enter key. You will see a panel similar to Figure 8 on page 36.
DISPAOPS

The System field shows the name of the system where the automated function is defined.

The Automated Function field shows the name of the automated function that is used in SA z/OS automation procedures.

The Primary field shows the NetView automation operator ID assigned to this automated function.

The Status field shows the current status of the primary automation operator.

The Secondary field shows the Backup NetView automation operator ID assigned to this automated function.

The Status field shows the current status of the backup automation operator.

The primary and backup NetView automation operator IDs are assigned to the automated function in the command dialogs.

DISPAPG

Purpose

The DISPAPG command displays detailed information about a specified application group that is of interest from the automation agent's point of view, such as:

- Automation agent-related information
- Automation manager-related information
- The message history for captured messages that are associated with the group
### Syntax

```plaintext
>>DISPAPG applgroup

TARGET=system_name
domain_ID
sysplex_name

OUTDSN=dsname
-OUTMODE=LINE
AUTO
NETLOG
```

### Parameters

**applgroup**

The name of the application group you want information about. If you do not specify a application group, a menu is displayed where you can specify the application group name.

**TARGET**

For information on the TARGET parameter, see "TARGET Parameter" on page 10.

**OUTDSN**

For information on the OUTDSN parameter, see "OUTDSN Parameter" on page 13.

**OUTMODE**

For information on the OUTMODE parameter, see "OUTMODE Parameter" on page 12.

### Restrictions and Limitations

The DISPAPG command can only be used when SA z/OS is initialized.

### Examples

If you enter dispapg followed by the name of an application group at a command line you will see a panel similar to Figure 9 on page 38.
You can use the Resource and Target fields to specify the name of a different application group or you can press PF9 to refresh the information about the current application group.

You can use the PF4 key to invoke the INGINFO command to get details about the application group from the automation manager's point of view.

**DISPASF**

**Purpose**

DISPASF displays the information contained in the automation status file. This file holds information that is to be retained across SA z/OS automation agent sessions.

**Syntax**

```
>>DISPASF [identifier] [TARGET=Destination Dest_list ALL] [OUTMODE=LINE AUTO NETLOG]
```

**Figure 9. DISPAPG Panel**

The panel shows the details of an application group, including:

- **Resource**: MOVSYSTM/APG/KEY1
- **Description**: System move pref 700/500/300 MOVSYS1/2/3
- **Nature**: MOVE
- **Behaviour**: ACTIVE
- **Desired Available**: ALWAYS
- **Monitor for IPL complete**: YES
- **Prepare Move**: YES
- **Move mode**: PARALLEL
- **Status determination**: CSONLY
- **Inform list**: SDF
- **Availability target**: 1
- **Satisfactory target**: 1

The command line at the bottom of the panel shows the available PF keys for navigation and command execution.
DISPASF

Destination:

- system_name
- domain_ID
- sysplex_name

Dest_list:

- Destination

Parameters

identifier
  Specifies the name of an object in the status file. Such an object can be, for example, a subsystem. If you do not specify an identifier, a panel is displayed where you can specify an object.

TARGET
  For information on the TARGET parameter, see "TARGET Parameter" on page 10.

OUTMODE
  For information on the OUTMODE parameter, see "OUTMODE Parameter" on page 12.

Restrictions and Limitations

The DISPASF command can only be used when SA z/OS is initialized.

Examples

If you enter dispasf tso, you see a panel similar to Figure 10 on page 40.
DISPAUTO

Purpose

DISPAUTO provides a menu where you can initiate command dialogs that display information about your automation.

Syntax

```
--- DISPAUTO
```

Parameters

Selection

This is one of the commands that is listed for the DISPAUTO command.

Additional parameters

This indicates additional parameters that can be specified with the selected command.

Note: All keywords can be abbreviated to the minimum unique abbreviation. Additional parameters are passed ‘as is’ to the appropriate command routine.

Restrictions and Limitations

The DISPAUTO command can only be used when SA z/OS is initialized.

Examples

If you enter dispauto on an operator station task (OST) the panel in Figure 11 on page 41 is displayed.
After the command selection number, you can specify appropriate parameters for the selected commands, for example, 10 TARGET=xxx, where xxx is the domain ID, or the system name of a system in the sysplex.

**DISPERRS**

**Purpose**

The DISPERRS command displays information about resources that errors have been recorded for in the status file.

**Syntax**

```
   DISPERRS [filter] TARGET= [Dest_list ALL]
   OUTMODE= [LINE AUTO NETLOG]
   Destination:
   - system_name
   - domain_ID
   - sysplex_name
```

Figure 11. DISPAUTO Command Dialog Panel

After the command selection number, you can specify appropriate parameters for the selected commands, for example, 10 TARGET=xxx, where xxx is the domain ID, or the system name of a system in the sysplex.
**DISPERRS**

**Dest_list:**

- **Destination**

**Parameters**

**filter**

This is the filter to be applied for the output. Only resource names that match the filter criteria are displayed.

The filter consists of one or more qualifiers separated by a period. A wildcard, `*`, is supported at the beginning or the end of each qualifier.

**TARGET**

For information on the TARGET parameter, see "TARGET Parameter" on page 10.

**OUTMODE**

For information on the OUTMODE parameter, see "OUTMODE Parameter" on page 12.

**Restrictions and Limitations**

The DISPERRS command can only be used when SA z/OS is initialized.

**Examples**

If you enter `disperrs` you see a panel similar to Figure 12.

![Figure 12. DISPERRS Command Dialog Panel](image-url)

- The **System** field shows the name of the system where the resource is defined.
- The **Resource** field shows the name of the resource.
- The **Type** field shows the type of resource.
DISPERRS

- The *Thrs* field shows the type of threshold, if any, that was exceeded when the
  last error was recorded. This is either CRIT (critical), FREQ (frequent), or INFR
  (infrequent).
- The *No* field shows the sequence number assigned to the error.
- The *Date* and *Time* fields show the date and time that the error occurred.

DISPEVT

**Purpose**

The DISPEVT command displays relationships between an event and the related
resources. DISPEVT shows all the resources that use the event.

**Syntax**

```
DISPEVT event
```

**Parameters**

- **event**
  Specifies the name of the event.

- **TARGET**
  For information on the TARGET parameter, see “TARGET Parameter” on page
  10. Note that if you specify a sysplex name as a target, this is translated to the
  next system that is available in the sysplex and the DISPEVT command is
  processed there.

- **OUTMODE**
  For information on the OUTMODE parameter, see “OUTMODE Parameter” on
  page 12.

- **WAIT**
  Specifies the number of seconds to wait before reporting that a timeout
  occurred if the automation manager does not provide the requested data. The
  maximum time interval is 999 seconds.

  If omitted, the time interval is 30 seconds.

**Restrictions and Limitations**

The DISPEVT command can only be used when SA z/OS is initialized.

**Usage**

The DISPEVT command operates sysplexwide. For an overview see “Overview of
Commands that Operate Sysplexwide” on page 9.

If you specify OUTMODE=LINE, specifying *event* becomes mandatory.
DISPEVT

Examples

If you enter disp EVT b00s100, Figure 13 is displayed.

If you enter disp EVT b00s100, Figure 13 is displayed.

- The **Event** field shows the name of the event. You can overwrite the event to initiate a new display.
- The **Target** field shows the name of the system where the command is executed. You can overwrite the system name to initiate a new display.
- The **Description** field shows the description of the event.
- The **Unset condition** field shows the status that an application must have to reset an event.
- The **Resource** field shows the name of the resource where the event is defined.
- The **Type** field shows the type of the resource.
- The **System** field shows the name of the system where the resource is defined.
- The **Trigger** field shows the trigger that the resource is linked to.
- The **Status** field shows the status of the event for that particular resource. It can contain the following values:
  - **SET** The event is set. It is shown in white.
  - **UNSET** The event is unset. It is shown in red.
  
  If the status of the event is neither SET nor UNSET, the event is in an unknown state. This is treated as UNSET.

DISPEVTS

Purpose

DISPEVTS displays all events defined in a sysplex. From the fullscreen panel you can set or reset an event or display the related resources of an event.
DISPEVTS

Syntax

```
DISPEVTS
   TARGET=system_name
   domain_ID
   sysplex_name
   OUTMODE=LINE
   AUTO
   NETLOG
   WAIT=nnn
```

Parameters

**TARGET**
For information on the TARGET parameter, see "TARGET Parameter" on page 10. Note that if you specify a sysplex name as a target, this is translated to the next system that is available in the sysplex and the DISPEVTS command is processed there.

**OUTMODE**
For information on the OUTMODE parameter, see "OUTMODE Parameter" on page 12.

**WAIT**
Specifies the number of seconds to wait before reporting that a timeout occurred if the automation manager does not provide the requested data. The maximum time interval is 999 seconds.

If omitted, the time interval is 30 seconds.

Restrictions and Limitations

The DISPEVTS command can only be used when SA z/OS is initialized.

Examples

If you enter dispevts, a panel that shows all events defined in the sysplex is displayed, as shown in Figure 14 on page 46.
The **Cmd** field lets you specify the command codes shown on the panel. Type the appropriate letter next to the resource name and press Enter. The following command codes are available:

- **D** Shows the related resources.
- **S** Sets or resets the event.

**Event** field shows the name of the event.
**Unset** field shows the status that an application must have to reset an event. The following values can occur:

- **START**
  - Unsets the event when the resource is being started.

- **UP**
  - Unsets the event when the resource becomes available.

- **DOWN**
  - Unsets the event when the resource becomes unavailable.

If you do not specify anything the event will not be unset.

**Description** field shows the description of the event.

Type **S** next to an event and press Enter. The panel shown in Figure 15 on page 47 is displayed.
The Event field shows the name of the event.

The Description field shows the description of the event.

The Unset condition field shows the status that an application must have to reset an event.

The Function field shows the action to be performed. It can be overwritten.

The Resource field shows the name of the application that is the only affected application when the command is executed. It can be overwritten.

The Target field shows the name of the target system where the processing takes place. It can be overwritten. You only need to specify the name of the target system if you want to address a remote sysplex.

**DISPFLGS**

**Purpose**

The DISPFLGS command shows all resources that have explicit agent automation flags defined for them, what those flags are, and what the effective automation flags resulting from them are.

**Syntax**

```
DISPFLGS [filter] [TARGET=Destination] [Dest_list] [ALL] [OUTMODE=LINE] [AUTO] [NETLOG]
```
DISPFLGS

Destination:
- system_name
- domain_ID
- sysplex_name

Dest_list:

Parameters

filter
The filter to be applied for the output. Only resource names that match the filter criteria are displayed. Wildcard is supported. The filter consists of the major name and, optionally, of a minor name separated by a dot, for example, CICS*.TRANS*, or *.*CEMT. Specify * to view major resources. To view minor resources specify xyz.*.

TARGET
For information on the TARGET parameter, see “TARGET Parameter” on page 10.

OUTMODE
For information on the OUTMODE parameter, see “OUTMODE Parameter” on page 12.

Restrictions and Limitations

The DISPFLGS command can only be used when SA z/OS is initialized.

Usage

Subsystems that are not shown have no explicit automation flags set and have effective flags defined in the SUBSYSTEM entry.

Examples

If you enter dispflgs within SA z/OS you see a panel similar to Figure 16 on page 49.
The following command codes are available:

A  Lets you define automation flags for a resource using the same flag settings as the selected resource. On the next panel displayed you can specify the resource name and optionally overtype the flag settings.

C  Lets you modify the automation flags of the selected resource. The next panel shows the current flag settings. You can overtype the flag values.

R  Resets the automation flags to the values specified in the automation control file (ACF).

S  Shows the scheduled override settings of the automation flags for the selected resource.

- The **Resource** field shows the name of the resource. Names shown indented one character to the right represent minor resources, for example, MVSESA.DUMP as shown below. If the minor resources MVSESA.DUMP.ONE, MVSESA.DUMP.TWO, and MVSESA.DUMP.TWO.THREE existed, the Resource column would look like the example below:

```
MVSESA
DUMP
ONE
TWO
THREE
```

Major resources correspond to subsystems and minor resources correspond to specific situations or resources within a major resource. By default, automation flags for minor resources are derived from their major resources.

The entries are sorted alphabetically.

- The **Actual** flag settings are the flags that have been set for the resource.
- The **Effective** flag settings are the flags that are in effect for the resource.

The columns are headed with an abbreviation of the name of the automation flag.

The value of each flag is:
- There is no explicit setting.
- The flag is turned off.
E A user exit is invoked whenever SA z/OS checks the flag.
Y The flag is turned on.
L The resulting commands or replies, or both, of an automated action are
written to the netlog only but not issued when the event occurs.
? There is an error.

See IBM Tivoli System Automation for z/OS Customizing and Programming for further
information on automation flags and minor resources.

### DISPGW

**Purpose**

DISPGW displays gateway definitions, statuses, and connectivity information.

**Syntax**

```plaintext
>>DISPGW
TARGET=system_name
   domain_ID
   sysplex_name
OUTDSN=dsname
OUTMODE=LINE
   AUTO
   NETLOG
```

**Parameters**

**TARGET**

For information on the TARGET parameter, see "TARGET Parameter" on page 10.

**OUTMODE**

For information on the OUTMODE parameter, see "OUTMODE Parameter" on page 12.

**OUTDSN**

For information on the OUTDSN parameter, see "OUTDSN Parameter" on page 13.

**Restrictions and Limitations**

The DISPGW command can only be used when SA z/OS is initialized.

**Usage**

There are related panels that display gateway information. Use PF11 and PF10 to
scroll through them.

**Examples**

If you enter `dispgw` you see a panel similar to Figure 17 on page 51.
The amount of data depends on the screen size. Press PF10/11 to scroll horizontally.

- The **Domain** field shows the domain ID of the system that a gateway has been defined to, known as the *gateway domain*.
- The **SDF Root** field shows the system name of the gateway domain.
- The **Status** field shows the current status of the gateway.

The following are statuses that may occur for gateways:

**ACTIVE**
- The connection was established.

**BAD DOMAIN**
- The specified domain is unknown. Message: DSI031I.

**INACTIVE**
- The connection cannot be established. One of the following messages was issued: DSI008I, DSI015A, DSI027I.

**INV OPID**
- The specified operator ID is invalid. Messages: DSI021A, DSI077A, DSI400A.

**NOT STARTD**
- The domain has not been started by gateway initialization.

**SESSFAIL**
- The attempt to start a session with the domain ended with a VTAM® request failure. See DSI046I in the netlog for details.

- The **Comm** field shows the communication method that is used for the outbound connection with the gateway domain:
  - *XCF*  
    - XCF is used as the transport mechanism. This is the case when the gateway domain resides in the same logical sysplex.
  - *IP*  
    - The transport mechanism is via RMTCMD using IP.
  - *SNA*  
    - The transport mechanism is via RMTCMD using SNA.
  - *RPC*  
    - The remote procedure call technique is used as the transport mechanism.

**Note:** The RMTCMD method is derived from the RMTSYN entries found in CNMSTYLE. If you use RMTALIAS to describe more than one transport method, then you can influence which is selected as the RMTSYN entries are sorted in ascending order.

- The **Release level** field shows the SA z/OS release level of the gateway domain.

![Figure 17. DISPGW Command Dialog Panel 1](image-url)
The **In/Outbound Status** field shows the status of the inbound and outbound NetView sessions from this domain to the gateway domain. If this domain communicates with the gateway through another domain, the status is **NO DIRECT**.

The **Last out/in** field shows the first 8 bytes of the last gateway request from the gateway domain to this domain, and the last gateway request from this domain to the gateway domain. It can be any of the following: CHECK, CONNECT, DISCONN, or FPRESET.

The **System** field shows the name of the system.

The **SMF ID** field shows the SMF ID assigned to the system.

The **SA Plexname** field shows the SA z/OS SYSPLEX name. In most cases the name is identical to the physical SYSPLEX name.

The **XCF Group** field shows XCF group name the system belongs to.

The **Net Id** field shows the Networking ID.

The **SYSPLEX** field shows the physical sysplex name.

The **PrimaryFP** field shows the name of the primary focal point defined for the domain.

The **BackupFP** field shows the name of the backup focal point defined for the domain.

The **Description** field shows a description of the gateway domain, obtained from the automation control file.

### DISPINFO

**Purpose**

DISPINFO displays detailed information about a specified subsystem.

If the subsystem is a CICS® or IMS™ subsystem that is being controlled by CICS Automation or IMS Automation, a PF10 option is available to give access to CICS or IMS related information. This PF key is not available if the subsystem is down. See “Examples” on page 53 for more details.

**Syntax**

```
ISPINFO subsystem

TARGET subsystem

OUTDSN subsystem

OUTMODE subsystem

AUTO subsystem

NETLOG subsystem
```

**Parameters**

**subsystem**

The name of the subsystem you want information about. If you do not specify a subsystem, a menu is displayed where you can specify the subsystem.

**TARGET**

For information on the TARGET parameter, see “TARGET Parameter” on page 10.

**OUTDSN**

For information on the OUTDSN parameter, see “OUTDSN Parameter” on page 13.
OUTMODE

For information on the OUTMODE parameter, see “OUTMODE Parameter” on page 12.

Restrictions and Limitations

The DISPINFO command can only be used when SA z/OS is initialized.

Examples

Figure 18. DISPINFO Command Dialog Panel

For the subsystem and the system fields, you can override the values and refresh the display. You can also scroll forward and backward.
DISPINFO

For further details about the fields on the panel, see the customization dialogs and
IBM Tivoli System Automation for z/OS Defining Automation Policy.

When obtaining automation flag settings, no user exits are invoked.

To display the default settings use the subsystem name DEFAULTS or
SUBSYSTEM.

PF4 lets you toggle to the automation manager command INGINFO.

If PF10 is shown the DISPINFO command invokes the INGCICS or INGIMS
commands. For more information see “INGCICS” on page 112 or “INGIMS” on
page 140.

DISPMSGS

Purpose

The DISPMSGS command displays a list of automation operators that receive
automated messages.

Syntax

```
   DISPATCH SYSTEM system_name
   DISPATCH DOMAIN domain_ID
   DISPATCH SYSLX sysplex_name
   OUTMODE LINE AUTO NETLOG
```

Parameters

**TARGET**

For information on the TARGET parameter, see “TARGET Parameter” on page
10.

**OUTMODE**

For information on the OUTMODE parameter, see “OUTMODE Parameter” on
page 12.

Restrictions and Limitations

The DISPMSGS command can only be used when SA z/OS is initialized.

Examples

Enter dispmsgs on a command line to display the Authorized Message Receivers
panel, as shown in Figure 19 on page 55.
The information contained on the panel is:

- The **Message** field shows the message or message prefix.
- The **Primary Receivers** field shows the automation operators, identified by their NetView IDs. The Primary Receivers column lists automation operators that can receive the messages listed beside their names. These messages go to the first automation operator listed in the Primary Receivers column that is active.
- The **Secondary Receivers** field shows the Alternate automation operators, identified by their NetView IDs. Secondary Receivers receive copies of the messages listed beside their names.

### DISPMTR

#### Purpose

The DISPMTR command displays and allows you to manage monitors that you have defined using the customization dialogs for your system.

#### Syntax

```
DISPMTR monitor
    REQ=DETAIL
    RESOURCE=resname
```

Figure 19. Authorized Message Receivers (DISPMSGS) Panel
Parameters

\textit{monitor} \\
The name of the monitor resource. More than one name can be specified. \newline
Wildcard is supported, for example, SAP*.

\textit{resource} \\
The resource name that the associated monitor, or monitors, should be 
displayed for. If more than one name is specified, they must be enclosed in 
parentheses. Wildcard is supported.

\textit{REQ=DETAIL} \\
Displays detailed information for the specified monitor.

\textbf{Note:} Specification of the full monitor resource name is necessary if it is a 
remote monitor. Also, when using this option only one monitor can be 
specified.

\textbf{TARGET} \\
For information on the \textbf{TARGET} parameter, see \textit{“TARGET Parameter” on page 10.}

\textbf{OUTDSN} \\
For information on the \textbf{OUTDSN} parameter, see \textit{“OUTDSN Parameter” on page 13.}

\textbf{OUTMODE} \\
For information on the \textbf{OUTMODE} parameter, see \textit{“OUTMODE Parameter” on page 12.}

\textbf{WAIT} \\
Specifies whether or not to wait until the request is complete. The default is 
\textbf{YES}.

\textit{nnn} is the number of seconds to wait before giving up and reporting that a 
timeout has occurred. The maximum time interval is 999 seconds.

Restrictions and Limitations

The \textit{DISPMTR} command can only be used when \textit{SA z/OS} is initialized.

Examples

If you enter \textit{dispmtr} a panel similar to that shown in \textit{Figure 20 on page 57} is 
displayed.
The amount of data depends on the screen size. Press PF10/11 to scroll horizontally.

- The **Cmd** field lets you specify command codes. To use one of the command codes shown, type the appropriate letter next to the resource name and press Enter. The following command codes are available:
  
  **A**  If the resource is active, this lets you reset the health status of the resource being monitored to NORMAL.
  
  If the resource is inactive, it sets the status of the monitor resource to SOFTDOWN.
  
  **B**  Lets you start the monitor resource (that is, make it available).
  
  **C**  Lets you stop the monitor (that is, make it unavailable).
  
  **D**  Allows you to view definition details about the monitor and history data.
  
  **F**  Invokes the INGINFO command to show details about the monitor.
  
  **I**  Invokes the INGSCHED command to show details about the overrides that exist for the monitor.

- The **Monitor** column shows the name of the monitor.
- The **System** column shows the name of the system where the monitor runs.
- The **Status** column shows the status of the monitor. It can have one of the following values:
  
  **ACTIVE**  The monitor is running.

  **INACTIVE**  The monitor is not running.

  **FAILED**  The monitor has failed. Recovery may be in progress. No acceptable health status is provided.
BROKEN

Both the monitor and recovery failed. This is a permanent condition. The monitor will not be re-invoked.

- The **Last monitored** column shows when the resource was last monitored.
- The **Status message** column shows message that is associated with the status. If the message has been truncated it ends with ..., use command code D to view full details about the message.
- The **Description** column shows descriptive information about the monitor resource.
- The **Monitored Object** column shows the name of the real object that is being monitored by this monitor.
- The **Jobname** column shows the name of the job this Monitor resource accepts events from.

The **SORT, FIND, and RFIND subcommands** are supported. See “Varying the Format of the Command Output” on page 13 for information.

If you enter command code D for a resource, a panel similar to Figure 21 is displayed.

---

**DISPSCHD**

**Purpose**

DISPSCHD displays all resources that have scheduled override settings for the automation flags. Scheduled overrides are specified using the customization dialog. They are specific times when automation is turned off for a particular flag and resource.

**Syntax**
**DISPSCHD**

**Destination:**
- **system_name**
- **domain_ID**
- **sysplex_name**

**Dest_list:**
- ( **Destination** )

**Parameters**

**filter**
Is the filter to be applied for the output. Only resource names that match the filter criteria are displayed. Wildcard is supported. The filter consists of the major name and, optionally, of a minor name separated by a dot, for example, CICS*TRANS*, or *.*CEMT. Specify * to view major resources. To view minor resources specify xyz.*.

**TARGET**
For information on the TARGET parameter, see "OUTMODE Parameter" on page 12.

**OUTMODE**
For information on the OUTMODE parameter, see "OUTMODE Parameter" on page 12.

**Restrictions and Limitations**
The DISPSCHD command can only be used when SA z/OS is initialized.

**Examples**
If you enter dispschd a panel similar to Figure 22 on page 60 is displayed.
DISPSCHD

The System column shows the name of the system where the resource is defined.

The Resource column shows the name of resources that have NOAUTO specified in the automation control file. The resources are displayed in the order that they occur in the automation control file in.

The Fl (Flag) column shows an abbreviation of the automation flag name.

The Day column shows the day or days when automation is turned off. Valid values are:
- Monday through Sunday
- * (for every day)
- Weekend
- Weekday

The Start column shows the start time (hh:mm) of the non-automation period.

The End column shows the end time (hh:mm) that automation is disabled for.

Note: The display does not take into account any changes made with the INGAUTO command.

Entries for DEFAULTS or SUBSYSTEM will affect all subsystems that do not have their own automation flags coded.

DISPSFLT

Purpose

The DISPSFLT command lets you set up view filters for DISPSTAT.

Syntax
Parameters

REQ
Specifies the type of the request. It can be one of the following:

SET Specifies setting new filter settings.
GET Specifies returning the current filter settings. This is the default.

RESOURCE
Specifies the name of the resource, or resources, to be displayed. The format is
resource name. It can be a list of names. Wildcards are supported. If you specify
only one resource you need not put it in parentheses.

STATUS
Specifies the resource statuses that you want to display. If you specify a list of
statuses, separate them with a comma and enclose them in brackets (brackets
are not needed for a single status). If ‘¬’ or ‘\’ is used, all statuses except the
ones that you specify are displayed.

The following are predefined lists:

STATUS=* Lists all resource statuses.
STATUS=1 Lists all resources that are currently UP or ENDED.
STATUS=2 Lists all resources that are in a ‘normal’ status (DOWN, STARTED, UP,
AUTOTERM, AUTODOWN, ACTIVE, ENDED, ENDING, RESTART,
EXTSTART, RUNNING).
STATUS=3 Lists all resources that are in a ‘down’ status (DOWN, INACTIVE,
RESTART, AUTODOWN, CTLDOWN, STOPPED, BROKEN, MOVED,
FALLBACK).
STATUS=4 Lists all resources that are in a start transition status (RESTART,
STARTED, STARTED2, ACTIVE, RUNNING, EXTSTART).
STATUS=5 Lists all resources that are in a stop transition status (AUTOTERM,
ENDING, STOPPING, ABENDING, BREAKING)
DISPSFLT

TARGET
   For information on the TARGET parameter, see "TARGET Parameter" on page 10.

OUTMODE
   For information on the OUTMODE parameter, see "OUTMODE Parameter" on page 12.

If no parameters are specified, the filter settings are used.

If a subsystem name is specified, but no other parameters, for example, TARGET=, are specified, all subsystems of this name found on any system will be displayed.

Usage

The DISPSFLT command can also be used when SA z/OS is not initialized.

The DISPSFLT command lets you set filter options at operator logon from within the operator's initial REXX script.

Examples

If the operator is responsible for all resources starting with xyz, DISPSFLT will be called with the parameter RESOURCE=xyz. If you enter dispsflt a panel similar to Figure 23 is displayed.

<table>
<thead>
<tr>
<th>AOFKSTA3</th>
<th>SA z/OS - Command Dialogs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain ID = IPSNO</td>
<td>DISPSFLT --------</td>
</tr>
<tr>
<td>Operator ID = NETOP1</td>
<td>Date = 06/27/00</td>
</tr>
<tr>
<td>Time = 15:08:40</td>
<td></td>
</tr>
</tbody>
</table>

Resources ( *, *esour*, resource or jobname )
   *

Statuses ( *, status or choice, leading ~ or \ negates)
   RESTART STARTED STARTED2 ACTIVE RUNNING EXTSTART

Precoded: 1 All resources that are currently UP or ENDED
           2 All resources that are in a normal status
           3 All resources that are in a down status
           4 All resources that are in a start transition state
           5 All resources that are in a stop transition state

Targets ( system name, domain ID, sysplex name or * )
   KEY3

Command ===> PF1=Help PF2=End PF3=Return PF4=Clear PF5=Reset PF6=Roll PF9=Save PF12=Retrieve

Figure 23. DISPSFLT Command Dialog Panel

Filters are used to tailor the DISPSTAT display. To be displayed, a subsystem must pass all the filter criteria. If a filter is not explicitly set, it defaults to "*", which means that all subsystems pass it. Some filters allow you to specify a wildcard character, in the form of a trailing "."

- The Resources field contains the name of the resources or their jobs.
- The Statuses field shows the subsystem's current status. If "~", or \\
  "\" is specified, all statuses not specified are used.
- The Targets field shows the name of the target system.
If you press PF4 the filter settings are cleared.

Subsystems are evaluated to see if they meet the current filter criteria when:

- DISPSTAT is invoked with filter parameters.
- The filters are changed through the DISPSTAT Filters panel.
- The Refresh key (PF9) is selected from the DISPSTAT panel.

Note: If you make changes on this panel, and then decide that you do not want these changes to take effect, you can select the Reset key (PF5), which reinstates the field values that were present when you entered the panel. If you press PF9 the filter settings are saved until the session is terminated.

**DISPSTAT**

**Purpose**

DISPSTAT displays current information about one or more subsystems.

**Syntax**

- ALLKIDS subsystem
- KIDS subsystem
- ALLPARENTS subsystem
- PARENTS subsystem
- ALL subsystem

- ALL2CONS=YES
- ALL2CONS=NO

- STATUS=(status)

- TARGET= Destination
  - Dest_list
  - ALL

- OUTDSN=dsname
- OUTMODE=LINE
- AUTO
- NETLOG

**Destination:**

- system_name
- domain_ID
- sysplex_name

**Dest_list:**
Parameters

subsystem
Specifies the particular subsystem whose status or automation flags you want to check. You can specify any automated subsystem. Wildcards are supported. An asterisk (*) matches a string of arbitrary length and a percentage (%) matches a single character.

*ALLKIDS
Lists all children and children of children of the selected subsystem. Wildcards are not available for the subsystem here.

*KIDS
Lists the direct children of the selected subsystem. Wildcards are not available for the subsystem here.

*ALLPARENTS
Lists all parents and parents of parents of the subsystem. Wildcards are not available for the subsystem here.

*PARENTS
Lists the direct parents of the selected subsystem. Wildcards are not available for the subsystem here.

ALL
Displays the current information for all automated subsystems.

STATUS
Specifies the resource statuses that you want to display. If you specify a list of statuses, separate them with a comma and enclose them in brackets (brackets are not needed for a single status). If ‘~’ or ‘\’ is used, all statuses except the ones that you specify are displayed.

The following are predefined lists:

STATUS=*  
Lists all resource statuses.

STATUS=1  
Lists all resources that are currently UP or ENDED.

STATUS=2  
Lists all resources that are in a ‘normal’ status (DOWN, STARTED, UP, AUTOTERM, AUTODOWN, ACTIVE, ENDED, ENDING, RESTART, EXTSTART, RUNNING).

STATUS=3  
Lists all resources that are in a ‘down’ status (DOWN, INACTIVE, RESTART, AUTODOWN, CTLDOWN, STOPPED, BROKEN, MOVED, FALLBACK).

STATUS=4  
Lists all resources that are in a start transition status (RESTART, STARTED, STARTED2, ACTIVE, RUNNING, EXTSTART).

STATUS=5  
Lists all resources that are in a stop transition status (AUTOTERM, ENDING, STOPPING, ABENDING, BREAKING)
**ALL2CONS**

Specifies whether or not a complete list of columns is displayed. The parameter only applies to line mode. The following are valid values:

- **YES** Displays the complete list of columns. This is the default.
- **NO** Displays only 60 characters of display data for each resource. This command is beneficial when DISPSTAT is entered on the system console.

For example, you can specify:

```
DISPSTAT cics* OUTMODE=LINE ALL2CONS=NO
```

**TARGET**

For information on the TARGET parameter, see “TARGET Parameter” on page 10.

**OUTDSN**

For information on the OUTDSN parameter, see “OUTDSN Parameter” on page 13. Note, however, that for the DISPSTAT command the minimum record length of the output data set is 256 bytes.

**OUTMODE**

For information on the OUTMODE parameter, see “OUTMODE Parameter” on page 12.

If no parameters are specified, the filter settings will be used, or, if not set, DISPSTAT will show all resources from the local system.

If a subsystem name is specified, but no other parameters, for example, TARGET=, are specified, all subsystems of this name found on any system will be displayed.

**Restrictions and Limitations**

The DISPSTAT command can only be used when SA z/OS is initialized.

**Examples**

If you enter `dispstat`, you see information about all automated resources, as shown in Figure 24 on page 66.
The amount of data displayed depends on the screen size used. Press PF10 and PF11 to scroll horizontally.

PF4 lets you toggle to the INGLIST command dialog showing the same resources from the automation manager view.

- The **CMD** field allows you to invoke other SA z/OS command dialogs, or control the amount of information displayed. To use one of the command dialogs, type the appropriate letter next to the resource name, and press Enter. When you return to this panel the data within the command scope is refreshed. The current filters are not reapplied until you select Refresh (PF9).

For example, to invoke the INGTHRES dialog in order to change thresholds for JES2, move the cursor in front of JES2 and type `d`.

You can specify `X` to exclude subsystems.

**Note:** The U command code is supported if the AOFEXC04 exit is installed.

- The **Resource** field shows the name of the resource in its corresponding status color, for example, green when the system is UP.

- The **Status** field shows the current status of the resource as supplied by the automation agent.

- The **System** is the name of the system that this instance of the resource is from.

- The **Jobname** is the job name of the resource.

- The automation flags are as follows:

  **A** If this flag is set to N (No), no automation is on, regardless of how the other flags are set. If this is set to Y (Yes), SA z/OS checks each of the other flags to see if they are set to Y or N.

  **I** If this is on, automation issues the correct MVS start command for each subsystem when SA z/OS is initialized and assures that subsystems are started in the order specified in the automation control file.

  **S** If this is on, automation manages and completes the startup process of subsystems.
If this is on, automation takes the specified action for any MVS message that is not associated with the initialization, startup, shutdown, or restart of a subsystem. For example, if this flag is on, automation takes actions to restart the subsystem in case of an abend.

If this is on, automation manages the shutdown process for subsystems by issuing shutdown commands and responding to prompts for additional information.

If this is on, automation monitors subsystems to ensure that they are running and, if one fails, attempts to restart it according to the policy in the automation control file.

Each flag has a value of:

- **N** Off
- **E** User exit
- **Y** On
- **L** Log
- **?** Error

**Note:** DISPSTAT shows settings that have been set by the INGAUTO command. It does not show the effective setting for the specific flag. Use the DISPFLGS command to view the effective settings for the flags.

- The **Type** field shows the type of job, either MVS, NONMVS, TRANS(jent), or ONE-SHOT.
- The **Activity** field shows whether the resource is in a starting or stopping process.
- The **Last Changed** field shows the date and time that the status of the subsystem last changed.
- The **Last Monitored** field shows the time that the system was last monitored. If monitoring has been disabled this field is set to dashes.
- The **Description** field shows a description of the subsystem, from the automation control file.

Press PF5 to display the DISPSFLT panel. For a description of the DISPSFLT command see “DISPSFLT” on page 60.

**DISPSYS**

**Purpose**

The DISPSYS command displays information about the NetView automation agent setup and the environment that it runs on.

Note that if no data is defined or it is not relevant (for example, the sysplex group name for a system that is not in a sysplex) the word 'none' is displayed in the appropriate field.

**Syntax**

```
DISPSYS TARGET=system_name domain_ID sysplex_name OUTMODE=LINE AUTO NETLOG
```
DISPSYS

Parameters

TARGET
For information on the TARGET parameter, see "TARGET Parameter" on page 10.

OUTMODE
For information on the OUTMODE parameter, see "OUTMODE Parameter" on page 12.

Restrictions and Limitations

The DISPSYS command can only be used when SA z/OS is initialized.

Usage

DISPSYS is used when you want to see the system parameters and SA z/OS status information for the Sysplex Timer.

Examples

If you enter dispsys, you will see a panel similar to Figure 25.

For further details about the fields on the panel, see the online help, the customization dialog and IBM Tivoli System Automation for z/OS Defining Automation Policy.
Purpose

The DISPTREE command displays the dependency graph for a particular resource.

Syntax

```
    DISPTREE subsystem
```

```
    DEPENDENCY=START
    DEPENDENCY=STOP
    DEPENDENCY=GROUP
```

```
    TARGET=
    domain_ID
    sysplex_name
    OUTMODE=
    LINE
    AUTO
    NETLOG
```

Parameters

**subsystem**

The name of the subsystem the display is to be focused upon.

**DEPENDENCY**

Specifies the type of dependency that should be used to draw the dependency graph. The following options are available:

- **START**
  - The start relationship that is defined for the named subsystem will be used to draw the graph.

- **STOP**
  - The stop relationship that is defined for the named subsystem will be used to draw the graph.

- **GROUP**
  - The group relationship that is defined for the named subsystem will be used to draw the graph.

**TARGET**

For information on the TARGET parameter, see "TARGET Parameter" on page 10.

**OUTMODE**

For information on the OUTMODE parameter, see "OUTMODE Parameter" on page 12.

Restrictions and Limitations

The DISPTREE command can only be used when SA z/OS is initialized.

Examples

If you enter `disptree jes2`, a panel similar to Figure 26 on page 70 is displayed.
The Subsystem field shows the name of the subsystem

The Target field shows the name of the system where the subsystem resides

The Dependency field shows the dependency type (START, STOP, or GROUP)

If you position the cursor next to a resource and press PF4 the tree structure of the resource is shown.

PF5 lets you toggle to the DISPINFO command dialog.

**DISPTRG**

**Purpose**

The DISPTRG command displays the trigger conditions of a subsystem.

**Syntax**

```
>>>DISPTRG-resource
```

```
TARGET=<system_name>
```

```
OUTMODE=LINE
```

```
domain_ID
```

```
NETLOG
```

Figure 26. DISPTREE Command Dialog Panel
Parameters

resource
The name of the resource that the trigger condition is displayed for. The format is name/type[/system]. Wildcard is supported.

TARGET
For information on the TARGET parameter, see “TARGET Parameter” on page 10.

OUTMODE
For information on the OUTMODE parameter, see “OUTMODE Parameter” on page 12.

WAIT
Specifies the number of seconds to wait before reporting that a timeout has occurred if the automation manager does not provide the requested data. The maximum time interval is 999 seconds.
If omitted, the time interval is 30 seconds.

Restrictions and Limitations

The DISPTRG command can only be used when SA z/OS is initialized.

Examples

If you type the name of a resource that a trigger has been defined for, a panel similar to Figure 27 is displayed.

Figure 27. DISPTRG Command Dialog Panel

- The Resource field shows the name of the resource that uses the trigger.
- The Target field shows the name of the system where the subsystem is defined.
- The Trigger field shows the trigger that is defined for the subsystem.
DISPTRG

- The **Cmd** field allows you to specify command codes. The following command code is available:
  - **S** Shows details of the system defined.
- The **Type** field shows the type of the specified condition.
- The **Events** field shows the names of the specified events. Event that are satisfied are shown in white. Unset events are shown in red.

DRAINJES

**Purpose**

DRAINJES drains (that is, halts) JES2 resources prior to JES2 shutdown. Specifically, DRAINJES issues commands to drain the initiators, syslog, offloader tasks, lines, printers, punches, and readers (depending on which resources are listed in the automation control file).

**Syntax**

```
DRAINJES subsystem
```

```
TARGET= Destination
Dest_list
```

```
OUTMODE= LINE AUTO NETLOG
```

**Destination:**

```
-system_name
-domain_ID
-sysplex_name
```

**Dest_list:**

```
(Destination)
```

**Parameters**

**subsystem**

Specifies the name of the JES2 subsystem whose resources you want to drain. If you are in fullscreen mode and you do not specify a subsystem, a panel is displayed where you can specify the subsystem. If you are in line mode an error message is displayed.

**TARGET**

For information on the TARGET parameter, see "TARGET Parameter" on page 10.

**OUTMODE**

For information on the OUTMODE parameter, see "OUTMODE Parameter" on page 12.
Restrictions and Limitations

The DRAINJES command can only be used when SA z/OS is initialized.

Usage

Normally, you do not need to issue this command because it is performed automatically as part of the SHUTDOWN processing. However, if SA z/OS is unable to shut down a particular JES2 subsystem, you can use DRAINJES.

Examples

Assuming the JES2 subsystem name is JES2, type drainjes jes2 to drain the JES2 resources.

EXPLAIN

Purpose

The EXPLAIN facility gives you a description of the automation statuses that can occur for SA z/OS subsystems. It explains what the status means, why the subsystem may have got into the status, and how another status can be achieved.

Syntax

```
EXPLAIN status subsystem
```

Parameters

If you do not specify any parameters, a menu panel is displayed where you can specify either an automation manager status, automation agent status, or the name of a subsystem.

status
    The status you want to see explained.

subsystem
    The name of the subsystem you are investigating. Its status is queried and an explanation of the automation agent status is displayed.

TARGET
    For information on the TARGET parameter, see “TARGET Parameter” on page 10.

Restrictions and Limitations

The EXPLAIN command can only be used when SA z/OS is initialized.

Usage

Use EXPLAIN when you want information about an automation status.
**EXPLAIN**

**Examples**

If you enter EXPLAIN, a panel similar to Figure 28 is displayed.

![Figure 28](image)

**INGAMS**

**Purpose**

The INGAMS command displays details about currently registered automation managers and automation agents, and lets you change the operation mode (PAM or SAM) of the automation managers. Note that it is only possible to make a secondary automation manager a primary one, not vice versa.

INGAMS lets you dynamically refresh the definition data for an automation manager and the corresponding ACFs with the contents of a specified configuration data set. The consistency of the configuration data as used by the automation agent and the associated ACFs is maintained. Data that would cause resources to contain conflicting definitions is rejected during configuration refresh with an appropriate message.

You can also use the INGAMS command to initiate and view diagnostic information for the primary automation manager.

**Syntax**
Parameters

**STATUS**
Displays a list of all automation manager and automation agents in the sysplex that are registered to the same XCF group.

**XSTATUS**
Displays a list of all automation manager and automation agents in the sysplex that are registered to the same XCF group denoted by the PLEXID specification for the local automation agent.

**SET**
Changes the operation mode of the specified automation manager.

name
The name of the automation manager.

**MODE**
Specifies the mode of operation, this can be:

- **START=HOT**
- **START=WARM**
- **START=COLD**
- **STOPREC**
- **DISABLE**
- **ENABLE**
- **SUSPEND SYSTEMS=(*all|system1 system2 ...)**
- **RESUME SYSTEMS=(*all|system1 system2 ...)**
- **REFRESH CFG=dsnname**
- **OUTDSN=dsnname**
- **OUTMODE=LINE**
- **AUTO**
- **NETLOG**

**TARGET**
- **system_name**
- **domain_id**
- **sysplex_name**

**STATS**
Report option:

- **SUMMARY**
- **DETAIL**
INGAMS

PAM
Primary automation manager

START
Specifies how an automation manager switch should be carried out. Valid values are HOT, WARM, or COLD.

DETAILS
Displays detailed information about the specified automation manager.

DIAG
Displays the diagnostic panel where you can either request that a snapshot of the current state image queue be taken or that recording of the inbound work items be started or stopped. The output file must be a sequential data set.

REQ
Is the function to be performed. It can be one of the following:
- SNAPSHOT
  Write a snapshot of automation manager data.
- STARTREC
  Start recording.
- STOPREC
  Stop recording.
- STATS
  Display work item statistics.
- STARTMON
  Activate work item queue monitoring.
- STOPMON
  Stop work item queue monitoring.

INTERVAL
The time period in seconds between two monitoring cycles. The value can be in the range of 10–999.

STATS
Specify the type of work item statistics report to generate: A SUMMARY (the default) or a DETAIL report, which shows details about all tasks.

DISABLE
Causes the automation manager to stop writing to the takeover file. It also deallocates the takeover file from the automation manager. This is only necessary when performing repair operations on the takeover file.

ENABLE
Causes the automation manager to start updating the takeover file for each work item processed.

SUSPEND
Tells the automation manager to suspend sending orders to start or stop resources that are hosted by the systems that are specified with the SYSTEMS parameter.

RESUME
Tells the automation manager to resume sending orders to start or stop resources that are hosted by the suspended systems that are specified with the SYSTEMS parameter.

SYSTEMS
The parameter to define the systems that the sending of orders is suspended or resumed for. If you specify multiple systems, they must be enclosed in parentheses.
Alternatively, you can use `*all` to specify all the systems that are controlled by the automation manager.

**REFRESH**

Updates an automation manager with the definition data that is stored in the specified configuration file. This also reloads the appropriate ACFs on the active automation agents.

**Notes:**

1. INGAMS REFRESH does not delete global variables that are associated with ENTRY TYPEs that were deleted from the policy database before the last build if those ENTRY TYPEs are not associated with a subsystem. However, if a keyword/value is removed from the ENTRY TYPE, this is a change to the E-T. Therefore, INGAMS REFRESH rebuilds the E-T and the deleted keyword/value pair is removed from the global variables.
   All deleted entry types and their associated global variables are reset during the next SA z/OS COLD start or during ACF COLD processing.

2. The configuration refresh on the automation agent runs parallel to other activities. Thus any start/stop orders may be processed before the new data is loaded.

**CFG**

This is the name of a partitioned data set that holds the configuration file member and the ACF fragments. The name can be a fully qualified data set name or generation data group (GDG) name. You can use an asterisk (*) to indicate a reload of the configuration file that was previously used or a different generation of the GDG that was previously used.

- **dsname**
  
  Specifies a data set name. If you specify a GDG base data set name, the latest generation (0) is used.

- **dsname (generation)**
  
  Specifies the name of a generation data group (GDG) and the generation number (zero or a negative signed integer) of a generation data set within the GDG, for example, `*(-2)`.

**DSN**

Lets you specify the name of the data set that contains the snapshot or work item recording data. You can only use a sequential file.

The output data set must be pre-allocated with a RECFM FB and LREC 80. It must be allocated as a single volume data set, without candidate volumes for further expansion.

**TARGET**

For information on the TARGET parameter, see “TARGET Parameter” on page 10.

**OUTDSN**

For information on the OUTDSN parameter, see “OUTDSN Parameter” on page 13.

**OUTMODE**

For information on the OUTMODE parameter, see “OUTMODE Parameter” on page 12.

**Restrictions and Limitations**

None.
Security Considerations

The INGAMS command supports resource level security. If turned on, the following profiles in class SYSAUTO are checked:

<table>
<thead>
<tr>
<th>Profile</th>
<th>Authority</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGT.sysplex.xcfgrp.RES._CONFIG</td>
<td>UPDATE</td>
<td>When function option is REFRESH</td>
</tr>
<tr>
<td>AGT.sysplex.xcfgrp.RES._MANAGER</td>
<td>UPDATE</td>
<td>When function option is SET</td>
</tr>
<tr>
<td>AGT.sysplex.xcfgrp.RES._MANAGER.CONTROL</td>
<td>CONTROL</td>
<td>When function option is DISABLE, ENABLE, SUSPEND or RESUME</td>
</tr>
<tr>
<td>AGT.sysplex.xcfgrp.RES._MANAGER.DIAG</td>
<td>UPDATE</td>
<td>When function option is DIAG for all function other than STATS (other than REQ=STATS)</td>
</tr>
</tbody>
</table>

Usage other than listed in the above tables does not cause a resource level security check. For further details, refer to IBM Tivoli System Automation for z/OS Planning and Installation.

Usage

The INGAMS command operates sysplexwide. For an overview see "Overview of Commands that Operate Sysplexwide" on page 9.

When using the DIAG parameter make sure that the data set for writing a snapshot or starting recording is online and free before issuing a command. The data set must be used exclusively by the automation manager (DISP=OLD) for making snapshots or recording.

Examples

If you enter ingams a panel similar to Figure 29 on page 79 is displayed.
The **Cmd** field allows you to specify command codes to initiate various activities related to automation managers. To use one of the command codes shown, type the appropriate letter next to the resource name, and press Enter. The amount of data displayed depends on the screen size. Press PF10/PF11 for horizontal scrolling. The following command codes are available:

**A** Displays the automation manager menu panel allowing you to:
- Modify the automation manager's operation mode. You can only change a secondary automation manager to a primary one.
- Enable and disable takeover file processing
- Suspend or resume the sending of orders from the automation manager to the automation agent on the system specified. This is useful if, for example, you have erroneously issued a command to shut down a system or sysplex. In such a situation, you now have the chance to rectify the error.

You should also note that while the automation manager is suspended for a particular system, it continues to respond to queries (such as INGLIST or INGVOTE) and any automation action for other systems within its scope.

You should see the ["Controlling the Secondary Automation Manager's Functions" on page 81.](#)

**B** Displays additional information about the automation manager or automation agent. For an automation agent, the DISPSYS command is issued to show details about the agent and the environment the agent runs on. See the ["Showing Details about the Automation Manager" on page 82.](#)

**C** Displays the refresh configuration data panel where you enter the name of the data set that contains the automation manager configuration members. This is then used to update the definition data for the automation manager in real time and to reload the corresponding automation configuration files for the active automation agents. This command thus maintains the consistency of the configuration data that is used by the automation agent and its automation configuration files. See the ["Refreshing the Configuration Data Sysplexwide" on page 84.](#)
Displays the utility panel where you can perform the following diagnostic functions:
- Write snapshot.
- Start or stop recording.
- Display work item statistics
- Start or stop monitoring of the work item queue.

For further information, see “Diagnostic Functions” on page 85 and “How to Use the Diagnostic Option of INGAMS” in IBM Tivoli System Automation for z/OS User’s Guide.

- The **System** field shows the name of the system where an instance of the automation manager or automation agent runs.
- The **Member** field shows the name of the agent or manager. The name must be unique within the XCF group. The member name for an automation manager is automatically generated and consists of the z/OS system name plus a 1-character suffix running from 1 to 9. The z/OS system name is padded with ‘$’ to create an 8-character system name before appending the suffix.

**Note:** The member name of an automation agent is equal to the z/OS system name.

- The **Role** field shows the type of the system. It can be one of the following:
  - **AGENT**
    - An automation agent.
  - **PAM**
    - The primary automation manager.
  - **SAM**
    - The secondary automation manager. There can be more than one.
- The **Status** field shows the status of the member. It can be one of the following:
  - **COMM**
    - The type of communication being used between the automation manager and the automation agent.
  - **NOT READY**
    - The automation agent or manager is initializing.
  - **PENDING**
    - The automation manager is in the process of initializing as a primary automation manager (PAM).
  - **READY**
    - The member is completely initialized.
  - **REFRESH**
    - The automation manager is performing a configuration refresh.
  - **SELECTED**
    - The automation manager is selected to become the next primary automation manager (PAM).
  - **STOPPING**
    - The automation manager is terminating.
  - **SUSPENDED**
    - The automation agent has been placed in suspended mode.
- The **Sysplex** field shows the name of the sysplex.
- The **XCF-Group** field shows the name of the associated XCF-Group.
- The **Release** field shows the SA z/OS Version and Release of the automation manager or automation agent.
- The **Comm** field shows the type of communication in use between the automation manager and the automation agent.
- The **PA** field shows if this is a primary agent (PA) communicating with a higher level automation manager.
- The **Runmode** field shows the Runmode assigned to this system (agents only).
The **Pref** field shows the preference value assigned to various automation instances.

The **SMF ID** field shows the SMF ID assigned to the system.

The **Primary FP** field shows the name of the primary focal point defined to the system.

The **Backup FP** field shows the name of the backup focal point defined to the system.

Values for the columns **SMF ID**, **Primary FP** and **Backup FP** are only available if the agent is a member of the extended XCF communication group.

The FIND/RFIND/SORT commands are supported. See “Varying the Format of the Command Output” on page 13 for information.

**Controlling the Secondary Automation Manager's Functions**

If you enter command code A for a secondary automation manager, a panel is displayed where you can specify the new mode of operation, as shown in Figure 30. Switching into primary mode is considered a directed takeover.

**Make primary Automation manager**

This allows you to change the operation mode of a secondary automation manager to become primary (PAM). This is considered a directed takeover. Optionally, you can specify the start mode for the new PAM. Valid start modes are HOT, WARM or COLD. The default is HOT.

**Note:** A primary automation manager cannot be changed to a SAM.

**Enable Takeover file**

This causes the automation manager to start updating the Takeover file for
each work item that is processed. This is necessary after the Takeover file has been enlarged or another error condition resolved.

**Disable Takeover file**
This causes the automation manager to stop writing to the Takeover file. The Takeover file is freed from the automation manager.

**Suspend System**
This causes the automation manager to stop sending orders to the agent running on the system specified left of the action. You can specify more than one system name separated by a blank character. Specify *all to suspend all systems that are controlled by the automation manager.

**Resume System**
This causes the automation manager to resume sending orders to the agent running on the system specified left of the action. You can specify more than one system name separated by a blank character. Specify *all to resume all systems that are controlled by the automation manager.

Fill in the following fields:

**Automation manager**
Specify the name of the automation manager.

**Target**
The name of the system where the automation manager resides. This is only necessary when the automation manager is not within the local sysplex. You can specify either the system name, the domain ID or the sysplex name.

**Showing Details about the Automation Manager**
If you enter command code B for an automation manager, a panel similar to Figure 31 on page 83 is displayed.
The **Takeover file** section shows the data set name of the takeover file, its status, and the I/O interval time.

The status may be ENABLED, ENABLED - IO SUPPRESSED (meaning the takeover file is still allocated but the PAM has stopped writing to it because of an earlier I/O error), or DISABLED (meaning the takeover file has been deallocated and the manager has stopped writing to it as a result of the operator selecting to DISABLE the takeover file).
SA z/OS delays I/O to the takeover file for the number of seconds specified by the I/O interval. The in-storage pages are only marked to be written. This reduces the amount of I/O to the takeover file significantly, giving an update work item the same performance characteristics as a query work item.

The **Service level** section shows the current APAR level including the compilation date of the HSACMAIN module and the automation manager framework. This is used primarily for debugging purposes.

The **PARMLIB settings** section shows the BLOCKOMVS and Preference Settings.

The **Workitem Statistics** section shows the number of work items received from the various automation agents (that is, external work items) and the internally generated work items.

The **CPU Time** shows the processor time (in seconds) used by the automation manager.

The **Suspended Systems** section shows those systems that the sending of orders by the automation manager has been suspended for.

The **Logic Deck** section shows the date and time when the logic deck was built and the last APAR number of the logic deck.

The **Diagnostic Info** section shows details about the size of the state image and other useful information. This can be used when allocating the data set that will hold the snapshot data.

The **Configuration** section displays the data set name that the configuration is loaded from, the main configuration member name, and the include members with their timestamps.

Note that the last two sections (Configuration, and Diagnostic Info) are only present for the primary automation manager.

**Refreshing the Configuration Data Sysplexwide**

If you enter command code C for an automation manager, a panel similar to Figure 32 on page 85 is displayed.
In the **Configuration data set name** field specify the name of the data set containing the automation manager configuration members. The name can be a fully qualified data set name or a generation data group name. You can use * to indicate a reload of the configuration from the data set that was used previously, or a different generation of the previously used GDG.

In the **Target** field specify the name of the system where the automation manager resides. Specifying this is only necessary if the automation manager is not within the local sysplex. You can specify either the system name, the domain, or the sysplex name.

**Diagnostic Functions**

If you enter command code D for a primary automation manager, a panel similar to [Figure 33 on page 86](#) is displayed.
**Initiating Diagnostic Functions**

For options 1–3, fill in the following fields:

**Data set name**

Specify the name of the data set that will hold the snapshot data. The data set can be a sequential file or a member of a partitioned data set. However the recording data set can only be a sequential file.

**Notes:**

1. A data set name is required when writing a snapshot or when starting recording.
2. Make sure that the automation manager has the appropriate authority to write to the data set.

**Target** Specify the name of the system where the automation manager resides. It is only required when the automation manager is not in the local sysplex. You can specify either the system name, the domain ID or the sysplex name.

Use options 5 and 6 to start and stop work item queue monitoring. The monitor examines the work item input queue for expired work items and issues message INGX1011I when more work items arrive than can be processed by the automation manager within the monitoring interval.

When you start monitoring, fill in the following field:

**Interval** Specify monitoring interval in seconds. The valid range is 10–999.

**Work item Statistics**

To display work item statistics, you enter 4 in the Action field without specifying a data set name or system. This displays a panel similar to Figure 34 on page 87.
This panel shows history information about the work items processed by the automation manager. The automation manager keeps track of the last 500 work items processed by each of the tasks that build the automation manager kernel.

The following data is shown:

**Snapshot**
The snapshot timestamp shows the time of the query.

**Task**
The name of the task.

**Queue**
The number of elements in the workitem input queue.

**CPU Time**
The processor time (in seconds) used by the automation manager.

**Status**
The status of the task. It is one of the following:

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUSY</td>
<td>The task is processing a work item.</td>
</tr>
<tr>
<td>IDLE</td>
<td>The task is waiting for work.</td>
</tr>
<tr>
<td>NOTIDLE</td>
<td>The task has completed work but is not waiting for work.</td>
</tr>
<tr>
<td>UNKNOWN</td>
<td>History records have not yet been defined for this task.</td>
</tr>
</tbody>
</table>

**Since**
This is the number of seconds that the task has been processing the work item (elapsed time). If this number is unexpectedly high, it is an indication that something is wrong. The following coloring is used depending on how long the work item has been active for:

<table>
<thead>
<tr>
<th>Color</th>
<th>Active longer than</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red</td>
<td>60 seconds</td>
</tr>
<tr>
<td>Yellow</td>
<td>30 seconds</td>
</tr>
<tr>
<td>Blue</td>
<td>15 seconds</td>
</tr>
</tbody>
</table>

The columns to the right of the **Since** column show the number of work items that have been processed in one minute, starting from the time the statistics query was made or refreshed (with INGAMS DIAG REQ=STATS).
You can use the S command code to display details of the work item history for the selected task, such as the starting time and total processing time of the work item, as shown in Figure 35.

**INAUTO**

**Purpose**

The INAAUTO command lets you change the settings of the automation flags for the specific subsystem, for the MVSESA, SUBSYSTEM or DEFAULTS resources or for a minor resource under one of those major resources. You can change the setting to ON, OFF, EXITS or LOG either indefinitely or for a specified time period. You can also ask for a flag to be set back to the value specified in the agent's current ACF.

**Syntax**

```
>>INAUTO  EXITS
         OFF
         ON
         RESET
         LOG

MVSESA
     DEFAULTS
     SUBSYSTEM
     subcomponent

SCOPE=ONLY

FLAG=RECOVERY
    AUTOMATION
    INITSTART
    RESTART
    START
    TERMINATE

INTERVAL=dd:hh:mm
```
### INGAUTO

**TARGET**
- Destination
- Dest_list

**OUTMODE**
- LINE
- AUTO

### Destination:
- system_name
- domain_id
- sysplex_name

### Dest_list:
- Destination

### Parameters

If you do not specify a resource (MVSESA, SUBSYSTEM, DEFAULTS, resname or subcomponent) and are running in full screen mode, a menu appears prompting you for a resource name. If you are in line mode, a resource is a required parameter.

**EXITS**
- Enables user exits for the flag that you specify.

**OFF**
- Sets the automation flag that you specify to off.

**ON**
- Sets the automation flag that you specify to on.

**RESET**
- Resets the automation flag to the values that you specified in your policy database.

**LOG**
- Writes the commands and replies to the netlog if an event occurs that triggers the automated action.

**DEFAULTS**
- Specifies the value DEFAULTS. This sets automation flags for all resources that do not have their own flags defined.

**SUBSYSTEM**
- Sets defaults for all subsystems.

**MVSESA**
- Sets defaults for all MVS subcomponents.

**resname**
- Specifies the name of a particular resource, for example, a subsystem name like JES2 or a resource under MVSESA such as WTOBUF. You can specify any resource known to the Automation Agent.
- Wildcards are supported, for example, IMS* or *PRD.
**INGAUTO**

*subcomponents*
Specifies the subcomponents of the resource. Enter a period (.) between each subcomponent name. When specifying a subcomponent without specifying a major resource, MVSESA is used as a major resource. The period (.) in front of the subcomponent must then be omitted.

**SCOPE**
Specifies whether to change automation for this resource and its dependent resources, dependent resources only, or this resource alone. Valid options are the following:

- **ONLY**
  Specifies this resource alone. This is the default.

- **ALL**
  Specifies this resource and its dependent resources.

- **CHILDREN**
  Specifies dependent resources only.

**FLAG**
Specifies the automation flags to set on, off, reset, or enable a user exit for, (depending on whether you specified INGAUTO ON, OFF, RESET or EXITS). See *IBM Tivoli System Automation for z/OS Customizing and Programming* for further information on automation flags. Valid options are the following:

- **AUTOMATION**
  This sets the Automation flag, which is the controlling flag. If you set the Automation flag to OFF, SA z/OS turns all automation off. If the Automation flag is ON, SA z/OS checks the other flags.

- **INITSTART**
  Sets the Initstart flag. This is checked after SA z/OS initialization for the first start of an application. If this flag is on, SA z/OS will start the resource, provided its goal is to be available and all other conditions for its startup have been met.

- **RECOVERY**
  Sets the Recovery flag. If this is set to on, SA z/OS will perform recovery actions other than restarting a resource.

- **RESTART**
  Sets the Restart flag. If this is set to on, SA z/OS will restart the resource if possible.

- **START**
  Sets the Start flag. If this is set to on, SA z/OS will manage and complete the startup of the resource.

- **TERMINATE**
  Sets the Terminate flag. If this is set to on, SA z/OS will process automated shutdown requests for the resource.

**INTERVAL=dd:hh:mm**
Specifies the period of time the flag will be changed for.
At the end of the interval, the flag is reset to the state defined in the automation control file.
If no interval is specified, the flag is changed indefinitely.
Restrictions and Limitations

The INGAUTO command can only be used when SA z/OS is initialized.

A flag cannot be reset for an interval.

Usage

If the Automation flag is currently set to off, and you have changed one of the other flags, your change will not be effective until the Automation flag is set to on. Occurrences will be detailed in the netlog.

If you have selected EXITS for an automation flag, and none have been specified in your automation control file, automation will be set to on. Occurrences will be detailed in the netlog.

SCOPE=ONLY is the only valid scope for DEFAULTS, a resource type, and subcomponents. ALL and CHILDREN are not valid.

Make sure you specify the correct subcomponent name. Even if you specify a subcomponent that does not exist, the message FUNCTION SUCCESSFULLY COMPLETED will appear.

Examples

To turn all automation on in SA z/OS for TSO and for 2 hours only, enter the following:

```
ingauto on,tso,flag=automation,interval=00:02:00
```

To turn all automation on for the WTOR buffer shortage recovery, enter the following:

```
ingauto on,mvsesa.wto
```
Purpose

The INGCF command supports all the functions of SA z/OS that deal with coupling facilities. It supports full mode and line mode; for line mode capability, see "INGCFL" on page 110. If you issue INGCF in line mode, only the display function is available.

The INGCF command supports the following parameters:

- **DRAIN**
  Removes all allocated structures from the coupling facility, to disconnect the coupling facility from the systems of the sysplex, and to inactivate the coupling facility.

- **ENABLE**
  Activates a coupling facility, to connect it with the systems of a sysplex and to populate it with structures.

- **MAINT**
  Puts the coupling facility into or takes it out of maintenance mode.

- **PATH**
  Displays and controls the sender paths of the target coupling facility. It sets the sender paths ONLINE and OFFLINE physically and logically.

- **STRUCTURE**
  Displays detail information and rebuilds or deletes a selected structure on the target coupling facility. It also lets you start and stop duplexing.

INGCF associates a status with every coupling facility, and a condition with every structure (instance) that is allocated on the target coupling facility. The structure condition is influenced by the release level of the system that allocated the structure. The INGCF functions use the coupling facility state and the structure conditions to determine which action can be performed in any given situation.
Therefore, the DRAIN and ENABLE functions can enforce a correct sequence of actions for complex tasks such as draining or restoring a coupling facility.

If the selected action impacts the sysplex configuration it must be confirmed before execution.

**Authorizations**

The actions that you can initiate with INGCF depend on your authorizations. The panels show your authorization type. Note that the authorization types apply to the current function, and that your authorization type may vary for different functions.

The following authorization types exist:

**DISPLAY**

You cannot initiate any action that affects the sysplex configuration.

**ALL BUT (ACTIVATE|SHUTDOWN)**

This type only occurs in the DRAIN and ENABLE command dialogs. You can rebuild structures, force the deletion of structures and set the sender paths offline and online, but you cannot inactivate or activate the coupling facility.

**ALL**

You can initiate all actions from the corresponding panel.

Depending on your authorizations, it is possible that you have, for example, authorization type **ALL** for the STRUCTURE function, and authorization type **DISPLAY** for the DRAIN function.

**Note:** The actions that modify the sysplex configuration are marked by an asterisk (*) in the following descriptions.

**Syntax**

```plaintext
/SM590000/SM590000

INGCF

Drain cfname
Enable cfname
Maint cfname MODE= OFF ON
Path cfname
Structure cfname CONDITION= NO

TARGET= system_name
       domain_ID
       sysplex_name

OUTMODE= LINE AUTO NETLOG

Cfname:

CF_name
```
INGCF

Parameters

Drain
Prepares a coupling facility for removal from the sysplex.

Enable
Integrates or reintegrates a coupling facility into a sysplex.

Maint
Controls the maintenance mode of a coupling facility.

Path
Controls the sender paths of a coupling facility.

Structure
Offers manipulation of individual structures (detail information, rebuild, deletion).

\texttt{CF\_name}
The name of the target coupling facility for the specified function. The default is a selection panel that shows all available coupling facilities of the sysplex.

CONDITION
Specify YES if you want to get the current condition for each structure. Selecting this option increases the response time required to build the display. The default is NO.

MODE
Specify ON if you want to put the coupling facility into maintenance mode, or OFF to take it out of maintenance mode.

TARGET
For information on the TARGET parameter, refer to "TARGET Parameter" on page 10.

OUTMODE
For information on the OUTMODE parameter, refer to "OUTMODE Parameter" on page 12.

Restrictions and Limitations
The ENABLE and the PATH functions require that the active IODF is catalogued. Otherwise, sender path information cannot be retrieved in certain situations.

INGCF ENABLE assumes that the receiver paths from the coupling facility to the systems of the sysplex have been defined and activated. This requires a POR of the CPC that the coupling facility resides on.

Coupling Facility States
The status of a coupling facility can be as follows:

\textbf{ACTIVATING}
The coupling facility is being activated and will then become DRAINED.

\textbf{DEACTIVATING}
The coupling facility is being deactivated and will then become INACTIVE.

\textbf{DRAINING}
The coupling facility is being disconnected from the connected systems.

\textbf{DRAINED}
The coupling facility does not have a connection to any system and can be removed from the sysplex.
DRAINED NOHWACC
The coupling facility does not have a connection to any system, but cannot be removed from the sysplex because the BCP (Basic Control Program) internal interface is not available.

Note: This status is also displayed when the coupling facility has been deactivated from the HMC (Hardware Management Console) but the XCF display commands still return the name of the coupling facility.

ENABLING
The coupling facility is being connected to the systems of the sysplex that use it.

FORCING
Allocated structures are being deleted from the coupling facility. This only happens with structures that have no active connectors, and with these only when they cannot be rebuilt by system-managed rebuild.

INACTIVE
The coupling facility is not active.

INACTIVE NOHWACC
The coupling facility is not active and cannot be activated because the BCP Internal Interface does not have access to the appropriate Support Element.

MAINTMODE
The coupling facility is in maintenance mode and has not allocated any structures. It can now be removed from the configuration or deactivated without touching the sender paths. If you want to move structures back to the coupling facility you need to take it out of maintenance mode before populating it.

MAINTMODE OFFLINE
The coupling facility is in maintenance mode and may have allocated structures. At least one system has set all its sender paths to the coupling facility to OFFLINE. XES will reject any rebuild command to the coupling facility.

MAINTMODE NOHWACC
The coupling facility is in maintenance mode and has not allocated any structures. It can now be removed from the configuration without touching the sender paths. However, this must be done manually because automation does not have access to the appropriate Support Element. If you want to move structures back to the coupling facility you need to take it out of maintenance mode before populating it.

NORMAL
The coupling facility may have allocated structures and is connected to all systems.

NORMAL MAINTMODE
The coupling facility is in maintenance mode but has allocated one or more structures. The coupling facility is no longer eligible for structure allocations.

NORMAL OFFLINE
The coupling facility may have allocated structures. At least one system has set all its sender paths to this coupling facility to OFFLINE. XES will reject any rebuild request for this coupling facility.
The coupling facility is active but not defined in the active CFRM policy.

The coupling facility is being populated with all those structures that have it on the first place in their preference list.

Either all allocated structures that can be rebuilt are being removed from the coupling facility by the XES rebuild process (initiated by DRAIN), or one particular such structure is being removed (initiated by the STRUCTURE).

Structure Conditions

The condition of an allocated structure can be:

- **Rebuild is not supported.**
  The structure can neither be rebuilt, nor can its deletion be forced.
  The structure has at least one active connector that does not support user-managed rebuild, and at least one active connector that does not support system-managed rebuild.

- **System-managed processes not supported.**
  The structure cannot be rebuilt, nor can its deletion be forced.
  System-managed rebuild, which is a system-managed process, is not possible for one of the following reasons:
  - The structure was allocated from a system with OS/390 V2.7 or earlier.
  - The CFRM couple data sets have not been formatted to support system-managed processes (ITEM NAME(SMREBLD) NUMBER(1) was not specified).

  **Note:** In certain rare cases system-managed processes are not supported although the condition that is displayed on the DRAIN panel seems to indicate the contrary. Then, the rebuild will be initiated, but will fail with message IXC367I indicating that system-managed processes are not supported for the structure.

- **No alternate coupling facility defined or available.**
  The structure can neither be rebuilt, nor can its deletion be forced.
  The structure has an active connector and supports rebuild but does not have an alternate coupling facility defined in its preference list, or the alternate coupling facilities that are defined in the preference list are currently unavailable.

- **Insufficient space detected for rebuild.**
  The structure cannot or could not be rebuilt. Its deletion cannot be forced.
  No alternate coupling facility has enough space to rebuild the structure.

- **Preference list is empty.**
  The structure cannot be rebuilt because its preference list is currently empty. A possible reason for this is a pending policy change, see P column.

- **Structure is pending deallocation.**
  XES accepted a forced deletion of the structure but does the real deallocation later.
Note: This status can only occur when MVS APAR OW39404 has not been installed.

**Structure is being rebuilt.**
The structure is being rebuilt to another coupling facility.

**Duplex rebuild is being stopped.**
Two instances of the structure were maintained on different coupling facilities. The application is being disconnected from that instance that is allocated on the target coupling facility. After disconnecting, the instance is deleted.

**No connection exists.**
The structure cannot be rebuilt, but you can force its deletion.

The structure does not have any connections and cannot be rebuilt with system-managed rebuild.

**No alternate coupling facility for structure with no connections.**
The structure cannot be rebuilt, but you can force its deletion.

The structure does not have any connections. It could be rebuilt with system-managed rebuild, but no alternate coupling facility is defined in its preference list or available.

**No alternate coupling facility for structure with no active connections.**
The structure cannot be rebuilt, but you can force its deletion.

The structure has only DISCONNECTING, FAILED, or FAILED-PERSISTENT connections. It could be rebuilt with system-managed rebuild, but no alternate coupling facility is defined in its preference list or available.

**The structure’s initial size is less than its actual size.**
The SIZE value is greater by twice the INITSIZE value. The structure can be rebuilt, but a SIZE value that is greater than twice the INITSIZE might cause the following:

- It might be impossible to allocate a structure at a size of INITSIZE, because the amount of control storage that is required to support the SIZE value might actually be larger than INITSIZE.
- If the allocation succeeds, it might result in a structure with a proportionally large amount of its storage allotted to structure controls, leaving too few structure objects to be exploited usefully by the associated application.

For example, if you have requested a maximum size that is very much larger than the initial size, the system will attempt to use a proportionally large amount of the allocated storage for its controls. The result could be that the allocated storage contains control structures for the future maximum size of the structure and insufficient storage might remain for the application’s initial use.

**No active connection exists.**
The structure cannot be rebuilt, but you can force its deletion.

The structure has only DISCONNECTING, FAILED, or FAILED-PERSISTENT connections and cannot be rebuilt with system-managed rebuild.

**Note:** INGCF DRAIN deallocates structures with this condition as part of the REBUILD action (see REBUILD(10) in “INGCF DRAIN” on page 99).
INGCF STRUCTURE accepts a rebuild request for structures with this condition, but deallocates them (see Rebuild (R)).

No connections. System-managed rebuild supported.
The structure can be rebuilt.
The structure does not have any connections, but can be rebuilt with system-managed rebuild.

No active connections. System-managed rebuild supported.
The structure can be rebuilt with system-managed rebuild.
User-managed rebuild is not possible for the structure because it has only DISCONNECTING, FAILED, or FAILED-PERSISTENT connections.

System-managed rebuild is supported
The structure can be rebuilt.
The structure has active connectors. At least one active connector does not support user-managed rebuild, but all active connectors support system-managed rebuild.

Duplex rebuild is active.
The application is connected to two instances of the same structure on different coupling facilities.

[No condition]
When no condition is displayed, the structure can be rebuilt.
The structure has at least one active connection, and all its active connectors support user-managed rebuild.

Structure is awaiting rebuild.
The structure has been selected for rebuild but has not been processed yet.

Structure is currently allocated on cf_name.
The structure can be rebuilt on the target coupling facility with the POPULATE action of the ENABLE function. It is currently allocated on the cf_name coupling facility, but the target coupling facility precedes cf_name in the structure’s preference list. This condition is displayed only in the ENABLE command dialog.

Structure allocated in cf_name cannot be rebuilt to this CF.
The structure can probably not be rebuilt on the target coupling facility with the POPULATE action of the ENABLE function. It is currently allocated in the cf_name coupling facility, but the target coupling facility precedes cf_name in the structure’s preference list. And, the actual size of the structure is greater than the free space of the target coupling facility. This condition is displayed only in the ENABLE command dialog.

Example
If you issue INGCF without any parameters, a panel with all coupling facilities of the sysplex is displayed, as shown in Figure 37 on page 99.
Specify a function for a selected coupling facility and press Enter.

**INGCF DRAIN**

### Purpose

The DRAIN function of INGCF facilitates the removal of a coupling facility from the sysplex, for example, for maintenance purposes. With this option, you can perform the following sequence of tasks:

1. Display information for all allocated structures of the coupling facility.
2. Put the coupling facility into maintenance mode if the MAINT function is available.
3. Rebuild all rebuildable structures on another coupling facility, and delete instances of structures on the target coupling facility that are being duplexed on another coupling facility.

### Notes:

- The scope of the structures that can be rebuilt depends on the release level of the sysplex members.
- INGCF DRAIN rebuilds structures one at a time (SETXCF START,REBUILD,STRNAME=), not globally (SETXCF START,REBUILD,CFNAME=), and always on a coupling facility that is different from the target coupling facility (LOCATION=OTHER).
- Generally, you should be aware that it is XES that performs the actual rebuild. Not all of the factors that XES takes into account when allocating a structure are accessible to SA z/OS. Therefore, a rebuild request for a structure that should be rebuildable according to its condition can fail in certain rare cases.
- Force the deletion of structures that have no active connectors and cannot be rebuilt.
Note: There are structures that you can neither rebuild nor delete with the force action. Included are the structures that have at least one active connector and do not support rebuild. To remove such structures, first disconnect all active connectors, and then delete the structure manually if it is persistent or has persistent connections.

5. When the coupling facility is not in maintenance mode, disconnect the coupling facility from the systems that it is connected to.

6. Inactivate the target coupling facility.

INGCF DRAIN ensures that these actions are performed in the correct order, as specified above.

Actions

The following F-keys are supported:

*REBUILD (F10)
Starts the rebuild of structures that can be rebuilt on another coupling facility. Thus, a rebuild is only initiated for structures whose preference list contains more than one coupling facility.

There are two methods for rebuild, user-managed and system-managed rebuild. User-managed rebuild is supported for all release levels. System-managed rebuild is only available with systems that have been enabled by formatting the CFRM couple data sets with the specification:

ITEM NAME(SMREBLD) NUMBER(1)

System-managed rebuild is only performed when the requirements for user-managed rebuild are not met. This applies, for example, to structures without active connectors.

The REBUILD action also deletes all structure instances on the target coupling facility that are being duplexed on another coupling facility.

Note: The REBUILD action deallocates structures with the condition No active connection exists. See No active connection exists.

*FORCE (F5)
Forces the deallocation of structures with one of the following conditions:

- No connection exists.
- No alternate coupling facility for structure with no active connections.
- No alternate coupling facility for structure with no connections.

This action is only made available after all structures that can be rebuilt have been rebuilt.

*MAINTON (F5)
Puts the coupling facility into maintenance mode.

Note that this function is only available before you start the rebuild process. Once you have started the rebuild process and you want to put the coupling facility into maintenance mode you need to issue the command INGCF MAINT cf_name MODE=ON.

*DRAIN (F4)
Disconnects the coupling facility from its connected systems by setting the sender paths OFFLINE.
This action is only enabled after all structures of the target coupling facility have been removed to another coupling facility or deallocated. Note that structures that have active connectors but do not support rebuild cannot be removed with F10 or F5. They must be deallocated manually before executing this step is enabled.

**SHUTDOWN (F11)**

This action inactivates the coupling facility. It is only made available when all connections between the coupling facility and the systems of the sysplex have been disconnected.

_Note:_ This function key is unavailable when running on a z/OS image that runs under z/VM®.

Note that these actions can only be performed if INGCF DRAIN is issued in full mode. In line mode, only the display function is available.

To avoid performance degradation due to multiple rebuild processes, or unpredictable results due to multiple executions of an action, all actions are locked. Therefore, an action is rejected if any lock exists even if the action does not affect the action currently being performed. Because the action can take a long time it is also executed asynchronously on a dedicated autotask, preventing the operator from being blocked. To check progress, use the refresh function (F9).

**Example not Using MAINTMODE**

In the following example, a coupling facility is drained:

1. All of its structures that can be rebuilt are rebuilt on another coupling facility, and duplexing is stopped.
2. For all structures that have no active connector and cannot be rebuilt deletion is forced.
3. All systems that are connected with the coupling facility are disconnected.
4. The coupling facility is inactivated.

When you issue INGCF with the DRAIN option, you can specify the coupling facility to be drained, for example by entering INGCF DRAIN CF01. If you do not specify a coupling facility name, INGCF displays a selection panel with all coupling facilities that are defined in the sysplex.

The status of the coupling facility (NORMAL) and the authorization type of the operator (ALL) are displayed on the right side of the panel header. The main part of the panel consists of a list containing the structures allocated in CF01 and their conditions. The conditions are classified by color and an asterisk. The asterisk signifies that a structure cannot be rebuilt.

Depending on the availability of the MAINT function, either the two actions, MAINTON with F5 and REBUILD with F10, are enabled, or only REBUILD with F10, is enabled. Pressing F10 calls the confirmation panel of the rebuild process.

After F10 has been pressed on the confirmation panel and the rebuild is complete the command dialog can be refreshed with F9.

One structure could not be rebuilt because no alternate coupling facility is specified in its preference list. The REBUILD action is no longer available. Instead, the FORCE action (F5) is available because the structure that could not be rebuilt has a condition that allows forcing the deallocation of the structure. Pressing F5
calls a confirmation panel similar to that for REBUILD. Pressing F10 on the confirmation panel and refreshing the command dialog after the action has been completed results in an empty panel.

No more structures are allocated in the coupling facility and the coupling facility is not in maintenance mode, so that the coupling facility can be released from the connections with the systems of the sysplex. Consequently, INGCF DRAIN enables the DRAIN action (F4). After completion of that action, the status of the coupling facility changes to DRAINED.

Because the coupling facility is no longer connected to any system, it can be inactivated. After pressing F11 and confirming the action the status of the coupling facility changes to INACTIVE.

Example Using MAINTMODE

In the following example, a coupling facility is drained:

1. The coupling facility is put into maintenance mode.
2. All of its structures that can be rebuilt are rebuilt on another coupling facility, and duplexing is stopped.
3. Deletion is forced for all structures that have no active connector and cannot be rebuilt.
4. The coupling facility is made inactive.

Note: It is no longer necessary to turn the sender paths offline when the coupling facility is in maintenance mode, regardless of whether or not all systems in the sysplex run z/OS 1.9 or later.

When you issue INGCF with the DRAIN option, you can specify the coupling facility to be drained, for example, by entering INGCF DRAIN CF02. If you do not specify a coupling facility name, INGCF displays a selection panel with all coupling facilities that are defined in the sysplex.

The status of the coupling facility (NORMAL) and the authorization type of the operator (ALL) are displayed on the right side of the panel header. The main part of the panel consists of a list containing the structures allocated in CF02 and their conditions. The conditions are classified by color and an asterisk. The asterisk signifies that a structure cannot be rebuilt.

Two actions are enabled, MAINTON with F5 and REBUILD with F10. Pressing F5 calls the confirmation panel for starting the maintenance mode.

After pressing F10 on the confirmation panel, the main panel shows the new status (NORMAL MAINTMODE) and leaves only F10 for the rebuild process. Pressing F10 calls the confirmation panel for the rebuild process.

After pressing F10 on the confirmation panel and the rebuild is complete, you can refresh the command dialog with F9.

One structure could not be rebuilt because no alternate coupling facility is specified in its preference list. The REBUILD action is no longer available. Instead, the FORCE action (F5) is available because the structure that could not be rebuilt has a condition that allows forcing the deallocation of the structure.
Pressing F5 calls a confirmation panel similar to that for REBUILD. Pressing F10 on the confirmation panel and refreshing the command dialog after the action has been completed results in an empty panel and the status of the coupling facility has changed to MAINTMODE.

Because no more structures are allocated in the coupling facility, it can be inactivated. After pressing F11 the status of the coupling facility changes to INACTIVE.

**INGCF ENABLE**

**Purpose**

The ENABLE function of the INGCF command is intended to support the integration AND re-integration of a coupling facility into a sysplex. With this option, you can:

1. Activate the target coupling facility.
2. Connect the systems of the sysplex with the coupling facility.
3. Take the coupling facility out of maintenance mode when it is in this mode and the MAINT function is supported.
4. Switch to another CFRM policy if the target coupling facility is not defined in the active policy and a suitable policy is available.
   A suitable CFRM policy must contain:
   - A definition of the target coupling facility
   - Appropriate definitions for every active coupling facility and every allocated structure
5. Rebuild all structures on the target coupling facility whose preference list starts with this coupling facility, unless this is excluded by other requirements.

INGCF ENABLE ensures that these actions are performed in the correct order, as specified above.

**Actions**

The possible actions and the associated F-keys are:

- **ACTIVATE (F11)**
  This action activates the CFCC (Coupling Facility Control Code) through the BCP Internal Interface by an ACTIVATE command.
  
  **Note:** This function key is unavailable when running on a z/OS image that runs under z/VM.

- **ENABLE (F4)**
  Sets the sender paths of all systems of the sysplex to ONLINE. This action is enabled when the coupling facility is active.

- **MAINTOFF (F5)**
  Takes the coupling facility out of maintenance mode.

- **SWITCH (F5)**
  Switches to another CFRM policy when the target coupling facility is not defined in the active CFRM policy and a suitable policy is available. When there is more than one suitable policy you can choose one of these from a selection panel.
A CFRM policy is suitable when it contains:
- A definition of the target coupling facility
- Definitions for every active coupling facility and every allocated structure

This action is only made available when the target coupling facility is active, but not defined in the current CFRM policy.

**POPULATE (F10)**

Starts a rebuild process by which all structures that have the target coupling facility at the beginning of their preference list but are currently allocated on another coupling facility are allocated on the target coupling facility.

This action requires that the coupling facility be enabled, connected to all members of the sysplex, and defined in the current CFRM policy. The action is offered whenever INGCF ENABLE detects that a structure is not allocated on the target coupling facility although it is the preferred coupling facility of that structure.

**Note:** When you have drained a coupling facility with INGCF DRAIN and then reintegrate it with INGCF ENABLE, be aware that the set of structures that are allocated on the target coupling facility after population can be different from the original set before the draining. Typically, this happens when the original set does not contain exactly those structures that have the target coupling facility at the first position in their preference list.

Note that these actions can only be performed when INGCF ENABLE is called in full mode. In line mode, only the display function is available.

**Example with CF not in MAINTMODE**

In the following example, a coupling facility that has already been activated is reintegrated into the sysplex in two steps:

1. The coupling facility is connected to all systems of the sysplex.
2. All structures that have the target coupling facility as the first coupling facility in their preference list are allocated on the target coupling facility.

If you issue INGCF with the ENABLE option, you can specify the coupling facility to be reintegrated, for example by entering INGCF ENABLE CF02. If you do not specify a coupling facility name, INGCF shows a selection panel with all coupling facilities that are defined in the sysplex.

The selected coupling facility has already been activated manually, therefore its status, as shown on the right of the panel header, is DRAINED. The authorization type of the operator (ALL) is also displayed on the right side of the panel header. The main part of the panel is empty because no structures are allocated in CF02. The only action that is activated is ENABLE with F4. If you press F4 a confirmation panel is displayed.

After pressing F10 on the confirmation panel, the command dialog shows that structures are currently allocated in CF01.

The status has changed to NORMAL, and F10 is enabled for populating the coupling facility. This implies that the target coupling facility is defined in the active CFRM policy.
The structure list contains 10 entries with the condition Structure is currently allocated in CF01. These are the structures that are currently allocated in CF01, but have CF02 in the first position in their preference list.

Pressing F10 and confirming the action populates the coupling facility.

Example with CF in MAINTMODE

In the following example, a coupling facility that has already been activated is reintegrated into the sysplex in two steps:

1. The coupling facility is taken out of maintenance mode.

   **Note:** It is no longer necessary to turn the sender paths offline when the coupling facility is in maintenance mode, regardless of whether or not all systems in the sysplex run z/OS 1.9 or later.

2. All structures that have the target coupling facility as the first coupling facility in their preference list are allocated on the target coupling facility.

If you issue INGCF with the ENABLE option, you can specify the coupling facility to be reintegrated, for example, by entering INGCF ENABLE CF02. If you do not specify a coupling facility name, INGCF shows a selection panel with all coupling facilities that are defined in the sysplex.

The selected coupling facility has already been activated manually, therefore its status, as shown on the right of the panel header in Figure 39 on page 106, is MAINTMODE. The authorization type of the operator (ALL) is also displayed on the right of the panel header. The structure list contains 10 entries with the condition Structure is currently allocated in CF01. These are the structures that are currently allocated in CF01, but have CF02 in the first position in their preference list. The only action that is activated is MAINTOFF with F5.

---

**Figure 38. ENABLE Command Dialog Panel: After Populating**
If you press F5 a confirmation panel is displayed.

After pressing F10 on the confirmation panel, the command dialog shows the new status and a new function key.

The status has changed to NORMAL, and F10 is enabled for populating the coupling facility. This implies that the target coupling facility is defined in the active CFM policy.

Pressing F10 and confirming the action populates the coupling facility.

**INGCF ENABLE**

<table>
<thead>
<tr>
<th>Domain Id</th>
<th>IPSFP</th>
<th>Date</th>
<th>09/29/08</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operator Id</td>
<td>USER1</td>
<td>Time</td>
<td>13:55:52</td>
</tr>
</tbody>
</table>

Coupling Facility = CF02
Status = MAINTMODE
Sysplex = KEYPLEX
Permission = ALL

Structure Condition

- DBNA_GBP2: Structure is currently allocated in CF01.
- DBNA_LOCK1: Structure is currently allocated in CF01.
- HSA_LOG: Structure is currently allocated in CF01.
- IXPLEX_PATH3: Structure is currently allocated in CF01.
- IXPLEX_PATH4: Structure is currently allocated in CF01.
- JES2CKPT1: Structure is currently allocated in CF01.
- LOGGER_STR2: Structure is currently allocated in CF01.
- SYSZwLM_9F092064: Structure is currently allocated in CF01.
- S941EMHQ: Structure is currently allocated in CF01.
- S941MSGQ: Structure is currently allocated in CF01.

Command ===>
F1=Help  F2=End  F3=Return  F5=MaintOFF  F6=Roll
F9=Refresh  F12=Retrieve

**Figure 39. ENABLE Command Dialog Panel: After Issuing Command "INGCF E cf_name"**

If you press F5 a confirmation panel is displayed.

After pressing F10 on the confirmation panel, the command dialog shows the new status and a new function key.

The status has changed to NORMAL, and F10 is enabled for populating the coupling facility. This implies that the target coupling facility is defined in the active CFM policy.

Pressing F10 and confirming the action populates the coupling facility.

**INGCF MAINT**

**Purpose**

The MAINT function of the INGCF command is intended to put a coupling facility into or take it out of maintenance mode.

After the corresponding XCF command has been processed the command issues the INGCF command without any positional parameters.

**Restrictions**

The function requires:
- At least one system in the SA z/OS sysplex must run z/OS 1.9, or later.
- The MAINT function must be supported by all SA z/OS applications that are involved in command processing.

**Example**

In the following example, a coupling facility is put into maintenance mode using the command INGCF M cf_name MODE=ON OUTMODE=LINE:
INGCF PATH

Purpose

The INGCF PATH function displays the sender paths, that is, the paths from the connected systems to the specified coupling facility.

Restrictions

The last sender path of each system can only be set to OFFLINE when no more structures are allocated.

Example

The following command codes are available:

F  Sets the sender path OFFLINE.
N  Sets the sender path ONLINE.

The fields on the command dialog panel display the following information:

- If you have issued INGCF with the PATH parameter, the Coupling Facility field is an input field. To display the path list of another coupling facility specify the name of the coupling facility in this field and press Enter.
- The Allocated Structures field shows the number of allocated structures.
INGCF PATH

- The **Permission** field shows your authorization level.
- The **System** field contains the names of the systems that are connected to the target coupling facility.
- The **CHPID** field shows the IDs of the sender channel paths.
- The **Physical** field shows the status of the sender channel paths.
- The **Logical** field shows the logical status of the paths to that coupling facility.
- The **Type** field shows the type of the sender channel paths.

INGCF STRUCTURE

**Purpose**

The STRUCTURE function of the INGCF displays the allocated structures of a coupling facility. You can initiate a rebuild or deallocation of a selected structure if the conditions for these actions are satisfied.

**Example**

![Structured Panel]

The following command codes are available:

- **D** Displays detail information about the structure.
- **+F** Forces the deallocation of the structure if it has one of the following conditions:
  - No connection exists.
  - No alternate CF for structure with no active connections.
  - No alternate CF for structure with no connections.

  When you try to force the deallocation of a structure that can be rebuilt, an error message is issued.

- **+P** Stops duplexing of the selected structure.
- **+R** Starts the rebuild of the selected structure. Depending on the PENDING status, the automation starts the rebuild with a different LOCATION parameter (PENDING uses the parameter LOCATION=NORMAL, otherwise
INGCF STRUCTURE

LOCATION=OTHER). A rebuild with the parameter LOCATION=OTHER is only initiated for structures whose preference list contains more than one coupling facility.

There are two methods for rebuild, user-managed and system-managed rebuild. User-managed rebuild is supported for all release levels. System-managed rebuild is only available with systems that have been enabled by formatting the CFRM couple data sets with the specification:

```
ITEM NAME(SMREBLD) NUMBER(1)
```

System-managed rebuild is only performed when the requirements for user-managed rebuild are not met. This applies, for example, to structures without active connectors.

INGCF STRUCTURE accepts a rebuild request for structures with the condition 'No active connection exists,' but deallocates them. See "No active connection exists".

$S Starts duplexing of the selected structure.

There are two methods for duplexing, user-managed and system-managed duplexing. User-managed duplexing is supported for all release levels. System-managed duplexing is only available when it has been enabled by formatting the CFRM couple data sets with the specification

```
ITEM NAME(SMDUPLEX) NUMBER(1)
```

System-managed duplexing is only performed when the requirements for user-managed duplexing are not met. This applies, for example, to structures without active connectors.

Starting the duplex rebuild of a structure requires at least the policy entry allowing the duplex rebuild of the structure. If there is no entry the duplex rebuild is disabled. The other requirements depend on the type of the duplex rebuild. When all connectors to a structure allow user-managed duplex rebuild, this type takes precedence over system-managed duplex rebuild. However, user-managed rebuild also requires at least one active connector. Thus, when the operator starts the duplex rebuild for a structure allowing user-managed duplex rebuild as well as system-managed rebuild but without having active connectors, XCF tries to initiate a system-managed duplex rebuild.

System-managed duplex rebuild has the following requirements:

- System-managed rebuild must be supported by all connectors.
- The structure must be allocated in a coupling facility supporting system-managed duplexing and another coupling facility supporting system-managed duplexing must be defined in its preference list.
- The CFRM couple data set must support system-managed duplex rebuild and the structure must not have a policy change pending.
- The structure must be defined in the active CFRM policy when any connection state is not active.

The fields on the command dialog panel display the following information:

- The Coupling Facility field is an input field, if you have specified INGCF with the STR parameter. To display the structure list of another coupling facility, specify the name of the coupling facility in this field and press Enter.
- The Include Condition field is an input field. By specifying Yes or No in this field you determine whether or not the conditions of the structures are displayed in the Structure column.
**INGCF STRUCTURE**

- The **Permission** field shows your authorization level. There are two possible values, ALL and DISPLAY. DISPLAY signifies that you can only use the display functions. ALL signifies that you can also rebuild and delete structures.
- You can specify an action code before every structure entry. The codes you can enter depend on your authorization level.
- The **Structure** column shows the names of the structures.
- If the **P** column contains the letter *P*, this indicates that policy changes are pending for the structure.

A structure has policy changes pending if it was allocated at the time of a CFRM policy switch, but XES could not bring the switch into effect for that structure. One reason for a pending policy change is that the old and the new policy define the structure differently, for example, with different preference lists.

- The **D** column indicates the type of duplexing that is possible. The following values are possible:
  - **U**: User-managed duplexing
  - **S**: System-managed duplexing
  - **B**: User-managed and system-managed duplexing
- The **Condition** column shows the status of the structures. You can switch the display of the conditions on and off with the **Include Condition** field.

---

**INGCFG**

**Purpose**

The INGCFG command may be used with the CLEAR_HISTORY parameter to clear the workitem history for the given resource(s).

**Syntax**

```
>>-INGCFG-CLEAR_HISTORY, resource
```

**Parameters**

resource

Specifies the name of the resources to have workitems cleared. The format is `name[/type[/system]]`. It can be a list of names. The resource names must be separated by a blank. Asterisks (*) and a percentage sign (%) can be used as wildcard characters.

---

**INGCFL**

**Purpose**

The INGCFL routine supports line mode for INGCF other than display capability. For further information see “INGCFL”.

Support of the maintenance mode of a coupling facility depends on the function that is specified:
ACTIVATE, ENABLE, and POPULATE automatically handle the maintenance mode of the coupling facility. However, each command will fail when the coupling facility is in maintenance mode but the mode is not supported by SA z/OS (see Restrictions).

DEACTIVATE and DRAIN put the coupling facility into maintenance mode when possible. Otherwise, both functions proceed with setting the sender paths offline after the rebuild has been performed.

REBUILD does not make use of the maintenance mode capability.

Syntax

```
+---+-------------------+-------------------+
|   | INGCFL             |                  |
+---+-------------------+-------------------+
      |                    |                   |
      | CF_name            | RESP=SYNC         |
      |                    | RESP=ASYNC        |
      | TARGET =           |                   |
      |   system           |                   |
      |     domain         |                   |
      |       sysplex      |                   |
      +-------------------+-------------------+
```

Parameters

- **Activate**
  Activates the coupling facility.

- **DEactivate**
  Deactivates the coupling facility after performing DRAIN.

- **DRain**
  Performs REBUILD, then puts the coupling facility into maintenance mode if possible. Otherwise, it sets the sender paths to OFFLINE.

- **Enable**
  Performs ACTIVATE, then takes the coupling facility out of maintenance mode if it is in this mode. Finally, it sets the sender paths to ONLINE.

- **Populate**
  Starts the populate process of the coupling facility after performing ENABLE.

- **Rebuild**
  Starts the rebuild process of the coupling facility.

- **CF_name**
  The name of the target coupling facility for the specified function.

- **RESP**
  Specifies whether the final result is returned synchronously as a return code or asynchronously as a message (default: synchronous response).

- **TARGET**
  Specifies the system where the command is executed (default: local system).

**Note:** The real activation and deactivation of a coupling facility are unavailable when running on a z/OS image that runs under z/VM.
Return Codes

0  The command completed normally.
1  An initialization error occurred. Check the netlog for explanatory messages.
3  The slave routine failed. Check the netlog for explanatory messages. If you are still unable to detect the reason for failure, attempt a rerun after turning on debugging for the INGRVX92 slave routine.
4  The lock is unavailable to update the coupling facilities.
5  The authorization check failed for function. Check the netlog for explanatory messages.
6  The status of the coupling facility is not as expected for the current request. Running with GDPS debugging should provide a listing of status changes that have taken place. This return code does not necessarily indicate that the request failed.
7  The permission check has failed for the POPULATE request.

Restrictions and Limitations

The real activation and deactivation of a coupling facility are unavailable when running on a z/OS image that runs under z/VM.

INGCICS

Purpose

The INGCICS command lets you:
• Issue any console-enabled CICS transaction.
• Broadcast messages to all or selected CICS users.
• Issue a list of defined transactions and view the output.
• Display the output of CICS transactions in fullscreen or pipeable line mode.

Syntax

```
INGCICS resource
   command_specification
       broadcast_specification
       info_specification
   TARGET= system_name domain_ID sysplex_name
       OUTMODE= LINE AUTO NETLOG
```

command_specification:

```
REQ=CMD
   CMD=cics_transaction
   MSG=message ID
```
info_specification:

Parameters

resource

The resource specifies the name of the CICS subsystem that the command will be issued to. The format is subsystem[/APL[/system]], where the subsystem is the specific name of the desired CICS subsystem and system is the name of the system that the CICS is running on.

Note: You need not specify the fully qualified name of the resource. However, the command requires enough information to execute the request against a unique CICS subsystem. The command does not support issuing requests to multiple CICS subsystems at the same time.

REQ

Specifies the request to be issued to the CICS subsystem. It can be one of the following:

CMD  Issues a CICS transaction and displays the results.

BROADCAST  
Issues a CICS broadcast to all or selected users.

INFO  Issues a set of predefined commands and displays the results. The list of commands is obtained from the CICSINFO user message policy of the subsystem specified by the resource name.

CMD

Specifies the CICS transaction and its parameters to be executed for a REQ=CMD type request. If the command and parameters contain spaces, enclose the command in quotation marks (single or double). For CEMT transactions INGCICS waits up to twice the period that is defined by the system wait time parameter for a result.

MSG

Specifies the message to be sent to all or selected CICS terminals or users for the REQ=BROADCAST request.

For REQ=CMD the MSG command can be used to contain a list of words that will be used to end processing if they are received anywhere in the output results. The words will normally be a required message ID and when entered INGCICS waits up to the period that is defined by the system wait time parameter for a result. If no words are entered, CMD is processed until any result is received.

ROUTE

Specifies the CICS routing information that is used to select the CICS terminals or users are to receive the message for a REQ=BROADCAST request. The contents of the ROUTE= parameter are the parameters to be supplied to the CMSG transaction to route messages to CICS terminals or users. See the CMSG transaction for the details of the parameters for that transaction.
INGCICS

TARGET
For information on the TARGET parameter, see “TARGET Parameter” on page 10.

OUTMODE
For information on the OUTMODE parameter, see “OUTMODE Parameter” on page 12.

Restrictions and Limitations
To use the INGCICS command system operations must be initialized. CICS subsystems must be enabled for system console commands using the MVS MODIFY command.

Usage
The INGCICS command operates sysplexwide. For an overview see “Overview of Commands that Operate Sysplexwide” on page 9.

Examples
If you enter INGCICS REQ=CMD a panel similar to Figure 43 is displayed.

Figure 43. INGCICS REQ=CMD Command Dialog Panel

- The Resource field shows the name of the CICS subsystem to be used for issuing the requests. The format is name/type[/system]. Wildcards are supported.
- The Target field shows the name of the system (system name, domain ID, or sysplex name) that the command should be routed to. Specifying this is only necessary if the resources are not part of the local sysplex.
• The Request field shows the request to be carried out. It can be CMD, BROADCAST or INFO.

• The CICS Transaction field shows the CICS transaction to be executed on the CICS subsystem specified by the resource field.

• The CICS Route field shows the routing parameters for the CMSG transaction when the request is BROADCAST.

• The CICS Message field shows the message to be sent to users or terminals for the BROADCAST request.

• The output from the CICS transaction appears in the blank area after the CICS Message field.

If you specify INGCICS REQ=BROADCAST a panel similar to Figure 44 is displayed.

If you specify INGCICS REQ=INFO a panel similar to Figure 45 on page 116 is displayed.
INGDB2

Purpose

The INGDB2 utility lets you:
- Start or stop a table space
- Terminate active threads
- Inform TSO users that their thread is about to be terminated
- Check for indoubt threads

Syntax

```
INGDB2 TABLE subsystem TERM parm

CHECK_INDOUBT
INFORM_TSO

TARGET system_name domain_ID sysplex_name
```

Parameters

**TABLE**

The TABLE parameter requests to start or stop a table space.
For DB2® table space stop, certain active threads that use the specified table space are terminated. These include REMOVE, DB2CALL, BATCH and TSO. TSO users are issued with a message informing them that their thread is about to be terminated before actual thread termination.

**TERM**

The TERM parameter requests the termination of all active threads for a DB2 subsystem. These include REMOTE, DB2CALL, BATCH, TSO and all remaining threads.

All threads are discovered using the DIS THD() command and canceled by token using CAN THD(tokenid). UTILITY threads are treated differently: after having been discovered using DIS UTIL() and checked not to be COPY, REORG, REPAIR or LOAD, they are terminated using the TERM UTIL() command. UTILITY threads are ignored during the cancel-threads-by-token process.

TSO users that are using threads that cannot be stopped are canceled.

**CHECK_INDOUBT**

This parameter requests a check for indoubt threads.

**INFORM_TSO**

With this parameter, TSO users with an active stoppable thread are informed that their thread is about to be terminated.

**parm**

Specifies a positional parameter string.

When INGDB2 is called with TABLE as the first parameter, one or three values are expected at this position in the parameter sequence, as follows:

- **parm1** START or STOP to start or stop a table space.
- **parm2** A database name. This is not required if **parm1** is START.
- **parm3** A table space name. This is not required if **parm1** is START.

When all three parameters are passed to INGDB2, the START or STOP request applies to the specified table space.

If only START is passed to INGDB2, a start command is issued for all table spaces that are currently in the status LPL or GRECP. Table spaces of database DSNDB01 are started before those of database DSNDB06. After this, the table spaces of all other databases are started.

When requesting a STOP for a specified tablespace, the following processing cycles are done:

1. The command STO DB(*dbname*) SPACENAM(*tsname*) is issued.
   - If the specified table space is then still used by any DDF threads or BATCH jobs, these processes are canceled.
   - TSO users of threads still using the table space are notified that the table space is to be stopped and they are requested to stop using it.
2. If the table space is still used by TSO users, these users are canceled. After one minute, a message is send to the cancelled users to inform them at logon, that they were canceled due to the stop of a table space.
3. If the table space is still used by any processes, a message is issued listing the name and type of all these processes.
The delay time between the processing cycles is determined by the STOP Tablespace Delay option in the DB2 CONTROL policy item of the DB2 master. The default value is 2 minutes.

**TARGET**

For information on the target parameter, see “The TARGET Parameter” in IBM Tivoli System Automation for z/OS Operator’s Commands.

**Version**

When you specify INGDB2 TABLE,&SUBSAPPL,START [dbname tsname] as a command in the automation policy that is to be issued in response to a received message, code definitions for message ID DATABASE are used to decide whether any databases or table spaces are excluded from recovery.

To exclude databases or table spaces from recovery, specify a code definition as follows with IGNORE or EXCLUDE as the Value Returned.

<table>
<thead>
<tr>
<th>Code1</th>
<th>Code2</th>
<th>Code3</th>
<th>Value Returned</th>
</tr>
</thead>
<tbody>
<tr>
<td>database</td>
<td>tablespace</td>
<td></td>
<td>IGNORE</td>
</tr>
</tbody>
</table>

**Restrictions and Limitations**

The INGDB2 utility can only be used when:

- SA z/OS is initialized
- The DB2 subsystem is defined to SA z/OS as application of category DB2 and subcategory MSTR
- The status of the DB2 subsystem is UP, but for TERM and INFORM_TSO the status can be AUTOTERM to allow for SHUTNORM passes.

For non-utility threads, only those with a non-zero token can be canceled with INGDB2 TERM.

**Return Codes**

- 0  Function completed successfully
- 1  Indoubt threads were found when being called with the CHECK_INDOUBT parameter

**Note:** When your INGDB2 CHECK_INDOUBT command is part of your DB2 shutdown commands in the policy and you have return code checking enabled (specified “*” in the AutoFnl field), System Automation does not issue any other shutdown commands. The Automation Status of the DB2 is set to PROBLEM. Manual intervention is required.

- 4  Incorrect parameters were used in the call
- 5  Timeout or other error occurred (for example, the DB2 subsystem is not UP).
- 6  SA z/OS initialization incomplete, unable to process command request

**Examples**

To terminate threads for a DB2 subsystem called DB2P, enter the following command:

INGDB2 TERM DB2P
To start a table space where the DB2 subsystem is DB2P, the database name is DB2PDBN and the table space name is DB2PTSN, enter the following command:

```
INGDB2 TABLE DB2P START DB2PDBN DB2PTSN
```

---

**INGDLA**

**Purpose**

The INGDLA command invokes the SA z/OS Discovery Library Adapter to discover SA z/OS configuration data, which is primarily the SA z/OS resources and their relationships. It then saves the data in an Identity Markup Language (IdML) book, which can be forwarded to a system that is hosting a Tivoli Application Discovery Dependency Manager (TADDM).

The discovery scope of the INGDLA command is the entire sysplex, that is, all the systems that the SA z/OS automation manager controls.

You can use the INGDLA command, for example, after a major configuration change, or periodically driven by a NetView timer.

**Syntax**

```
/SM590000/

INGDLA

CODEPAGE=nnnn

ODSN=dsname

MEMBER=INGBOOK

ORGNAME=orgname

ENCODING=nnnn

HOSTNAME=name

WAIT=nnn

CHECKSUM=YES

CONTACTINFO=data

/SM590000/
```

**Parameters**

**CODEPAGE**

Specifies the encoding codepage — this is the one that NetView uses. The default is 1047, unless it is overwritten by the installation defaults (using the INGCNTL command, see *IBM Tivoli System Automation for z/OS Programmer’s Reference*).

You must specify this parameter if you are running with a different codepage. Failure to do so will result in the generation and downloading of a corrupt Identity Markup Language (IdML) book that Tivoli Application Discovery Dependency Manager (TADDM) cannot load.

**ODSN**

Specifies the name of the output data set. The data set must be a pre-allocated (catalogued) PDS with attribute VB=3000. The name must be fully qualified, with or without surrounding quotes. The user ID that NetView is running under must have UPDATE access to it. With the INGCNTL command, this parameter can be preset (see INGCNTL in *IBM Tivoli System Automation for z/OS Programmer’s Reference*).

**MEMBER**

Specifies the name of the member that will contain the IdML data. The
default is INGBOOK unless it is overwritten with installation defaults (see INGCNTL in IBM Tivoli System Automation for z/OS Programmer’s Reference).

Reserved member name is @CHCKSUM.

**ORGNAME**
Specifies the name of the organization. The default is to take the default name from the IBM Tivoli Change and Configuration Management Database (CCMDB) unless it is overwritten by the installation defaults (see INGCNTL in IBM Tivoli System Automation for z/OS Programmer’s Reference).

Place quotes around the organization name in order to maintain mixed case names.

**ENCODING**
Specifies the encoding option. Valid values are EBCDIC, ASCII and UTF-8. The default is UTF-8 unless overwritten by the installation defaults (see INGCNTL in IBM Tivoli System Automation for z/OS Programmer’s Reference).

**HOSTNAME**
Specifies the name of the host of the management software system (that is, SA z/OS). It is used to address SA z/OS. If specified, it takes precedence over a discovered host name.

**WAIT**
Specifies the maximum number of seconds to wait for the automation manager query to respond. The maximum time interval is 999 seconds. The default is 30 seconds.

**CHECKSUM**
Specifies whether to skip generating the SA z/OS DLA book if the checksum matches:

**YES**
Skip generation if the checksum matches. This is the default.

**NO**
Ignore checksum results.

**CONTACTINFO**
Provides details of ports and security keys that are needed to establish a session with NetView over TCP/IP when used in conjunction with the host name.

**Return Codes**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful completion.</td>
</tr>
<tr>
<td>8</td>
<td>Error occurred, such as a bad return code received from the automation manager.</td>
</tr>
<tr>
<td>12</td>
<td>Bad input parameters or allocation failed.</td>
</tr>
<tr>
<td>16</td>
<td>Fatal error.</td>
</tr>
</tbody>
</table>

**Usage**

The INGDLA command can be invoked from the NetView command line on any system in the sysplex.

The INGDLA command operates sysplexwide. For an overview see “Overview of Commands that Operate Sysplexwide” on page 9.
Examples

If you enter ingdla at the command line, SA z/OS displays a report that informs you of the progress of the DLA.

```plaintext
ING502I SA z/OS Discovery Library Adapter started 14:18:05 29 NOV 2007
ING501I DRIVING SA Z/OS DISCOVERY
ING501I .QUERYING RESOURCES
ING501I .QUERYING RELATIONSHIP DATA
ING501I .FOR RESOURCES */APL/SAT1
ING501I .FOR RESOURCES */APL/SAT2
ING501I .FOR RESOURCES */APL/SAT3
ING501I .FOR RESOURCES */APL/SAT4
ING501I .FOR RESOURCES */APG/SAT1
ING501I .FOR RESOURCES */APG/SAT2
ING501I .FOR RESOURCES */APG/SAT3
ING501I .FOR RESOURCES */APG/SAT4
ING501I .FOR RESOURCES */APG
ING501I .FOR RESOURCES */MTR/
ING501I .OBTAINING AGENT INFORMATION FOR SAT1
ING501I PROCESSING CHECKSUMS
ING501I COMPOSING SA Z/OS DISCOVERY BOOK
ING501I WRITING BOOK: INGBOOK
ING501I .COMPOSING RESOURCE DEFINITIONS
ING501I .COMPOSING RESOURCE RELATIONSHIPS
ING501I .COMPOSING DELAYED RELATIONSHIPS
ING501I .FINISHED WRITING BOOK: INGBOOK
ING501I UPDATING CHECKSUM DATA
ING504I Checksum data updated
ING503I SA z/OS Discovery Library Adapter completed 14:18:17 29 NOV 2007, RC=0
```

Figure 46. SA z/OS Discovery Library Adapter Report

INGEVENT

Purpose

The INGEVENT command sets or resets a particular event for all affected resources in a sysplex, or for specific resources.

Syntax

```plaintext
INGEVENT event REQ= SET [RESET|UNSET] RESOURCE=resource
```

TARGET= system_name | domain_ID | sysplex_name

```plaintext
WAIT=YES
WAIT=NO
WAIT=nnn
```

Parameters

**event**

Specifies the name of the event.

**REQ**

Specifies the request to be performed. It can have the following values:

**SET**

Sets the event for all resources having the event defined.
**INGEVENT**

**RESET|UNSET**  
Resets or unsets the event for all resources having the event defined.  
Note that RESET and UNSET are synonyms.

**RESOURCE**  
Specifies the name of the resource where INGEVENT should be effective. It can be abbreviated to RES. The format is name/type[/system]

**TARGET**  
For information on the TARGET parameter, see “TARGET Parameter” on page 10.

**WAIT**  
Specifies whether or not to wait until the request is complete. The default is YES.  

\[ \text{nnn} \] is the number of seconds to wait before giving up and reporting that a timeout has occurred. The maximum time interval is 999 seconds.

**Restrictions and Limitations**

Wildcards are not supported.

**Usage**

The INGEVENT command operates sysplexwide. For an overview see “Overview of Commands that Operate Sysplexwide” on page 9.

**Examples**

To set an event for all resources that are linked to that event, type the following:  

\[ \text{INGEVENT B000P100 REQ=SET} \]

The command response is as follows:  

\[ \text{IPUFA AOF442I SET EVENT B000P100 DONE FOR RESOURCE */*/* ON AOCPLEX. -} \]
\[ \text{FROM=AOCA} \]
\[ \text{IPUFA AOF099I FUNCTION COMPLETED} \]

**INGFILT**

**Purpose**

The INGFILT command restricts the amount of information shown in the display. Use INGFILT to set a default filter for the operator's logon. If an asterisk (*) is specified, the filter is reset.

**Syntax**

\[ \text{INGFILT [REQ=SET | REQ=GET]} \]
filter_criteria:

- RESOURCE=
  - resource_name

- OBSERVED=
  - status

- DESIRED=
  - status

- AUTOSTAT=
  - status

- COMPOUND=
  - status

- AUTOFLAG= YES
  - NO
Parameters

**NETVASIS**
Prefix the INGFILT command with NETVASIS if you want to pass the description text in lower or mixed case.

**REQ**
Specifies the type of the request. The type can be:

**SET**
Sets new filter settings. This is the default. If no filter parameter is specified when the command is invoked, the appropriate value of the current filter settings is taken.

**GET**
Returns the current filter settings. Any filter parameter that is specified when the command is invoked will be added to the current filter settings.

**filter_criteria**
The filter criteria to be applied prior to displaying the data.

**RESOURCE**
Specifies the names of the resources to be displayed. The format is *name/type[/system]*. The resource names must be separated by a blank. Asterisks (*) can be used as wildcard characters.

**OBSERVED**
Specifies the observed statuses to be displayed. The statuses must be separated by a blank. It can be abbreviated, for example, to AV for available. You can also specify an asterisk (*) to reset the current filter setting, for example, INGFILT OBSERVED=*.

If '^' or '\' is used, all statuses except the ones you specify are displayed.

**DESIRED**
Specifies the desired statuses to be displayed. The statuses must be separated by a blank. It can be abbreviated, for example, to AV for available. You can also specify an asterisk (*) to reset the current filter setting, for example, INGFILT DESIRED=*.

If '^' or '\' is used, all statuses except the ones you specify are displayed.

**AUTOSTAT**
Specifies the automation status to be displayed. The statuses must be separated by a blank. It can be abbreviated, for example, to ID for idle. You can also specify an asterisk (*) to reset the current filter setting, for example, INGFILT AUTOSTAT=*.

If '^' or '\' is used, all statuses except the ones you specify are displayed.

**COMPOUND**
Specifies the compound status. The statuses must be separated by a blank. It can be abbreviated, for example, to SA for satisfactory. You can also specify an asterisk (*) to reset the current filter setting, for example, INGFILT COMPOUND=*.

If '^' or '\' is used, all statuses except the ones you specify are displayed.

**AUTOFLAG**
Specifies the automation flag to be displayed. It can be either YES or NO, and can be abbreviated. You can also specify an asterisk (*) to reset the current filter setting, for example, INGFILT AUTOFLAG=*.
INGFILT

CATEGORY
Specifies the IBM-defined or user-defined categories that the resource belongs to. More than one value can be specified. If an asterisk (*) is specified, the filter is reset (the filter previously set by the INGFILT command).

SUBCAT
Specifies the IBM-defined or user-defined subcategories of the resource. More than one value can be specified. The subcategory can contain wildcard characters. An asterisk (*) matches a string of arbitrary length and a percentage sign (%) matches a single character. If an asterisk (*) is specified, the filter is reset (the filter previously set by the INGFILT command). For compatibility reasons keyword SUBTYPE is still accepted.

GROUPTYPE
Specifies the type (nature) of the resource group. More than one value can be specified. You can also specify an asterisk (*) to reset the current filter setting, for example, INGFILT GROUPTYPE=*.

DESCR
Specifies the text string as a filter. The text can contain wildcards. An asterisk (*) matches a string of arbitrary length and a percentage sign (%) matches a single character. You can also specify an asterisk (*) to reset the current filter setting. The DESCR parameter is case-sensitive. The text string must be enclosed in single or double quotation marks or parentheses() to maintain the case-sensitivity of the entry.

HEALTH
Specifies the desired health statuses to be displayed. The statuses must be separated by a blank. It can be abbreviated, for example, to NO for normal. You can also specify an asterisk (*) to reset the current filter setting, for example, INGFILT HEALTH=*.
If ‘^’ or ‘\’ is used, all statuses except the ones you specify are displayed.

JOBNAME
The jobname assigned to the resource. More than one jobname can be specified. Wildcards are supported. You can also specify an asterisk (*) to reset the current filter setting.

RUNTOKEN
The runtoken assigned to the resource. More than one runtoken can be specified. Wildcards are supported. You can also specify an asterisk (*) to reset the current filter setting.

PGNAME
The pacing gate assigned to the resource. More than one pacing gate can be specified. Wildcards are supported. You can also specify an asterisk (*) to reset the current filter setting.

OUTMODE
For information on the OUTMODE parameter, see “OUTMODE Parameter” on page 12.

Restrictions and Limitations
None.
Usage

The INGFILT command can also be used when SA z/OS is not initialized.

The INGFILT command lets you set filter options at operator logon from within the operator's initial REXX script.

Examples

If you enter ingfilt a panel similar to Figure 47 is displayed.

![INGFILT Command Dialog Panel](image)

This panel shows the current INGLIST filters. The filter settings are kept in task globals, so when you next run INGLIST they will be used again. Press PF9 to save the currently displayed filters into these globals.

You can specify one or more of the following filters:

**Resources**
This is the list of resources in name/type/system format. You can also specify a wildcard.

**Observed status**
This is the list of observed statuses. You can abbreviate the status, for example, "av" for Available. Any resources whose observed status is in the list is eligible for the display.

**Desired status**
This is the list of desired statuses. You can abbreviate the status. Any resources whose desired status is in the list is eligible for the display.

**Automation status**
This is the list of automation statuses. You can abbreviate the status. Any resource whose automation status is in the list is eligible for the display.
**Compound status**
This is the list of compound statuses. You can abbreviate the status. Any resources whose compound status is in the list is eligible for the display.

**Health status**
This is the list of health statuses. You can abbreviate the status. Any resource whose health status is in the list is eligible for the display.

**Automation flag**
This is the automation flag for the resource. Specify either YES or NO.

**Category**
This is the list of IBM-defined or user-defined categories that the resource belongs to. IBM defined-categories are for example CICS, IMS, DB2, OPC, USS, IMAGE, JES2, JES3, INFOSPHERE, LIFELINE, MQ or ITM.

**Subcategory**
This is the list of IBM-defined or user-defined subcategories that the resource belongs to.

For IBM-defined categories the following values might be used:
- CICS: AOR TOR FOR DOR QOR CMAS WUI
- DB2: MSTR SPAS IRLM DBM1 DIST WLMS ADMT
- IMS: CTL TP DBRC DLS FP BMP FDR
- ITM: TEMA, TEMS
- LIFELINE: ADVISOR AGENT
- OPC: CONTROLLER TRACKER SERVER DATASERVER

**Group type**
This is the type (nature) of the resource group. You can specify more than one group type.

**Job Name**
The job name assigned to the resource. You can specify more than one job name. Wildcards are supported.

**Description**
The description of the resource. Wildcards are supported.

**Runtoken(s)**
The runtoken assigned to the resource. You can specify more than one runtoken. Wildcards are supported.

**Pacing gate(s)**
The pacing gate assigned to the resource. You can specify more than one pacing gate. Wildcards are supported.

In order to be eligible for the display, a resource must match all filter criteria. An asterisk indicates that the filter has not been set.

Use the PF4 key to clear the currently established filter settings. The filter will contain an asterisk, meaning that the filter not set, or a blank.

Use the PF5 key to go back to the currently established filter settings.

**Note:** If INGFILT was called from INGLIST, the filters that are displayed take effect when you press Enter. If you press PF9 first, the filters will be saved.
Purpose

The INGGROUP command displays the members of a group and their settings. INGGROUP can also be used to:

- Move an application from one system to another by terminating the application and starting it on a system of your choice.
- Activate or deactivate a group.
- Adjust the availability or satisfactory target (server group only).
- Initiate or cancel the rolling recycle of a group.

Syntax

```
/SM590000/SM590000
/INGGROUP group_name
ACTION options
WAIT=YES
WAIT=NO
WAIT=nnn

/SM590000
/TARGET system_name
domain_id
sysplex_name
OUTMODE LINE AUTO NETLOG
```

ACTION options:

- ACTION= EXCLUDE
- SYSTEMS=(sysname)
- AVOID
- INCLUDE
- ACTION= ACTIVATE
- PACIFY
- ACTION=MEMBERS
- ACTION=POLICY
- GROUPTYPE=(grouptype)
- ACTION=OVERRIDES
- GROUPTYPE=(grouptype)
- ACTION=RESET
- ACTION=DEFAULT
- ACTION=RECYCLE
- ACTION=CANCEL
- ACTION=ADJUST
- ADJUST options
- ACTION=CANCEL
- ADJUST options
- ACTION=ADJUST
```

ADJUST options:
Parameters

*group_name*

This is a list of application groups, possibly including wildcards, that are to be the target of the command.

- For an ADJUST action specify only a single application group.
- For an AVOID/INCLUDE/EXCLUDE action, you may specify several application groups.
- For an ACTIVATE/PACIFY action, you may specify several application groups.
- For a MEMBERS action, you may specify only a single application group.
- For the RECYCLE/CANCEL action, you may specify several application groups.

**ACTION**

Indicates that you want to remove movable members from a system (EXCLUDE), prohibit the manager from activating movable members on a system (AVOID) or that you want to undo either of the other two actions (INCLUDE).

**EXCLUDE**

Removes resources immediately from the excluded systems. This acts to disruptively force resources from a system.

**AVOID**

No new resources will be moved to the system and those that are there will be removed as scheduled service outages permit. This would be used to gradually (and non-disruptively) remove resources from a system in preparation for an Exclude at a later point in time (mainly to reduce the disruption caused by the Exclude). Note, however, that if there is no other way to maintain or restore application availability, groups will choose to ignore avoidance policies.

**INCLUDE**

The movement of resources to the system depends on the relative preferences of the resources on that system and on other systems. It is possible that the resources will not be moved to that system until the next scheduled service outage.

**ACTIVATE**

Changes the behavior of a group to active. If used on group that is already active there is no effect. When a group becomes active it will distribute votes to all of its non-passive members reflecting its nature and goals. Making a group active may cause some of its members to be stopped or started.

**PACIFY**

Changes the behavior of a group to passive. If used on group that is already passive there is no effect. When a group becomes passive, it withdraws all votes that it has propagated to its members. Making a
group passive may cause some of its members to either start or stop, depending upon the votes that it had propagated.

**MEMBERS**
Displays the group members and their settings.

**POLICY**
Displays the policies for the specified resource groups.

**OVERRIDES**
Displays the overrides for the specified resource groups that have been made with the INGGROUP command.

**RESET**
Sets the preference assigned to the members of the application group to its initial value. Applies only to MOVE and SERVER groups.

**DEFAULT**
Sets the following group policies to the values that are defined in the policy database:
- Availability and satisfactory target
- Preference value of group members
- Group passive attribute

**RECYCLE**
Causes a rolling recycle to be initiated for each selected server or move group.

*Note:* If you specify more than one group and any of them have common members, only one of the groups can be involved in a complete rolling recycle. The other groups are flagged as being involved in a partial recycle, that is, some of their members (possibly all of them) are being or have been recycled.

Specify CHUNK to indicate the number of server group members to recycle in parallel. Refer to [CHUNK](#) for further details. If the systems in a Server group all have the same preference value, the system that is active after a rolling recycle has completed may not be the same one as before the rolling recycle. Thus, triggering a rolling recycle always results in a move occurring. If you set the preference of the original system more than 250 points higher than that of any other system, the rolling recycle is followed by a second move to return the application to that system. This leads to an increase in the application down time.

*Note:* Preference values do not influence the order in which resources are recycled. The resources of the group are recycled from last to first as they are displayed in the list of group members.

**CANCEL**
Causes any rolling recycles amongst the selected groups to be canceled.

**ADJUST**
Indicates that you want to temporarily adjust a group's policy.

**SYSTEMS**
Is the list of systems to be excluded or included. If no groups are specified, all application groups with resources in these systems are affected.

This parameter must be specified for INCLUDE, AVOID and EXCLUDE actions.
INGGROUP

GROUPTYPE
   Specifies the type (nature) of the resource group. More than one value can be specified.

CHUNK
   Indicates the number of server group members to recycle in parallel.

   **Note:** SA z/OS adjusts the chunk size to ensure that at least one group member is active at any time to avoid an application outage. This is the case when the chunk size is greater or equal to the number of active group members.

AVTGT
   Specifies the availability target that is to be set for the group. The value, \( n \), is a positive number. It specifies the number of group members that should be made available when the group has to be available.

   You can also specify \( * \), which resets the AVTGT parameter to the value that is specified in the policy database.

   This parameter may only be specified for an ADJUST action.

   For details of how the AVTGT changes apply, see SATTGT below.

SATTTGT
   Specifies the number of members in the group that must be active before the automation manager treats the group as being in a satisfactory state. The value must be a positive, decimal number. The parameter will be ignored for groups other than server groups.

   You can also specify \( * \), which resets the SATTGT parameter to the value specified in the policy database.

   This parameter may only be specified for the ADJUST action.

   Changes to target values, as set by the AVTGT and SATTGT parameters, apply in the following manner:
   - The Availability and Satisfactory Target values in the automation policy are not affected.
   - An adjustment value is calculated that, when applied to the corresponding target value in the automation policy, produces the "result" value. (See also the Application Group Resources Policy in *IBM Tivoli System Automation for z/OS Defining Automation Policy*).
   - The "result" target value is then used by the automation when a target value is required.
   - The adjustment value, rather than the "result" value, is preserved when performing an automation manager HOT restart or an INGAMS REFRESH if the target is defined as "ALL" in the automation policy. Alternatively, if the target value is explicitly defined to be numerically equal to the "Number of selected Resources" in the automation policy, the "result" value is preserved rather than the adjustment.

MEMBERS
   Specifies a list of group members that the preference value should be set for. Wildcards may not be used.

   This parameter may only be specified on an ADJUST action.

PREF
   Specifies a preference value that is to be used for each specified member. The number of values specified should be the same as the number of members.
specified. Each preference is an integer value that replaces the member’s current preference value. You can also specify *, which resets the PREF parameter to the value that is specified in the policy database.

Preference values are used to determine which members in the group are selected to make the group active.

This parameter may only be specified on an ADJUST action when a list of members has been specified.

Changes to preference values, as set by the PREF parameter, apply in the following manner:

- The preference value in the automation policy is not affected.
- An adjustment value is calculated that, when applied to the preference value in the automation policy, produces the "result" value.
- The "result" preference value is then used by the automation when a preference value is required.
- The adjustment value, rather than the "result" value, is preserved when performing an automation manager HOT restart or an INGAMS REFRESH, but is lost during WARM and COLD restarts.

If an adjustment was made via the INGGROUP or INGMOVE command and to the base preference via the customization dialogs, the current effective preference and the new base preference will be taken to calculate a new adjusted preference. As a result, the old effective preference will be preserved.

**WAIT**

Specifies whether or not to wait until the request is complete. The default is YES.

*nnn* is the number of seconds to wait before giving up and reporting that a timeout has occurred. The maximum time interval is 999 seconds.

**TARGET**

For information on the TARGET parameter, see “TARGET Parameter” on page 10.

**OUTMODE**

For information on the OUTMODE parameter, see “OUTMODE Parameter” on page 12.

### Restrictions and Limitations

None.

### Security Considerations

The INGGROUP command supports resource level security. If turned on, the following profile in class SYSAUTO are checked:

<table>
<thead>
<tr>
<th>Profile</th>
<th>Authority</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGT.sysplex.xcfgrp.RES.resname.restype[.ressys]</td>
<td>UPDATE</td>
<td>When action option is RECYCLE, CANCEL, RESET, DEFAULT, EXCLUDE, AVOID or INCLUDE</td>
</tr>
<tr>
<td></td>
<td>CONTROL</td>
<td>When action option is ACTIVATE, PACIFY, or ADJUST</td>
</tr>
</tbody>
</table>
Usage other than listed in the table does not cause a resource level security check. If multiple resources are affected, multiple resources are checked. The request is carried out or rejected as a whole.

For further details, refer to IBM Tivoli System Automation for z/OS Planning and Installation.

Usage

The INNGROUP command operates sysplexwide. For an overview see “Overview of Commands that Operate Sysplexwide” on page 9.

Automation policy overrides that have been entered using the INNGROUP command are preserved when performing an automation manager HOT restart and an INGAMS REFRESH, but are lost during WARM and COLD restarts.

Examples

If you enter ingroup a panel similar to Figure 48 is displayed.

This panel allows you to change the policy of application groups. You may specify:

- The target sysplex for the command with the Target field. If omitted, this defaults to the local system or sysplex. Specify the system name only when viewing or changing the policy of an application group that is not part of the local sysplex.
- The policy change that you want to make with the Action field. The policy options are explained below.
- Next you must specify in the Group(s) field either the name of the target group or a set of templates that will match one or more groups. All policy items work...
with a list of group name templates, and apply to all Move and Server groups that match those names. Although applying a policy to a basic group is not an error, it will not achieve anything either.

**Note:** A group name of xxxx will be automatically expanded into xxxx/APG/*, except for ADJUST and MEMBERS actions.

- For the Exclude, Avoid, and Include policy items, you can specify a list names of systems that the policy is to be set for. Specify a question mark (?) to see the list of systems within the sysplex.

See the online help for details about the policy items that you can change.

After entering the appropriate information in the fields for the policy item that you want to change, press PF10 or Enter to make your changes active and proceed with the action.

You can press PF4 to display the members detail panel (Figure 50 on page 136), where you can fine tune the group by changing the preferences of its individual members. If you do not specify a group on the main INGGROUP panel, a selection panel (Figure 49) displays all the groups that are currently available, with the message Multiple instances found for */APG. Select one application group and press Enter.

The members detail panel, as shown in Figure 50 on page 136, allows you to change the preference policy values for the members of Move and Server groups.
On this panel you can change the values in the **Result** field to change the availability target for that server group. Also, in the **Result** column you can change the preference value for that member. The effective preference value is shown in the **Eff** column. (Preference value changes are also possible for Server groups).

Changes to preference values for members, as set with the **Result** field are implemented in the following manner:

1. The preference value in the automation policy is not affected.
2. An adjustment value is calculated, which produces the Result value, when it was applied to the preference value in the automation policy.
3. The Result preference value is used by the automation when a preference value is required.
4. The adjustment value, not the Result, is preserved across automation manager HOT restart and across INGAMS REFRESH, but is lost during WARM and COLD restarts.

To make your changes active, press PF10. To reset the initial settings, press PF11.

The action taken after PF10 is pressed may differ from the assumed or proposed action displayed in the action column of panel INGKYGRB. This is because the assumed or proposed action is calculated by processing the group’s resources in alphanumeric sequence, while the actual action performed is calculated by the automation manager that processes the group’s resources in a random sequence.

For further details about this panel and how to use it, and sample preference scenarios, see the online help.

**Note:** The avoid, exclude, and include parameters are only applied to the members of the group. They are not propagated further when the member is also a group.

To remove subsystems from the list of members that can be used, enter the following:

```
INGGROUP group1 TSO/APG ACTION=EXCLUDE
```
Purpose

The INGHIST command shows history information about the work item processed by the automation manager. It also lets you display messages that the manager issues to the system logger.

Syntax

```
_INGHIST_
  
  REQ=HISTORY
  
  REQ=LOG

  RESOURCE=resource

  START=timestamp
  END=timestamp
  MAX=n
  WIMAX=n

  OUTDSN=dsname
  OUTMODE=LINE

  TARGET=system_name

  domain_id
  sysplex_name
```

Parameters

**REQ**

Specifies the request to be performed. It can have the following keywords:

- **HISTORY**
  
  Displays the work item history. This is the default.

- **LOG**
  
  Retrieves the messages written to the system logger by the automation manager.

**RESOURCE**

Specifies the name of the resource to be displayed. The format is name/type[/system]. The RESOURCE parameter can only be specified if REQ=HISTORY.

**START**

Is the start date and, optionally, time for the display of the history data. The format is yyyy-mm-dd [hh:mm:ss]. If omitted, the history data from the last hour will be displayed, unless another limiting parameter (resource name) is specified. If you specify more than one value, they must be enclosed with parentheses, or separated by a period (.), for example:

```
inghist resource=stdt000an1x start=(2000-03-10 08:00:00)
```

or

```
inghist resource=stdt000an1x start=2000-03-10.08:00:00
```

**END**

Is the end date and, optionally, time for the display of the history data. The format is yyyy-mm-dd [hh:mm:ss]. If you specify more than one value, they must be enclosed with parentheses, or separated by a period (.).
INGHIST

MAX
Specifies the maximum number of work items to be shown. Use this parameter to limit the output if many work items exist within the specified time period.

WIMAX
Specifies the maximum number of work item records to be shown after work item expansion. The value can be * to indicate unlimited or a value to limit expansion. The default is 10000. If the report becomes truncated then the message “***OUTPUT TRUNCATED***” will appear.

OUTDSN
For information on the OUTDSN parameter, see “OUTDSN Parameter” on page 13.

TARGET
For information on the TARGET parameter, see “TARGET Parameter” on page 10.

OUTMODE
For information on the OUTMODE parameter, see “OUTMODE Parameter” on page 12.

Restrictions and Limitations
To use INGHIST system operations does not have to be initialized.

Usage
The INGHIST command operates sysplexwide. For an overview see “Overview of Commands that Operate Sysplexwide” on page 9.

Time-interval search criteria are entered in local time. However, because the history data is stored using GMT time stamps, it is necessary for INGHIST to convert the local-time search criteria to GMT before retrieving the data. This may make it necessary for the operator to make a manual adjustment to the time-interval search criteria after a daylight-saving-time change has been set. The local-time-to-GMT conversion calculation will be one hour different after a daylight-saving-time change. Data written after a daylight-saving-time change will be retrieved as expected but data written before will be offset by one hour.

Examples
If you enter inghist resource=stdt000an1x/apl/aoc7 a panel similar to Figure 51 on page 139 is displayed.
The FIND/RFIND subcommands are supported. See “Varying the Format of the Command Output” on page 13 for information.

**INGHWSRV**

**Purpose**

The INGHWSRV command lets you:

- Terminate all BCP internal interface connections and disable the BCP internal interface
- Display Processor Operations and BCP internal interface NetView common global variables (CGLOBALS)
- Display specific target hardware or system NetView common global variables (CGLOBALS)
- Free control blocks that are associated with abended hardware system connection

**Syntax**

```
/SM590000/SM590000
INGHWSRV

TERM
GLOBAL

target_system_name

force
```

**Parameters**

**TERM**

Terminate all BCP internal interface connections and disable the BCP internal interface.

**GLOBAL**

Display Processor Operations and BCP internal interface common global variables.
INGHWSRV

target_system_name
Display NetView common global variables (CGLOBALs) for the specified target system name.

target_hardware_name
Display NetView common global variables (CGLOBALs) for the specified target hardware name.

FORCE
Clear the control blocks for the specified target hardware name.

Restrictions and Limitations
None.

Usage
You should use the FORCE option only after a connection failure to free control blocks that are associated with the target hardware name.

INGIMS

Purpose
The INGIMS command lets you:

• Issue both type-1 and type-2 IMS commands, and to issue those commands against an IMSplex and its members.
• Broadcast messages to all or selected IMS users.
• Issue a list of defined transactions or commands and view the output.
• Display the output of IMS transactions in full screen or pipeable line mode.

Syntax

```
/// INGIMS <Resource specification> <IMSplex specification> TARGET=target
```

也可打印

```
OUTDSN=dsname
OUTMODE=LINE AUTO NETLOG
```

Resource specification:

```
/// resource <REQ=CMD=cmd—MSG=message>
///       <REQ=XCMD=cmd—MSG=message>
///       <REQ=INFO>
///       <REQ=XINFO>
///       <REQ=BROADCAST—MSG=message—ROUTE=route>
///       <REQ=DEP>
///       <REQ=TCO>
```
IMSplex specification:

```
IMSPLEX=name REQ=XCMD CMD=cmd MEMBERS=(name) CMDWAIT=time
```

Parameters

resource

The resource specifies the name of the IMS subsystem that the command is issued to. The format is subsystem[/APL[/system]], where the subsystem is the specific name of the desired IMS subsystem and system is the name of the system that the IMS is running on.

**Note:** If you specify REQ=XCMD or REQ=XINFO, SA z/OS determines the IMSplex name that the IMS subsystem belongs to and routes the command to that IMS.

REQ

Specifies the request to be issued to the IMS subsystem. It can be one of the following:

- **CMD** Issues the specified IMS command or transaction and displays the results. The command is forwarded to IMS using its console interface.
  
  Enclose the command in single or double quotation marks if the command and parameters contain spaces.
  
  REQ=CMD is the default.
  
  **Note:** Do not specify the prefix character, forward slash (/), for IMS commands.

- **XCMD** This is the same as CMD but the command is forwarded to IMS via the IMS Operations Manager if either the IMSPLEX parameter is specified or an IMSplex name for the given resource is available in the policy. IMS type 2 commands are also supported in that case. Otherwise the console interface is used that supports only type-1 commands. However, the output is formatted in the same way in either case.
  
  **Note:** Do not specify the prefix character, forward slash (/), for IMS commands.

- **INFO** Issues a set of predefined commands and displays the results. The list of commands is obtained from the user-defined keyword-data pairs for the IMSINFO message ID in the policy of the subsystem that is specified by the resource name. The individual commands are executed as with CMD.
This is the same as INFO but the individual commands are executed as with XCMD.

**BROADCAST**
Issues an IMS broadcast to all or selected users.

**DEP**
Displays the dependent region associated with the control region.

**TCO**
Displays the details about the IMS Timed Control Operations function.

**IMSPLEX**
Specifies the IMSplex name.

**MEMBERS**
Identifies a list of members where the command should be executed. It can be:
- Asterisk (*), which causes the command to be forwarded to all registered command processing clients in the IMSplex.
- Percent sign (%), which causes the command to be forwarded to one registered command processing client with MASTER capability.
- A list of member names or type identifiers, or both. The type identifiers are subcategories of the category IMS (that is, DBCTL, DCCTL, DBDC, or FDBR) and must be preceded by an asterisk, for example, *DBCTL or *FDBR. *CTL can be used as shorthand for "*DBCTL *DCCTL *DBDC".

MEMBERS is optional. If omitted the command is forwarded to all registered command processing clients.

**RESP**
Specifies whether output should be returned to the caller or not. RESP is optional and the default is YES.

**CMDWAIT**
Specifies the maximum wait time for a command to complete. CMDWAIT is optional and the default is the value of CGLOBAL WAITTIME.

**MSG**
Specifies the message to be sent to all or selected IMS terminals or users for the REQ=BROADCAST request. Enclose the message in single or double quotation marks.

For REQ=CMD the MSG parameter can be used to contain a list of words that will be used to end processing if they are received anywhere in the output results. The words will normally be a required message ID. If no words are entered the CMD will be processed until any result is received.

**ROUTE**
Specifies the IMS routing information used to select the IMS terminals or users are to receive the message for a REQ=BROADCAST request. The contents of the ROUTE parameter are the parameters to be supplied to the /BRO transaction to route messages to IMS terminals or users. See the /BRO transaction for the details of the parameters for that transaction.

**TARGET**
For information on the TARGET parameter, see "TARGET Parameter" on page 10.

**OUTDSN**
For information on the OUTDSN parameter, see "OUTDSN Parameter" on page 13.
OUTMODE

For information on the OUTMODE parameter, see “OUTMODE Parameter” on page 12.

Restrictions and Limitations

To use the INGIMS command, system operations must be initialized. IMS subsystems must be enabled for system console commands using the MCS/E-MCS Console.

Usage

The INGIMS command operates sysplexwide. For an overview see “Overview of Commands that Operate Sysplexwide” on page 9.

Examples

If you enter ingims a panel similar to Figure 52 is displayed.

EVIKYCMD SA z/OS - Command Dialogs Line
Domain ID = IPSFM ---------- INGIMS ---------- Date = 06/08/09
Operator ID = NETOP1 Time = 15:08:38
Resource => SREL1 -or- IMSPlex name =>
IMS Members =>
Target => System name, domain ID or sysplex name
Request => XCMD CMD or XCMD, INFO or XINFO, BROADCAST, DEP, TCO
IMS Command =>
Response => YES Yes/No Command wait time => 60 (seconds)
IMS Route =>
IMS Message =>

Figure 52. INGIMS Command Dialog Panel

- The Resource field shows the name of the IMS subsystem to be used for issuing the requests. The format is name/type[/system]. Wildcards are supported.
- The IMSPlex name field shows the name of the IMSpex.
- The IMS Members field identifies the members where the commands should be processed. The IMS is either named by its subsystem ID or its type identifier. Type identifiers are DB, DC, FDBR, CQS, and so on. You must prefix the type identifier with an asterisk (*), for example, *DBDC. You can use *CTL as shorthand for *DB *DC *DBDC.
- The Target field shows the name of the system (system name, domain ID, or sysplex name) that the command should be routed to. Specifying this is only necessary if the resources do not reside in the local sysplex.
- The Request field shows the request to be carried out. It can be CMD, BROADCAST, INFO, XCMD, or XINFO.
Note: If you specify XCMD or XINFO, you must use either the IMSplex name or IMS Members field, or both, to specify the destination of the request.

- The IMS Command field shows the IMS transaction to be executed on the IMS subsystem specified by the resource field.
- The Response field indicates whether the output should be returned to the caller.
- The Command wait time field specifies the maximum wait time for a command to complete. The default is the value of WAITTIME common global variable.
- The IMS Route field shows the routing parameters for the /BRO command when the request is BROADCAST.
- The IMS Message field shows the message to be sent to users/terminals for the BROADCAST request.

The output from the IMS transaction appears in the blank area below the IMS Message field.

If you specify, for example, INGIMS EIMSDBRC REQ=BROADCAST a panel similar to Figure 53 is displayed.

---

Figure 53. INGIMS REQ=BROADCAST Command Dialog Panel

If you specify, for example, INGIMS EIMSDBRC REQ=BROADCAST CMD='DIS ACT' output similar to the following is displayed in a NetView window (which allows you to scroll through the output):

OM=IMSAA1OM Rc=00000000 Rsn=00000000
Master=MAC1 User=STCUSER Cmnd=DIS ACT

<table>
<thead>
<tr>
<th>MbrName</th>
<th>MsgData</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAA1</td>
<td>REGID JOBNAME TYPE TRAN/STEP PROGRAM STATUS CLASS</td>
</tr>
<tr>
<td>MAA1</td>
<td>4 IMSAA1M1 TP WAITING 1</td>
</tr>
<tr>
<td>MAA1</td>
<td>JMPRGN JMP NONE</td>
</tr>
<tr>
<td>MAA1</td>
<td>JBPRegn JBP NONE</td>
</tr>
<tr>
<td>MAA1</td>
<td>BATCHREG BMP NONE</td>
</tr>
<tr>
<td>MAA1</td>
<td>1 IMSAA1F1 FPM NO MSG. DFSIVP4</td>
</tr>
<tr>
<td>MAA1</td>
<td>3 IMSAA1F3 FPM NO MSG. DBFSAMP3</td>
</tr>
<tr>
<td>MAA1</td>
<td>DBTRGN DBT NONE</td>
</tr>
<tr>
<td>MAA1</td>
<td>IMSAA1RC DBRC</td>
</tr>
</tbody>
</table>
If you specify, for example, INGIMS REQ=INFO output similar to the following is displayed:

```
QUERY CQS DATA-----------------------------
OM=IMSAA1OM Rc=00000000 Rsn=00000000
Master=MAC1 User=STCUSER Cmd=DIS CQS

MbrName  MsgData
MAA1  DFS4444I DISPLAY FROM ID=MAA1
MAA1  JOBNAME VERS#  STATUS
MAA1  IMSAA1SQ  1.5 CONNECTED
MAA1  *09139/155812*
```

Note that a separator line is inserted between the individual command outputs.

You can issue a command against members of an IMSplex with the IMSPLEX parameter, for example:

```
INGIMS IMSPLEX=IMSPLEX_1 MEMBERS=*CTL REQ=XCMD CMD='QUERY AREA' OUTMODE=LINE
```

If you wish to view IMS dependent regions, use the INGIMS command dialog with REQ=DEP and the following screen is displayed:
INGINFO

Purpose

INGINFO displays details about a particular group or resource from the automation manager point of view.

Syntax

```
/SM590000/SM590000

INGINFO resource

NOHIST

WAIT=nnn

TARGET=system_name
domain_id
sysplex_name

/SM590000

OUTDSN=dsname

OUTMODE=LINE

AUTO

NETLOG
```

Parameters

resource

Specifies the name of the resource to be displayed. The format is name/type[/system].

NOHIST

Specifies that no history information should be included in the display.

WAIT

Specifies the number of seconds to wait before reporting that a timeout occurred if the automation manager does not provide the requested data. The maximum time interval is 999 seconds.

If omitted, the time interval is 30 seconds.
The INGINFO command operates sysplexwide. For an overview see "Overview of Commands that Operate Sysplexwide" on page 9.

Examples

If you enter, for example, `inginfo TPXA/APL/KEY3` a panel similar to Figure 55 is displayed.

This panel shows detailed information about the specified resource that includes:
- Resource statuses and dependencies
- Resource settings
- Relationships that have been defined for the resource
- Requests that have been issued against resource
- Votes that are pending for the resource
- History data that has been collected for the resource
INGINFO

There are two input fields, **Resource** and **Target**, where you can specify the name of the resource and the system that is to be queried. You can specify a wildcard for the resource name. Specify the system name only when obtaining details about a resource that is not part of the local sysplex.

If the specified resource name is not unique a selection panel will be displayed where you can select the resource concerned. You can either enter a resource name to get details about the resource, or press the PF9 key to refresh the information about the current resource.

If the resource is of type APL (that is, application), you can use the PF4 key to invoke the DispINFO command to obtain details about the resource from the automation agent's point of view. PF4 invokes DispMTR when the resource is a monitor resource, DispAPG when the resource is an application group (that is, type APG), and it invokes DispSYS when the resource is a system resource (that is, SYS).

If the resource is of type APL (that is, application) and is also associated with a pacing gate, you can use the PF5 key to invoke the InGPAC command to obtain information about this pacing gate. The PF5 key is not shown for applications that are not linked to a pacing gate and also not shown for any other resource type.

You can use the FIND(F) and RFIND(RF) subcommands to search the displayed data for a specific string. The line containing the string is then displayed as top line of the display. For example, to locate the line containing "abc" enter FIND abc. Optionally you may add the direction parameter, which is either P (previous) or N (next). The default is next. When you want to locate the next or previous occurrence of the string, enter RFIND or RF optionally followed by the direction.

**Notes:**

1. The history data is stored using GMT time stamps so it is necessary for InGInfo to convert these time stamps to local-time before displaying the data. The GMT-to-local-time conversion calculation will be one hour different after a daylight-saving-time change. Data written after a daylight-saving-time change will be retrieved as expected but data written before will be offset by one hour.

2. For groups, you may see Failed :Yes under the heading Group Details.... This Group Failed flag is set when any member of the group (Basic, Move or Server) enters a HardDown state (CTLDOWN, BROKEN, STOPPED) and remains set until manually cleared via InGSet. Its sole purpose is to support the MakeAvailable/passive/WhenGroupHasNotFailed relationship.

INGLIST

**Purpose**

The InGLIST command displays details about one or more resources.

**Syntax**

```
INGLIST resource filter criteria
```
INGLIST

TARGET=
  system_name
  domain_id
  sysplex_name

OUTDSN=dsname
OUTMODE= LINE
AUTO= NETLOG
WAIT= nnn

MEMBERS=YES

filter criteria:

OBSERVED=
  status

DESIRED=
  status

AUTOSTAT=
  status

COMPOUND=
  status

AUTOFLAG= YES NO

Chapter 2. System Operations Commands  149
Parameters

NETVASIS

Prefix the INLIST command with NETVASIS if you want to pass the description text in lower or mixed case.

resource

Specifies the name of the resources to be displayed. The format is name/type/system. It can be a list of names. The resource names must be separated by a blank. Asterisks (*) and a percentage sign (%) can be used as wildcard characters.
filter criteria

The filter criteria to be applied prior to displaying the data. See also “INGFILT” on page 122 for further information. The following values can occur:

**OBSERVED**
Specifies the observed statuses to be displayed. The statuses must be separated by a blank. It can be abbreviated, for example, to AV for available. If ‘^’ or ‘\’ is used, all statuses except the ones you specify are displayed. If an asterisk (*) is specified, the filter is reset (the filter previously set by the INGFILT command).

**DESIRED**
Specifies the desired statuses to be displayed. The statuses must be separated by a blank. It can be abbreviated, for example, to AV for available. If ‘^’ or ‘\’ is used, all statuses except the ones you specify are displayed. If an asterisk (*) is specified, the filter is reset (the filter previously set by the INGFILT command).

**AUTOSTAT**
Specifies the automation status to be displayed. The statuses must be separated by a blank. It can be abbreviated, for example, to ID for idle. If ‘^’ or ‘\’ is used, all statuses except the ones you specify are displayed. If an asterisk (*) is specified, the filter is reset (the filter previously set by the INGFILT command).

**COMPOUND**
Specifies the compound status. The statuses must be separated by a blank. It can be abbreviated, for example, to SA for satisfactory. If ‘^’ or ‘\’ is used, all statuses except the ones you specify are displayed. If an asterisk (*) is specified, the filter is reset (the filter previously set by the INGFILT command).

**AUTOFLAG**
Specifies the automation flag to be displayed. It can be either YES or NO and can be abbreviated. If an asterisk (*) is specified, the filter is reset (the filter previously set by the INGFILT command).

**CATEGORY**
Specifies the IBM-defined or user-defined category that the resource belongs to. More than one value can be specified. If an asterisk (*) is specified, the filter is reset (the filter previously set by the INGFILT command).

**SUBCAT**
Specifies the IBM-defined or user-defined subcategory of the resource. More than one value can be specified. The subcategory can contain wildcard characters. An asterisk (*) matches a string of arbitrary length and a percentage sign (%) matches a single character. If an asterisk (*) is specified, the filter is reset (the filter previously set by the INGFILT command). For compatibility reasons keyword SUBTYPE is still accepted.

**DESCR**
Specifies the text string used as a filter. The text can contain wildcards. An asterisk (*) matches a string of arbitrary length and a percentage sign (%) matches a single character. The DESCR parameter is case-sensitive. The text string must be enclosed in single or double quotation marks or parentheses() to maintain the case-sensitivity of the entry. If an asterisk (*) is specified, the filter is reset (the filter previously set by the INGFILT command).
INGLIST

GROUPTYPE
Specifies the type (nature) of the resource group. More than one value can
be specified. If an asterisk (*) is specified, the filter is reset (the filter
previously set by the INGFILT command).

HEALTH
Specifies the desired health statuses to be displayed. The statuses must be
separated by a blank. It can be abbreviated, for example, to NO for normal.
You can also specify an asterisk (*) to reset the current filter setting, for
example, INGFILT HEALTH=*. If ‘^’ or ‘\’ is used, all statuses except the
ones you specify are displayed.

JOBNAME
The jobname assigned to the resource. More than one jobname can be
specified. Wildcards are supported. If an asterisk (*) is specified, the filter
is reset (the filter previously set by the INGFILT command).

RUNTOKEN
The runtoken assigned to the resource. More than one runtoken can be
specified. Wildcards are supported. If an asterisk (*) is specified, the filter
is reset (the filter previously set by the INGFILT command).

PGNAME
The pacing gate assigned to the resource. More than one pacing gate can
be specified. Wildcards are supported. You can also specify an asterisk (*)
to reset the current filter setting.

For further information about statuses, see IBM Tivoli System Automation for
z/OS User's Guide.

TARGET
For information on the TARGET parameter, see "TARGET Parameter" on page
10.

OUTDSN
For information on the OUTDSN parameter, see "OUTDSN Parameter" on
page 13. Note, however, that for the INGLIST command the minimum record
length of the output data set is 256 bytes.

OUTMODE
For information on the OUTMODE parameter, see "OUTMODE Parameter" on
page 12.

WAIT
Specifies the number of seconds to wait before reporting that a timeout
occurred if the automation manager does not provide the requested data. The
maximum time interval is 999 seconds.
If omitted, the time interval is 30 seconds.

MEMBERS
Displays the members of resource groups that match the filter criteria. The
only valid specification is YES. MEMBERS=NO is accepted but does not have
any meaning.

Note: When specifying MEMBERS=YES only one resource name can be
specified and it must be fully specified.
Restrictions and Limitations

None.

Usage

The INGLIST command operates sysplexwide. For an overview see “Overview of Commands that Operate Sysplexwide” on page 9.

Examples

If you enter `inglist`, you will see information about all automated resources, as shown in Figure 56.

```
<table>
<thead>
<tr>
<th>CMD Name</th>
<th>Type</th>
<th>System</th>
<th>Compound</th>
<th>Desired</th>
<th>Observed</th>
<th>Nature</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLOCAS</td>
<td>APL</td>
<td>KEY1</td>
<td>INHIBITED</td>
<td>AVAILABLE</td>
<td>UNKNOWN</td>
<td></td>
</tr>
<tr>
<td>ALLOCAS</td>
<td>APL</td>
<td>KEY2</td>
<td>INHIBITED</td>
<td>AVAILABLE</td>
<td>UNKNOWN</td>
<td></td>
</tr>
<tr>
<td>ALLOCAS</td>
<td>APL</td>
<td>KEY3</td>
<td>SATISFACTORY</td>
<td>AVAILABLE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALLOCAS</td>
<td>APL</td>
<td>KEY4</td>
<td>INHIBITED</td>
<td>AVAILABLE</td>
<td>UNKNOWN</td>
<td></td>
</tr>
<tr>
<td>AMCP</td>
<td>APL</td>
<td>KEY1</td>
<td>INHIBITED</td>
<td>AVAILABLE</td>
<td>UNKNOWN</td>
<td></td>
</tr>
<tr>
<td>AMCP</td>
<td>APL</td>
<td>KEY2</td>
<td>INHIBITED</td>
<td>AVAILABLE</td>
<td>UNKNOWN</td>
<td></td>
</tr>
<tr>
<td>AMCP</td>
<td>APL</td>
<td>KEY3</td>
<td>PROBLEM</td>
<td>AVAILABLE</td>
<td>HARDOWN</td>
<td></td>
</tr>
<tr>
<td>AMCP</td>
<td>APL</td>
<td>KEY4</td>
<td>INHIBITED</td>
<td>AVAILABLE</td>
<td>UNKNOWN</td>
<td></td>
</tr>
<tr>
<td>ANMAIN</td>
<td>APL</td>
<td>KEY1</td>
<td>INHIBITED</td>
<td>AVAILABLE</td>
<td>UNKNOWN</td>
<td></td>
</tr>
<tr>
<td>ANMAIN</td>
<td>APL</td>
<td>KEY2</td>
<td>INHIBITED</td>
<td>AVAILABLE</td>
<td>UNKNOWN</td>
<td></td>
</tr>
<tr>
<td>ANMAIN</td>
<td>APL</td>
<td>KEY3</td>
<td>SATISFACTORY</td>
<td>AVAILABLE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANMAIN</td>
<td>APL</td>
<td>KEY4</td>
<td>INHIBITED</td>
<td>AVAILABLE</td>
<td>UNKNOWN</td>
<td></td>
</tr>
<tr>
<td>AOCAPPL</td>
<td>APL</td>
<td>KEY1</td>
<td>INHIBITED</td>
<td>AVAILABLE</td>
<td>UNKNOWN</td>
<td></td>
</tr>
</tbody>
</table>

Figure 56. INGLIST Command Dialog Panel

The amount of data depends on the screen size. Press PF10/11 to scroll horizontally.

Use the PF5 key to restrict the list of resources to be displayed. This causes the INGFILT panel to be displayed, where you specify the filter criteria to be used for the display.

You can use the PF4 key to toggle to the DISPSTAT panel to view the same resources, but now from the automation agent viewpoint. To return to the INGLIST panel press the PF3 key on the DISPSTAT panel. Note that pressing PF4 on the DISPSTAT panel displays the INGLIST panel for all resources shown in the DISPSTAT panel. However, this might not be the same as the original INGLIST display because the DISPSTAT display might be modified due to filtering or suppressing of resources that the automation agent does not handle, such as application groups.

This panel displays the following information:
The CMD field lets you specify command codes. To use one of the command codes shown, type the appropriate letter next to the resource name and press Enter. The following command codes are available:

A  Resets the status of the resource or application group (observed status, automation status). Sets or resets attributes that are assigned to the resource. The automation agent status can also be updated if you enter this command code. See “Setting the Status or Attribute of a Resource or Application Group” on page 156.

B  Start. Calls the INGREQ command dialog to make the resource available. See “INGREQ” on page 211 for further information.

C  Stop. Calls the INGREQ command dialog to make the resource unavailable. See “INGREQ” on page 211 for further information.

D  Displays dependency information and its evaluation for the resource.

E  Invokes the INGVOTE command, showing all requests currently present for the resource. Allows the operator to cancel a request.

F  Shows details about the resource by invoking the INGINFO command.

G  Shows all members for the selected application group or processor in a new window.

H  Invokes the DISPTRG command to show the trigger definitions for the selected resource.

I  Invokes the INGSCHED command to show the resource overrides for the selected resource.

J  Invokes the INGGROUP ACTION=MEMBERS command to show member details.

K  Invokes the INGCICS command using the selected resource.

L  Invokes the INGIMS command using selected resource.

M  Invokes the DISPMTR command to show the monitors that have been defined for the selected resource.

P  Invokes the INGPAC command using the selected resource.

T  Invokes the INGTWS command using selected resource.

U  Invokes the AOFEXC04 user exit if it has been installed.

X  Invokes the INGKLUP REQ=ANALYSE command. The X character examines the automation manager and automation agent for the selected resource to provide information about why the resource could not be brought into the desired state. The display lists all of the potential problems that might stop that action from succeeding. The display simply shows the resources concerned and a note of the potential problems that have been detected with them.

- The Name field shows the name of the resource.
- The Type field shows the type of the resource.
- The System field shows where the resource is defined.
- The Compound field shows the compound status of the resource. This is a summary of all statuses of the resource and provides a single value to check the status of a resource.
- The Desired field shows the status that the automation manager is trying to move the resource to. It can either be available or unavailable.
The **Nature** field applies to group resources only and defines the type of the group. It can have the following values:

**BASIC**
Indicates that the group contains a number of different resources, all of which perform different roles to constitute a complete application.

**MOVE**
Indicates that the group contains alternate instances of the same resource.

**SERVER**
Indicates that the group contains a number of readily interchangeable resources. The group has a target that tells the automation manager how many of them should be made available for the group to be available.

- The **Automation** field shows the status representing the automation agents' automation for the resource.
- The **Startable** field indicates whether or not it is possible to start the resource if the automation manager is asked to do so at this point in time.
- The **Health** field shows the health status of the resource.
- The **Auto** field shows the automation flag that is maintained by the automation manager. No automation is performed for the resource by the automation manager if the flag is off.
- The **Hold** field shows the hold flag that is maintained by the automation manager.
- The **Description** field shows descriptive information about the resource.
- The **Starttype** field shows the preset start type to be used the next time the resource is made available (started). This value is set by INGSET and will override any TYPE value specified (or defaulted) on the next INGREQ start request.
- The **Stoptype** field shows the preset stop type to be used the next time the resource is made unavailable (shutdown). This value is set by INGSET and will override any TYPE value specified (or defaulted) on the next INGREQ stop request. However, a stop type of FORCE, wherever specified, will always be honored.
- The **Trigger** field shows the trigger that is associated with the resource.
- The **Schedule** field shows the schedule (service period) that the resource is linked to.
- The **Category** field shows the IBM-defined or user-defined category of the resources. For IBM-defined categories, these might be CICS, DB2, or IMS.
- The **Subcategory** field shows the IBM-defined or user-defined subcategories of the resource. For IBM-defined the resource categories might be CICS, DB2, IMS, OPC, INFOSPHERE, LIFELINE, MQ and ITM.
- The **Jobname** field shows the job name assigned to the resource.
- The **Qual** field indicates how a resource qualifies for the current runmode. There are three indicators:
  - **G** Resource qualifies because it is in one or more groups that qualify.
  - **R** Resource qualifies because it was explicitly added via the INGRUN command.
  - **T** Resource qualifies because of its runtokens.
Resources can be associated with a pacing gate during startup and shutdown phases to control how many resources can be started or stopped concurrently. For such resources the following information is available:

**Pacing Gate**
The name of the pacing gate that is associated with this resource.

**Pacing Status**
The pacing status of the resource. The following status values exist:

- **blank**
The resource is not associated with a pacing gate.

- **Inactive**
  There is no activity in any of the pacing gates associated with the resource.

- **Waiting**
  The resource is waiting for transition through a pacing gate.

- **Starting**
  The resource is currently in transition through the pacing gate to become AVAILABLE.

- **Stopping**
  The resource is currently in transition through the pacing gate to become UNAVAILABLE.

Different colors are used to indicate when a particular status is regarded as abnormal:

- A desired status of UNAVAILABLE is shown in blue.
- A compound status of PROBLEM is shown in red, DENIED and INHIBITED are shown in pink, while a compound status of DEGRADED is shown in yellow.
- An observed status of HARDDOWN or PROBLEM is shown in red, while an observed status that is not in line with the desired status is shown in yellow.

Additionally you can use the following command codes:

- The slash character (/) to make the selected line the first line of the display.

The SORT, FIND, and RFIND subcommands are supported. See “Varying the Format of the Command Output” on page 13 for information.

**Setting the Status or Attribute of a Resource or Application Group**

If you enter command code A for a resource, a panel similar to Figure 57 on page 157 is displayed.
Use this panel to update the observed or automation status of the selected resource, or to modify other resource settings. In detail, you can:

- Set the start or stop type to be used the next time
- Turn on or off the automation flag
- Turn on or off the hold flag
- Turn on or off the group failed flag
- Update the observed status
- Update the automation status
- Update the agent status

Fill in the following fields:

**Action**
This identifies what to update. This is a one-digit numeric character ranging from 1 to 8. Depending on the specified action, fill out the appropriate field.

**START type**
Specify the new start type if action 1 is entered. Specify a question mark (?) to see a list of defined startup types. To reset the start type to its default, specify NORM.

**STOP type**
Specify the new stop type if action 2 is entered. Valid shutdown types are NORM, IMMED and FORCE. To reset the stop type to its default, specify NORM.

**Group failed flag**
Specify YES or NO if action 5 is entered.

**Observed Status**
Specify the new observed status if action 6 is entered. You can abbreviate the observed status, for example, you can enter AV for available, Valid
observed statuses are: SYSGONE, SOFTDOWN, HARDDOWN, STARTING, AVAILABLE, DEGRADED, PROBLEM, STOPPING, WASAVAILabe and UNKNOWN.

Automation Status
Specify the new automation status if action 7 is entered. You can abbreviate the automation status. Valid automation statuses are: IDLE, INTERNAL, BUSY, DENIED, PROBLEM and UNKNOWN.

Agent status
Specify the automation agent status if action 8 is entered. The agent status can be one of the following: CTLDOWN, RESTART, UP, MOVED, AUTODOWN or ENDED.

Setting the Hold flag to YES will only be granted if:
- The resource is down, that is, its observed status is either HardDown, SoftDown, SysGone or Unknown.
- The automation manager is not in the process of starting the resource.

INGLKUP

Purpose

INGLKUP has three purposes:
1. The INGLKUP command lets you display all jobs that are not controlled by SA z/OS. You can either stop, cancel or force those jobs with the corresponding MVS command.
   You can list all active jobs, started tasks (STCs), and APPC/MVS transaction programs. You can define an exclude list to tailor the list that is displayed.
2. The INGLKUP command can also be used to obtain diagnostic information about a particular resource if you want to perform debugging.
3. The INGLKUP command queries the automation manager and the various agents for the reasons why the resource concerned could not be brought into the desired status.

Syntax
1. Syntax for JOB display
Notes:
1  Line Mode only
2. Syntax for diagnostic display
   \[ \text{INGLKUP} \ \\ \text{resource} \Rightarrow \text{REQ=COLLECT} \ \\ \text{TARGET} = \text{system\_name} \] 
   \[ \text{OUTDSN=dsname} \] 
   \[ \text{OUTMODE=LINE} \] 
   \[ \text{AUTO} \] 
   \[ \text{NETLOG} \]

3. Syntax for analyse function
   \[ \text{INGLKUP} \ \\ \text{resource} \Rightarrow \text{REQ=ANALYSE} \ \\ \text{TARGET} = \text{system\_name} \] 
   \[ \text{STATE=UP} \] 
   \[ \text{DOWN} \] 
   \[ \text{OUTDSN=dsname} \] 
   \[ \text{OUTMODE=LINE} \] 
   \[ \text{TIMEOUT=nnn} \] 
   \[ \text{WAIT=nnn} \]

Parameters
REQ
   Specifies the type of request to be performed.
   
   JOB  Displays all jobs, started tasks, and APPC/MVS transaction programs that are not controlled by SA z/OS. Note that the term JOB includes batch jobs, started tasks, and APPC/MVS programs.
   
   COLLECT  Causes diagnostic information to be displayed in the netlog if OUTDSN is not specified.
   
   ANALYSE  Reports possible reasons why the resource concerned could not be brought into the desired state.
   
resource
   Specifies the name of the resource to be processed. The format is name[/type[/system]]. Wildcard is supported. If more than one resource matches the wildcard specification, a selection panel is displayed.
A resource specification is only valid when REQ=COLLECT.

**QUAL**
Specifies the type of "job" to be displayed. QUAL can be ALL, JOB, STC, APPC or *. The default is ALL. You can specify more than one qualifier by enclosing the qualifiers in parentheses.

**COMMAND**
Specifies the command to be issued in order to terminate the jobs/started tasks (STC). Valid entries are STOP, CANCEL or FORCE. These commands may be abbreviated. This parameter is used with Line Mode only.

**STATE**
Specifies the state the resource should be in:
- **UP**: Resource is available.
- **DOWN**: Resource is in a "DOWN" state.

**TIMEOUT**
Specifies the maximum number of seconds the INGLKUP command waits for responses from the remote system that the command was sent to. The maximum is 999 seconds. The default is 30 seconds.

**WAIT**
Specifies the number of seconds to wait before reporting that a timeout occurred if the automation manager does not provide the requested data. The maximum time interval is 999 seconds. If omitted, the time interval is 30 seconds.

**TARGET**
For information on the TARGET parameter, see "TARGET Parameter" on page 10.

**OUTDSN**
For information on the OUTDSN parameter, see "OUTDSN Parameter" on page 13. The record length must be at least 80 bytes if a resource is specified or 256 bytes if no resource is specified.

**OUTMODE**
For information on the OUTMODE parameter, see "OUTMODE Parameter" on page 12.

**Restrictions and Limitations**

INGLKUP issues the MVS D J,L command to get all active jobs. Then the REXX system address spaces are added to this list through the MVS D A,AXR+ command.

With the DISPSTAT command, the list of subsystems managed by System Automation is obtained. For the INGLKUP display, the two lists are compared, also taking the user-defined exclude list into consideration. System address spaces other than AXR* are not displayed.

INGLKUP may display subsystem address spaces if the subsystems are NON-MVS subsystems but are represented by an MVS address space. This occurs, for example, in the case of the IMS PPI dependent region.
Usage

1. Exclude List for REQ=JOB: An exclude list may be built using User Entry Type (UET) pairs. The exclude list allows the Automation Administrator to reduce the list of active jobs that are shown when REQ=JOB.

The UET data is specified in the policy database as follows:

Entry    INGLKUP
Type     EXCLUDE
Keyword/Data
        JOB

Data     Is JOB=jobname (where jobname specifies the JOB or STC to be excluded from the display). Wildcard is supported in jobname. For more information, refer to IBM Tivoli System Automation for z/OS Defining Automation Policy.

2. Output for REQ=COLLECT: If you specify a resource name with REQ=COLLECT, the output from the following commands is collected:
   - INGINFO
   - INGRELs
   - INGSCHED
   - DISPAPG (for APG resources only)
   - DISPINFO
   - DISPMTR (for MTR resources only)
   - DISPTRG

If you do not specify a resource name with REQ=COLLECT, the output from the following commands is collected:
   - INGLIST
   - INGSESS
   - DISPMTR
   - DISPSTAT

Whether you specify a resource name or not, output from the following commands is always displayed with REQ=COLLECT:
   - INGAMS
   - INGHIST
   - INGVOTE

Examples

If you specify inlkup and cancel a job with command code C, a panel similar to Figure 58 on page 162 is displayed.
This panel displays all active jobs, started tasks (STC) and APPC/MVS transaction programs that are not controlled by SA z/OS. You can either stop or cancel or even force the job (or jobs).

For each job the following information is shown:

- The **Cmd** field lets you specify the command codes shown on the panel. Type the appropriate letter next to the job name and press Enter. The following command codes are available:
  - **C** Cancel the job.
  - **F** Force the job.
  - **P** Stop the job.
- The **Job** field shows the job name.
- The **Step** field shows the step name.
- The **Procedure** field shows the procedure step name.
- The **Type** field shows the job type.
- The **ASID** field shows the address space ID (this is only present if the job name is not unique).
- The **System** field shows the name of the system that the job is running on.

The SORT, FIND, and RFIND subcommands are supported. See "Varying the Format of the Command Output" on page 13 for information.

If you specify REQ=ANALYZE for the INGLKUP command, the following panel is displayed:

![INGLKUP Command Dialog Panel for the ANALYZE Option](image)

**Figure 59. INGLKUP Command Dialog Panel for the ANALYZE Option**
INGMDFY

Purpose

The INGMDFY command displays the currently defined actions for the startup or shutdown of a subsystem and allows you to modify them for the next startup or shutdown.

INGMDFY also allows you to define additional actions or to delete defined actions for the startup or shutdown of a subsystem.

Syntax

```
>>> INGMDFY subsystem phase
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subsystem</td>
<td>The name of the subsystem that the defined actions are to be displayed for.</td>
</tr>
<tr>
<td>jobname</td>
<td>The job name of the subsystem that the defined actions are to be displayed for.</td>
</tr>
<tr>
<td>phase</td>
<td>This parameter is used to specify whether startup or shutdown actions are to be displayed.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>phase</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>START</td>
<td>Startup actions are to be displayed.</td>
</tr>
<tr>
<td>STOP</td>
<td>Shutdown actions are to be displayed.</td>
</tr>
</tbody>
</table>

Restrictions and Limitations

- The INGMDFY command can only be used when SA z/OS is initialized.
- The INGMDFY command is provided for fullscreen mode.
To ensure consistency during the startup or shutdown process of an application, the save and reset functions are not applicable during startup or shutdown processing of the related application.

Persistence of modifications:
- NetView Recycle (or ACF COLD which should only be used in emergency cases)
  Both the modified start/stop definitions in storage and the previously loaded ACF data in the KEEP are lost. At restart, SA z/OS initializes with the automation policy data from disk.
- Configuration Refresh
  If the configuration data for a subsystem has been changed in the automation policy, both the modified start/stop definitions in storage and the previously loaded ACF data in the KEEP are lost during the configuration refresh. The new ACF data is loaded from disk.
- Temporarily Changing ACF Data with ACF REQ=REPL
  When using ACF REQ=REPL, only the currently loaded ACF data is modified. An existing KEEP remains untouched so that previously loaded ACF data is restored after the next start/stop or after INGMDFY expires.

Security Considerations

The INGMDFY command supports resource level security. If turned on, the following profiles in class SYSAUTO are checked:

<table>
<thead>
<tr>
<th>Profile</th>
<th>Authority</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGT.sysplex.xcfgrp.RES_CONFIG</td>
<td>UPDATE</td>
<td>Always</td>
</tr>
</tbody>
</table>

For further details, refer to IBM Tivoli System Automation for z/OS Planning and Installation.

Usage

INGMDFY also displays default startup commands if no startup commands are defined for applications with a job type other than NONMVS and they can be modified.

Examples

If you enter INGMDFY followed by a subsystem name and START or STOP, a panel similar to Figure 60 on page 165 is displayed.
Use the command code S or C to overwrite the command definition of a selected row in another panel or to define a new command for an empty row.

Use the command code S or R to overwrite the reply definition of a selected row in another panel or to define a new reply in an empty row.

Only one command code can be issued at a time.

Additionally you can use the slash character (/) to make the selected line the first line of the display.

When overwriting the displayed data, the changes are at first accumulated in storage, until PF5 is used to save the changes. Thereafter the previously loaded configuration definitions are saved and the changed definitions are loaded into storage. The changed definitions are used for the next startup or shutdown of the application.

After the changed definitions have been used to start up or shut down an application, the changes are discarded and the saved configuration data is restored into storage to be used for further startups or shutdowns of the application.

The Modified flag indicates that changed startup or shutdown definitions have been loaded into storage.

PF7/PF8 lets you scroll forward and backward if more than one screen is needed to display the data.

PF10/PF11 lets you shift right and left if the defined action is longer than the screen width.

For further details about the fields on the panel, see the online help and *IBM Tivoli System Automation for z/OS Defining Automation Policy*.
**Purpose**

INGMOVE makes moving sysplex application groups easier. Rather than manipulating the preference value of each member in the sysplex application group, you simply specify where the group should be moved to.

In a sysplex application group of type MOVE only one member is active at a time. By specifying the new location of the move group, the active member is terminated and the member associated with the new location is activated.

INGMOVE supports the moving of a sysplex application group to another system.

**Syntax**

To move sysplex application groups

\[
\text{INGMOVE } \text{resource_name} \rightarrow \text{SYSTEMS=}\{\text{system}\} \rightarrow \text{QUAL=} \{\text{ACTIVE, INACTIVE}\} \rightarrow \text{TO=} \{\text{system, ANY, HOME}\} \rightarrow \text{feedback option} \rightarrow \text{WAIT=} \text{nnn} \rightarrow \text{ACTION=} \{\text{MOVE, PREPARE}\} \rightarrow \text{VERIFY=} \{\text{YES, NO, WTOR}\} \rightarrow \text{OUTDSN=} \text{dsname} \rightarrow \text{OUTMODE=} \{\text{LINE, AUTO, NETLOG}\} \rightarrow \text{TARGET=} \{\text{system_name, domain_id, sysplex_name}\} \rightarrow \text{Feedback option:} \rightarrow \text{FDBK=} \{\text{MSG,.userid, max_time}\} \rightarrow \text{WAIT} \rightarrow
\]

**Feedback option:**

\[
\text{FDBK=} \{\text{MSG,userid, max_time}\} \rightarrow \text{WAIT} \rightarrow
\]

**Notes:**

1. Can only be used in line mode.
The default depends on the type of task that the INGMOVE command runs on. If the task that the command runs on, is an autotask associated with a *ANY* console, the default is WTOR. The default for an unattended task is NO.

**Parameters**

*resource_name*

Specifies the name of the sysplex application group. More than one name can be specified, separated by a blank or a comma. Each name can contain a wildcard.

If no group name is specified, all sysplex application groups of type MOVE are displayed.

**WAIT**

Specifies the number of seconds to wait before reporting that a timeout occurred if the automation manager does not provide the requested data. The maximum time interval is 999 seconds.

If omitted, the time interval is 30 seconds.

**TO**

Specifies the system that the application group should be moved to.

Alternatively, *ANY* can be specified. This causes the application to be moved to any other viable system in the sysplex. In the background, an INGGROUP ACTION=EXCLUDE is done. The advantage of this, however, is that the INGMOVE command checks for the successful completion of the move operation within the allowed time interval.

**FDBK**

Specifies the feedback parameter that causes the final result of the command to be reported back to the designated instance.

**MSG**

Causes message signaling of the successful or unsuccessful completion of the command to be performed.

The message is sent back to the originator of the command. The originator is determined by the NetView OPID(S) function unless the user ID is specified. Depending on the status, either message ING300I or ING301I is issued. The message text contains the command that was processed successfully or failed.

*userid*

Identifies the NetView user ID of the person to be notified if different to the originating user.

Use the following values to specify when to issue the message:

**G**

Issue the message only when the resource has reached its expected status.

**F**

Issue the message only if the resource did not reach its expected state in the specified time interval.

**B**

Issue the message in all cases.

**WAIT**

To wait for the completion of the command (denoted by the *max_time* parameter). Note that the WAIT option is rejected when the INGMOVE command executes on a work operator.

*max_time*

The time interval in NetView format (mm:ss, :ss, mm or hh:mm:ss) that
**INGMOVE**

SA z/OS will wait. If the specified resource has not reached the expected state, the command is considered to have failed and the operation is posted in error.

**SYSTEMS**
Specifies a list of systems to be used as a filter. Only groups that can run on one of the specified systems are displayed. If more than one system is specified, they must be separated by a blank character and enclosed in parentheses.

**QUAL**
Specifies a qualifier for the SYSTEMS filter. It can be one of the following:

- **ACTIVE**
  Shows only those groups that are running on one of the systems specified with the SYSTEMS parameter.

- **INACTIVE**
  Shows only those groups that are not running on one of the systems specified with the SYSTEMS parameter.

**ACTION**
Specifies when the action is to be done. Valid values are:

- **MOVE**
  Perform the move immediately. This is the default.

- **PREPARE**
  Perform the move at the next recycle.

**VERIFY**
Specifies whether the operator is prompted for confirmation prior to carrying out the requested action. The default depends on the type of task that the INGMOVE command runs on. It is NO for Unattended tasks, otherwise YES.

- **YES**
  Asks the operator for confirmation.

- **NO**
  No verification is performed. This setting is always for Unattended tasks.

**WTOR**
The list of affected resources is displayed before the requested action is carried out. The operator is prompted to verify the list via WTOR.

The default depends on the type of task that the INGMOVE command runs on. If the task the command runs on is an autotask associated with a *ANY* console, the default is WTOR. The default for an unattended task is NO.

**TARGET**
For information on the TARGET parameter, see “TARGET Parameter” on page 10.

**OUTDSN**
For information on the OUTDSN parameter, see “OUTDSN Parameter” on page 13.

**OUTMODE**
For information on the OUTMODE parameter, see “OUTMODE Parameter” on page 12.

**Restrictions and Limitations**
The FDBK parameter will only monitor resources within the local sysplex.
Security Considerations

The INGMOVE command supports resource level security. If turned on, the following profile in class SYSAUTO is checked:

<table>
<thead>
<tr>
<th>Profile</th>
<th>Authority</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGT.sysplex.xcfgrp.RES.rename.restype</td>
<td>UPDATE</td>
<td>Always</td>
</tr>
</tbody>
</table>

If multiple resources are affected, multiple profiles are checked. The request is carried out or rejected as a whole.

For further details, refer to *IBM Tivoli System Automation for z/OS Planning and Installation*

Examples

If you enter `ingmove` a panel similar to Figure 61 is displayed.

```
INGKYMV0 SA z/OS - Command Dialogs Group 1 of 7
Domain ID = IPSNO ---------- INGMOVE ---------- Date = 11/23/09
Operator ID = NETOP1 Sysplex = KEY1PLEX Time = 13:08:38

B Start C Stop D INGRESL E INGVOTE F INGINFO G Members I INGSCHED
J INGROUP M Move P Prepare R Reset X Box Group / scroll

--- -----------  -----------  -----------  -----------  -----------  -----------  -----------
A

ARMGROUP SOFTDOWN KEY1 KEY2 KEY3 KEY4

MOVDISRN AVAILABLE KEY1 KEY2 KEY3 KEY4

MOVPLEXA AVAILABLE KEY1 KEY2 KEY3 KEY4

MOVPLEXB SOFTDOWN KEY1 KEY2 KEY3 KEY4

MOVPLEXC AVAILABLE KEY1 KEY2 KEY3 KEY4

MOVPLEXD AVAILABLE KEY1 KEY2 KEY3 KEY4

---

Command ===>

PF1=Help PF2=End PF3=Return PF6=Roll
PF9=Refresh PF12=Retrieve

Figure 61. INGMOVE Command Dialog Panel
```

This panel shows all sysplex application groups of type MOVE (referred to as move groups) that match the specified filter criteria. The following information is shown:

- The **Group name** column shows the name of the move group. Because the move groups are unique within a sysplex, only the first part of the resource group name is shown.
- The **Obs Status** column shows the observed status of the move group as seen by the automation manager.
- The **Systems** columns show a list of the systems where a member of the MOVE group exists and could run. Different colors are used to indicate the state of the system or the member of the group that is running on that system, as follows:
  - Red if the system is down
  - Yellow if the member of the group that is running on that system is not startable
  - Green if a member of the group is currently active on that system
  - Pink if the system is excluded
  - White if the system is avoided
  - Turquoise in all other cases
The *home* system, or systems, which are those that have the highest preference value, are underlined.

The system that has the highest actual preference is shown in reverse video, unless it is a home system. Note that this can be more than one system.

- The **Move to** column is an input field where you can specify the system that the group should be moved to. Moving a sysplex application move group means terminating the member that is currently active and starting the member on the designated system.

Initially, SA z/OS shows the best system to be used for a move operation in this field, based on the state of the group members and their preference points. The proposed system is shown in blue.

You can designate the system by specifying in the **Move to** field:
- The name of the system.
- An asterisk (*). This means that the group is moved back to its home system, which is the one with the highest base preference value. If more than one home system exists, it is moved to the first one in the list.
- An equals sign (=). This is a shorthand form that means to take the value from the nearest field above with a value specified in it.

You can enter the following action codes in the **Cmd** field to launch other commands:

- **B** Start. Calls the INGREQ command dialog to make the resource available. See “INGREQ” on page 211 for further information.
- **C** Stop. Calls the INGREQ command dialog to make the resource unavailable. See “INGREQ” on page 211 for further information.
- **D** Invokes the INGRELS command to display dependency information and its evaluation for the resource.
- **E** Invokes the INGVOTE command, showing all requests currently present for the resource. Allows the operator to cancel a request.
- **F** Shows details about the resource by invoking the INGINFO command.
- **G** Invokes the INGLIST MEMBERS=YES command to show all members for the selected application group in a new window.
- **I** Invokes the INGSCHED command to show the resource overrides for the selected resource.
- **J** Invokes the INGGROUP ACTION=MEMBERS command to show member details.
- **M** Moves the group to the designated system.
- **P** Moves the group to the designated system when the group is next recycled.
- **R** Resets the preference values of the group members to the values that are defined in the policy database.
- **X** Boxes the group. The member that is currently active acquires a preference of 2800 so that if the system is terminated the group will not move to another system.

You can use the / character to place the selected line as the first line of the display.

**Note:** This panel does not support the FIND and SORT subcommands.
If you specify one or more systems Figure 62 is displayed to ask you for confirmation.

For each group that is to be moved the panel shows:

- The name of the group
- The name of the system where the group is currently active
- The name of the system that the group should be moved to
- Descriptive information about the action

Specifying GO at the command line or pressing the PF10 key carries out the move operation. Specifying CANCEL at the command line or pressing the PF11 key returns you to the previous panel.

Note: Pressing PF2 or PF3 has the same behavior as PF11.

**INGMSGS**

**Purpose**

The INGMSGS command displays all important (that is, exceptional) messages that currently exist for a given system. The command is also used to delete exceptional messages.

**Syntax**

```
INGMSGS resource_name
  REQUEST=DISPLAY
  REQUEST=DELETE delete options

TARGET system_name
  domain_id
  sysplex_name
  OUTDSN dsname
  OUTMODE LINE
          AUTO
          NETLOG

delete options:

  DOM=(msgid)
  QUAL=qualifier
  SEVERITY=sev
```

Figure 62. INGMOVE Confirmation Panel

Verify the Move Group activities

<table>
<thead>
<tr>
<th>Group name</th>
<th>Runs on</th>
<th>Move to</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOVDISRN</td>
<td>AOCB</td>
<td>AOCB</td>
<td>now</td>
</tr>
<tr>
<td>MOVPLEXA</td>
<td>AOCB</td>
<td>AOCB</td>
<td>now</td>
</tr>
<tr>
<td>MOVPLEXB</td>
<td>AOCB</td>
<td>at recycle</td>
<td></td>
</tr>
</tbody>
</table>

Figure 62. INGMOVE Confirmation Panel
Parameters

**resource_name**

An optional parameter used as a filter. Only exceptional messages that are associated with the specified resource are shown. The resource name can contain a wildcard. More than 1 resource name can be specified. If no resource name is specified, only messages that are not associated with a resource automated by SA z/OS are taken into account (MVSESA).

**REQ**

Specifies the type of request to be performed:

- **DISPLAY**
  - Displays all exceptional messages for the resource. This is the default setting.

- **DELETE**
  - Deletes the message(s) based on the delete settings:

**DOM**

Specifies one or more IDs of messages to be deleted. The message IDs must be separated by a blank character and enclosed in parenthesis or quotes if more than one message ID is specified. The message ID can contain wildcards.

**QUAL**

Specifies a qualifier that is used to identify the message in addition to its message ID.

**SEVERITY**

Specifies the severity of the message used as Filter Criteria. Valid entries are UNUSUAL, IMPORTANT and CRITICAL.

**EXPIRED**

Specifies the expiry date and time of the message. Any message that was issued prior to the specified timestamp will be deleted when also the other specified filter criteria match. The date format is yyyy-mm-dd and the time format is hh:mm. The time specification is optional.

**TARGET**

For information on the TARGET parameter, see "TARGET Parameter" on page 10.

**OUTDSN**

For information on the OUTDSN parameter, see "OUTDSN Parameter" on page 13.

**OUTMODE**

For information on the OUTMODE parameter, see "OUTMODE Parameter" on page 12.

Restrictions and Limitations

The INGMSGS command can only be used when SA z/OS has initialized.

A qualifier is not displayed when invoking INGMSGS in line mode.
Examples

If you enter ingmsgs a panel similar to Figure 63 is displayed.

![Figure 63. INGMSGS Command Dialog Panel](image)

This panel displays exceptional messages that have been captured by SA z/OS. An exceptional message is a message whose severity is either Unusual, Important or Critical. By default the messages are displayed in chronological order (the oldest message is shown at the top of the panel) but can be rearranged in any other order, for example, by severity with the SORT subcommand. The individual messages are colored depending on their severity. The color attribute is defined via SDF status definitions.

For each message the following information is shown:

- The **Timestamp** column shows the date and time when the message was captured.
- The **S** column shows the severity assigned to the message. It can be:
  - U Unusual
  - I Important
  - C Critical
- The **Message** column shows the message text, the system where the message was captured, the qualifier if present that is associated with the message and the name of the resource associated with the message. If no resource name is shown, the message is associated with the system.

You can enter the following action codes in the **Cmd** field to launch other commands:

- **D** Delete (DOM) the message. This also removes the message from any target it has been forwarded to (SDF, NMC, TEP).
- **F** Shows details about the resource that is associated with the message by invoking the INGINFO or DISPSYS command.
Purpose

INGNTFY allows you to add, change, delete, and display notification operator settings.

Syntax

```
/SM590000
  INGNTFY
  your_opid
  ON Settings
  your_opid
  OFF
  your_opid
  ADD
  your_opid
  Settings
  DELETE
  your_opid
  DISP
  opername
  your_opid
  QUERY
  opid
```

```
/SM590000
  OUTMODE=
    LINE
    AUTO
    NETLOG
```

```
/SM590000
  SETTINGs:
    CLASS=
      (class)
      (4080)
    HEldMSG=
      (type)
    DESC='description'
```

Parameters

**ON**  Allows the operator who issues the command to receive notifications.

**OFF**  Turns off notifications for the operator who issues the command.

**ADD**  Adds the operator specified in *opid* to the list of notification operators, or changes the definition of the existing operator specified in *opid*.

**Note:** Use of the ADD function to change the definition of an existing operator completely deletes the old definition. It is necessary to re-specify all message classes and held message types that you want this operator to retain. This allows you to remove unwanted message classes and held message types from an existing operator.

**DELETE**  Removes the operator specified in *opid* from the list of notification operators.
INGNTFY

**DISP**
Displays information about the operator that is specified in `opername`, which can also contain a wildcard, such as `*abc`, `abc*`, or `*abc*`.

**QUERY**
Displays information about the operator that is specified in `opid`.

`opid`
The ID of the operator to add, change, delete, or query.

`your_opid`
The ID of the operator issuing the command.

**CLASS**
The classes of messages the operator is to receive. See *IBM Tivoli System Automation for z/OS Messages and Codes* for information about which classes are valid. Up to ten comma-delimited classes, enclosed in parentheses, can be defined for an operator. The default classes are 40 and 80, unless defined otherwise in the automation control file.

**DESC**
An optional description of the operator. If the description contains blanks, commas, or other punctuation characters it must be enclosed in single quotation marks.

**HELDMSG**
The type or types of messages to hold on the operator's screen. If more than one type is specified, the types should be separated by commas and enclosed in parentheses. The types are:

<table>
<thead>
<tr>
<th>Type</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Immediate Action</td>
</tr>
<tr>
<td>D</td>
<td>Immediate Decision</td>
</tr>
<tr>
<td>E</td>
<td>Eventual Action</td>
</tr>
<tr>
<td>I</td>
<td>Information</td>
</tr>
<tr>
<td>W</td>
<td>System Wait</td>
</tr>
</tbody>
</table>

**TARGET**
For information on the TARGET parameter, see “TARGET Parameter” on page 10.

**OUTMODE**
For information on the OUTMODE parameter, see “OUTMODE Parameter” on page 12.

For further details of message types and message notifications, see *IBM Tivoli System Automation for z/OS Messages and Codes*.

**Restrictions and Limitations**
Any changes you make using the INGNTFY command are in effect only until the automation control file is reloaded.

The INGNTFY command can only be used when SA z/OS is initialized.

**Examples**
To turn your messages off, enter `ingntfy off`
To add notification operator Jim, assign him all messages in class 43, and hold Immediate Action type messages on his screen, enter:

```
ingntfy add jim class=43 heldmsg=a
```

If no parameter is specified in the INGNTFY command, a panel similar to Figure 64 is displayed.

```
<table>
<thead>
<tr>
<th>AUFKAANT</th>
<th>SA z/OS - Command Dialogs</th>
<th>Line 1 of 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain ID  = IPUN8</td>
<td>INGNTFY</td>
<td>Date = 06/19/09</td>
</tr>
<tr>
<td>Operator ID  = NETOP1</td>
<td></td>
<td>Time = 20:12:52</td>
</tr>
</tbody>
</table>

Cmd:  A Add   C Show/Change settings   D Delete   O Turn OFF msg receiving

<table>
<thead>
<tr>
<th>Cmd</th>
<th>Operator</th>
<th>System</th>
<th>Log</th>
<th>Rcv</th>
<th>Description</th>
<th>Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUTNOTI1</td>
<td>AOCB</td>
<td>Y</td>
<td>N</td>
<td>MSG AOP</td>
<td>40 80</td>
</tr>
<tr>
<td></td>
<td>DON</td>
<td>AOCB</td>
<td>Y</td>
<td>N</td>
<td>DON JONES</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>GIS</td>
<td>AOCB</td>
<td>Y</td>
<td>Y</td>
<td>GISELA STUART</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>MIK</td>
<td>AOC8</td>
<td>Y</td>
<td>Y</td>
<td>MIK SMITH</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>NOTI1</td>
<td>AOCB</td>
<td>N</td>
<td>Y</td>
<td></td>
<td>40 41 42 43 44</td>
</tr>
<tr>
<td></td>
<td>NOTI2</td>
<td>AOCB</td>
<td>N</td>
<td>Y</td>
<td></td>
<td>80 81 82</td>
</tr>
</tbody>
</table>

Command ===>
PF1=Help PF2=End PF3=Return PF6=Roll PF9=Refresh PF12=Retrieve
```

Figure 64. INGNTFY Command Dialog Panel

- The **Cmd** field lets you specify the command codes shown on the panel. Type the appropriate letter next to the resource name and press Enter. The following command codes are available:
  - **A** Add a notification operator using the settings of the selected operator.
  - **C** Show or change the settings for the selected operator.
  - **D** Delete the notification operator.
  - **O** Turn off message receiving for the selected notification operator.
- The **Operator** field shows the operator ID. It is not necessarily defined to NetView.
- The **System** field shows the name of the system where the operator is defined.
- The **Log** field shows a Y if the operator was logged on when the command dialog was started, or an N if the operator was not logged on.
- The **Rcv (Receive)** field shows a Y if the operator was receiving messages when the command dialog was started, or an N if the operator was not receiving messages.
- The **Description** field shows a description of the operator from either the automation control file or the INGNTFY command dialog.
- The **Classes** field shows the classes that are assigned to the notify operators.
Purpose

The INGPAC command displays information about pacing gates, their characteristics and runtime statistics.

It also shows details about pacing gate contention with the resources currently granted access to the gate and those waiting for access to the gate.

The RESOURCE-keyword can be used to have INGPAC list all pacing gate(s) that are associated with the resource denoted by the resname variable.

Syntax

```
- INGPAC
  pgnamelist
  RESOURCE=resname

  TARGET=system_name
domain_ID
sysplex_name

  OUTDSN=dsname
OUTMODE=LINE
AUTO
NETLOG

  WAIT=YES
WAIT=nnn

filter-criteria:

  TYPE=START
STOP
SYSTEM=name

```

Parameters

**pgnamelist**

A list of one or more pacing gate names, separated by blank or comma, enclosed in parentheses. Parentheses can be omitted if the list consists of only one element.

**resname**

The fully qualified name of the application in automation manager notation for which the associated pacing gates are listed.

**filter-criteria**

Specifies filter to further reduce the set of pacing gates being displayed. The following filters are displayed:

**TYPE**

Specify START to only display pacing gates effective for starting of applications.

Specify STOP to only display pacing gates effective for stopping of applications.

**SYSTEM**

Specify the name of a system to only display pacing gates defined for that system.
INGPAC

TARGET
For information on the TARGET parameter, see "TARGET Parameter" on page 10.

OUTDSN
For information on the OUTDSN parameter, see "OUTDSN Parameter" on page 13.

OUTMODE
For information on the OUTMODE parameter, see "OUTMODE Parameter" on page 12.

WAIT
Specifies whether or not to wait until the request is complete. The default is YES.

$nnn$ is the number of seconds to wait before giving up and reporting that a timeout has occurred. The maximum time interval is 999 seconds.

Restrictions and Limitations
None.

Security Considerations
The INGPAC command supports resource level security. If turned on, the following profile in class SYSAUTO is checked:

<table>
<thead>
<tr>
<th>Profile</th>
<th>Authority</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGT.sysplex.xcfgrp.RES._MANAGER_PACING</td>
<td>UPDATE</td>
<td>When command code $R$ is issued to release a resource on the INGPAC detail panel.</td>
</tr>
</tbody>
</table>

Usage other than listed in the table does not cause a resource level security check. For further details, refer to IBM Tivoli System Automation for z/OS Planning and Installation

Examples
If you enter INGPAC a panel similar to that shown in Figure 65 on page 179 is displayed:
The amount of data depends on the screen size. Press PF10/11 to scroll horizontally.

- The CMD field lets you specify command codes. To use one of the command codes shown, type the appropriate letter next to the pacing gate and press Enter. The following command codes are available:
  - D to display a detail panel showing pacing gate statistics and providing a snapshot of resources currently using the pacing gate or waiting for use.
  - L to list all resources that potentially may use this pacing gate for start or stop activities.

The SORT, FIND, and RFIND subcommands are supported. See “Varying the Format of the Command Output” on page 13 or online help for further information.

Refer to the online help on the panel (PF1) for a description of individual fields.

If you enter command code D for a pacing gate, a panel similar to Figure 66 on page 180 is displayed here:

![Figure 65. INGPAC Pacing Gate(s) Display Panel](image)

The amount of data depends on the screen size. Press PF10/11 to scroll horizontally.
The amount of data depends on the screen size. Press PF10/11 to scroll horizontally.

- The CMD field lets you specify command codes. To use one of the command codes shown, type the appropriate letter next to the resource name and press Enter. The following command codes are available:
  - I to display a panel with details about this application
  - R to release the resource to let it start or stop regardless of the pacing gate status at the moment.

The SORT, FIND, and RFIND subcommands are supported. See “Varying the Format of the Command Output” on page 13 or the online help for further information.

Refer to the online help on the panel (PF1) for a description of the individual fields.

### INGPW

**Purpose**

The INGPW command initializes the SA z/OS password protection feature. The INGPW command processor maintains a VSAM file containing passwords for user IDs, called the password data set. These passwords are used when establishing sessions to external products. The records in the password data set are keyed using a combination of the user ID and owner ID. Each record has these fields:

- The current-password field
- The new-password field
- The password mask
- The date-password-last-changed field
The password change interval
Passwords are stored in encrypted format

Syntax

```
-ingpw user_id owner_id
```

Syntax diagrams:

```
READ

INIT

INIT=password

DELETE

REGEN

UPDATE

Parameters

user_id

user id (4-32 characters). The parameter is case-sensitive.

owner_id

(1-8 characters) the owning entity for which password maintenance services are required (custom value, policy entry, and so on, depends on consumer service). The parameter is case-sensitive.

READ

Specifies that the appropriate password is retrieved from the data set. This is the default if no request is specified.

INIT

Used to create an entry in the password data set for the specified user_id and owner_id values and to specify an initial or new password value for the entry. Using INGPW with the INIT parameter is required as part of installing the password protection feature. An INIT request without a new password can be used to redefine expiration interval and password mask for the existing password record. If a value is specified for the MASK=mask keyword together with the INIT keyword, then the value of the mask becomes the default mask for the REGEN keyword.

password

Specifies initial/new password for the specified user_id and owner_id. The password value must be between 4 and 32 characters long. The password is case-sensitive.

DELETE

Specifies that the record with the matching user_id and owner_id should be deleted.

REGEN

Causes a new password to be generated (satisfying the requirements of the
mask if it is specified). The current and new passwords are shown in response to INGPW REGEN and any subsequent INGPW READ until there is a INGPW UPDATE.

**MASK**

Describes the format used when generating new passwords for specified user_id and owner_id records. The password generation mask consists of 4 to 32 pairs of characters, each pair defining the attributes of a single password character. The first character of each pair determines how the second character is used.

- **!** A password character of the value indicated by the following mask character is required.
- **%** A password character of the type indicated by the following mask character is required.
- **?** A password character of the type indicated by the following mask character is optional.

The password mask is processed from left to right, and generates a string of characters with a length equal to or greater than the number required. Passwords of at least 4 characters are required. The password character type codes indicate a set of characters to be used when generating a new password. The valid type codes are case-sensitive and are as follows:

- **$** - National only
- **A** - Uppercase alphabetic only
- **a** - Lowercase alphabetic only
- **B** - Uppercase Alphabetic/National
- **b** - Lowercase Alphabetic/National
- **C** - Uppercase Consonant (Alphabetic, no vowels)
- **c** - Lowercase Consonant (Alphabetic, no vowels)
- **E** - Even numerics
- **N** - Numeric only
- **O** - Odd numerics
- **V** - Uppercase vowels only
- **v** - Lowercase vowels only
- **W** - Uppercase alphanumeric (no vowels)
- **w** - Lowercase alphanumeric (no vowels)
- **X** - Uppercase alphanumerics
- **x** - Lowercase alphanumerics
- **Y** - Uppercase alphabetical/National
- **y** - Lowercase alphabetical/National
- **Z** - Uppercase alphanumeric/National (no vowels)
- **z** - Lowercase alphanumeric/National (no vowels)

For example, a password mask of !N!N!T%!S%?N%N%E would cause passwords to be generated that begin with 'NNT' followed by a national character, a consonant, and a two or three digit even number. If a value is not specified for mask, then the system-defined default is %X%X%X%X%X%X%X.
EXPINT
Defines the expiration interval used when defining or generating new
passwords.

int
number of days before current password expiration and next password
regeneration. Default is 0 (no password expiration required). The expiration
interval can be 0 to 365 days.

UPDATE
Makes the new password generated by REGEN become the current password.

Return Codes
0 Successful completion.
4 No storage obtained.
8 Record not found in VSAM data set.
12 Error Processing VSAM data set.
16 Bad Invocation.
20 Not Authorized.

Security considerations
For security reasons, using command-class checking on INGPW is strongly
recommended. Refer to IBM Tivoli System Automation for z/OS Planning and
Installation for more details about command authorization.

Access to the password records can be protected using the
ING.PW.user_id.owner_id resource in the SYSAUTO class defined in RACF. The
following example shows how to define an INGPW resource in RACF:
RDEFINE SYSAUTO ING.PW.ENSOPERATOR.ZBXENS UACC(NONE)

The user record for user_id ENSOPERATOR and owner_id ZBXENS is defined as a
resource in the RACF class SYSAUTO with a universal access attribute of NONE.
Note that you use a wildcard character to specify the resource more generically if
that is suitable for your environment.

<table>
<thead>
<tr>
<th>Level of password access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>READ:</td>
<td>read password record (READ request)</td>
</tr>
<tr>
<td>UPDATE:</td>
<td>update existing password (UPDATE request)</td>
</tr>
<tr>
<td>CONTROL:</td>
<td>initialize or change existing record (INIT,REGEN requests)</td>
</tr>
<tr>
<td>ALTER:</td>
<td>delete password record (DELETE request)</td>
</tr>
</tbody>
</table>

Usage
Use a blank instead of a comma to separate the user_id and owner_id values.

If parameter values are required in mixed case (for example, mixed case
passwords), you must use the 'NetVasis INGPW' form to make sure that the data is
passed without uppercase translation. Failure to do so may result in incorrect data
returned or generated by the INGPW command.
## INGPW

### Examples

**NETVASS INGPW TestUser TESTSRV,INIT=MYp01wDs,MASK=%A%A%a%N%N%A%A%a**

The command initializes a new record for `used_id=TestUser` and `owner_id=TESTSRV` with initial mixed case password `MYp01wDs` and set MASK to generate passwords in mixed case.

**INGPW TestUser TESTSRV,INIT=MYp01wDs,MASK=%A%A%a%N%N%A%A%a**

The command initializes a new record for `used_id=TestUser` and `owner_id=TESTSRV` with initial upper case password `MYP01WDS` and set MASK to generate passwords in uppercase.

## INGPLEX

### Purpose

The INGPLEX command comprises all the sysplex-related functions of SA z/OS. It can be called in full mode and in line mode.

### Syntax

```
INGPLEX CDS
    TYPE= ARM
    DETAIL=CHPID
    CFRM
    LOGR
    SFM
    SYSPLEX

(1)

CF
    CONsole
    DUMP

(2)

IPL DEL
    IPLRECD
    SHOW= IPLRECD
    DISP= IPLREC
    COMP= IPLREC WITH= IPLREC
    LIST=sysname

SDump

SLIP
    ID=slipid
    SYSTEM=sysname

SVCdump
    sysname
    LIST=COMP
    DUMP

SYStem

TARGET= system_name
    domain_ID
    sysplex_name

OUTMODE= LINE
    AUTO
    NETLOG
```

**IPLRECD:**
INGPLEX

IPLREC:

\[-\text{sysname/timestamp[member[/suffix]]}\\]

Notes:
1 For details see “INGCF” on page 92.
2 Line mode only
3 Line mode only

Parameters

CDS
Displays information about CDSs and supports replacement of the current alternate CDS by a new one as well as making the alternate CDS the new primary. For further information about INGPLEX CDS refer to “INGPLEX CDS” on page 189.

TYPE
The type of CDS that the CDS function is issued for. Possible values are ARM, CFRM, LOGR, SFM, and SYSPLEX.

DETAIL
If you specify this parameter with the CDS function, the channel paths for the respective CDS type are displayed.

CF
This is the equivalent of the INGCF command.

CONsole
Displays information about consoles.

DUMP
Shows the DUMP submenu.

IPL
Shows and compares IPL information. It can be issued with the following options:

DEL
Deletes a single IPL record and all its related information. Note that the DEL parameter is supported in line mode only.

SHOW
Shows the details panel of the specified IPL record.

DISP
Shows all, one, or particular PARMLIB members used by the IPL of the specified system and at the specified date and time.

COMP
Compares all, one, or particular PARMLIB members used by the IPL of the specified system and at the specified date and time with those specified in the WITH parameter.

LIST
Shows the IPL summary records of the specified system.
**WITH**

The COMP parameter compares all, one, or particular PARMLIB members used by the IPL of the specified system, at the specified date and time with those specified in parameter WITH.

*sysname*

Is the name of the system in the sysplex.

*timestamp*

Is the IPL date and time. The format is YYYYMMDDhhmm.

*member*

Is the name of the PARMLIB member without the suffix.

*suffix*

Is the suffix of the PARMLIB member.

For further information about INGPLEX IPL refer to “INGPLEX IPL” on page 200.

**SDUMP**

Displays and controls the SDUMP options being set on all systems in the sysplex. For further information about INGPLEX SDUMP refer to “INGPLEX SDUMP” on page 201.

**SLIP**

Displays and controls all SLIP traps of all systems in the sysplex. Controlling is limited to DISABLE, ENABLE, or REMOVE a SLIP trap. The following parameters are supported:

*ID*

Limits the line mode output and the initial full screen display to the particular SLIP trap ID.

*slipid*

Is the ID of a SLIP trap. It can consist of one to four characters. Wildcards are not supported.

*SYSTEM*

Limits the line mode output and the initial full screen display to the particular system.

For further information about INGPLEX SLIP refer to “INGPLEX SLIP” on page 207.

**SVCdump**

Allows you to issue a multisystem dump of up to 15 address spaces including data spaces owned by the address spaces, structures used by the address spaces, and XCF group members on the same or on other systems in the sysplex of those groups the address spaces have joined. The following parameters are supported:

*sysname*

Is the name of the system that has joined the XCF group of the NetView that the operator is logged on to.

*REQ*

Specifies the type of the line-mode request.

**LIST**

Requests the list of all active address spaces. This is the default when the parameter is omitted.

**COMP**

Requests the list of all components of the address spaces that are passed to the command.
DUMP
Requests a dump of the address spaces and their components that are passed to the command.

For further information about INGPLEX SVCDUMP refer to “INGPLEX SVCdump” on page 204.

SYStem
Displays information about a member system of the sysplex.

TARGET
For information on the TARGET parameter, refer to “TARGET Parameter” on page 10.

OUTMODE
For information on the OUTMODE parameter, refer to “OUTMODE Parameter” on page 12.

Example
If you specify INGPLEX without parameters, the selection panel shown in Figure 67 is displayed.

Use this panel to launch other command dialogs showing sysplex-related information.

Use one of the following command codes to invoke another command dialog:

1 INGPLEX SYStem
Displays the target sysplex name, its GRS mode and its systems. For each system the following details are shown:
- System name
- Status
- SSUM action
- SSUM interval time
- SSUM weight
INGPLEX

- SFM failure detection interval
- WLM query timestamp
- Total free SUs
- SUs used
- SU-consuming resources
- Free SUs

See “INGPLEX SYStem” on page 195.

2 INGPLEX CONsole
Displays the following details for the target sysplex:
- Master console name
- WTO & WTOR buffer utilization
- Number of queued messages of various types
- Awaiting mounts and operator requests
- List of consoles with more details, such as:
  - Name
  - Status
  - Authority
  - Number of WTOR buffers, etc.

See “INGPLEX CONsole” on page 197.

3 INGPLEX CF
Displays the coupling facilities in the sysplex, their spaces, and CF levels.
Each coupling facility can be drained and enabled. The structures of each
coupling facility can be displayed, rebuilt, and forced. Duplexing can be
started and stopped, when supported. The paths of the coupling facility
can be displayed and set online or offline.

See “INGCFL” on page 110.

4 INGPLEX CDS
Displays the couple data sets in the sysplex. For each couple data set,
paths and more detailed information can be displayed. The couple data
sets can be switched and new alternate data sets can be allocated.

See “INGPLEX CDS” on page 189.

6 INGPLEX IPL
Displays and compares IPL information being used during the IPL of an
operating system. The information shown or compared can be varied from
all available information to the content of a particular PARMLIB member.

See “INGPLEX IPL” on page 200.

7 INGPLEX DUMP
Displays the default dump options as well as the setting of the slip traps
for all systems in the sysplex. It also allows you to change the settings
either locally, sysplexwide, or for some systems in the sysplex. In addition,
the function allows to take multisystem SVC dumps.

See “INGPLEX SDUMP” on page 201, “INGPLEX SVCdump” on page 204,
and “INGPLEX SLIP” on page 207.

10 INGSTR
Displays all allocated and unallocated structures, independent of the
coupling facility. The structures can be displayed, rebuilt, forced, and
duplexing can be started and stopped, when supported. You can also
reallocate all structures from their current location to their preferred
location if all systems in the sysplex support the XCF REALLOCATE
command.

See “INGSTR” on page 247.

INGPLEX CDS

Purpose

The CDS function displays information about all the couple data sets in the
system, including details of the corresponding policies. For every CDS type that is
required by the implementation INGPLEX CDS allows the operator to:
• Switch from the primary to the alternate CDS
• Define a new alternate CDS
• Change the active policy (if applicable)

Actions are started by specifying an action code for a selected CDS type on the
panel.

Actions

The possible action codes are:

*A: Allocate alternate CDS

Replaces the current alternate CDS for a selected CDS type with a new
one. There are two options how to do this:
• The alternate CDS is allocated automatically by SA z/OS.
  This automatic allocation requires that spare volumes have been defined,
  and that one of these spare volumes is available.
• Specify the data set that is to be used as the new alternate CDS.
  If you specify your own data set, observe the following:
  – The data set must exist
  – It must have been formatted with the XCF formatting tool
  – It must be at least as large as the current primary CDS, which means
    that every value you have passed to the XCF formatting tool (for
    example, in the case of a sysplex CDS, the maximum number of
    systems supported) must be equal to or greater than the
    corresponding value of the primary CDS.

C: Display CHPIDs

Displays information about the channel paths for the selected CDS type.

D: Display CDS information

Displays detail information about the selected CDS type. This comprises
the formatting parameters and the policies that are contained in the CDS, if
applicable. When the CDSs of the selected type contain policies, the detail
information panel provides further actions:

D: Display policy

Displays details about the selected policy.

S: Start policy

Makes the selected policy the active policy.

The policy switch must be confirmed before it is executed.
**INGPLEX CDS**

*P: Switch alternate CDS to primary CDS

Makes the alternate CDS the primary one. Because an alternate CDS is no longer available after the switch, SA z/OS shows a confirmation panel before the action is performed. On the panel, you can specify a new alternate CDS. When CDS recovery is switched on and you do not supply your own alternate CDS, SA z/OS tries to allocate a new alternate CDS automatically. The special requirements for manual and automatic creation of the new alternate CDS are the same as those for the replacement of the alternate CDS (action code A).

**Examples**

The following example illustrates the switch from the primary to the alternate CDS.

The following examples start with issuing INGPLEX CDS and pressing F8 on the CDS command dialog to scroll down the CDS list. The panel shown in Figure 68 is displayed.

```
Figure 68. INGPLEX CDS Command Dialog Panel
```

The panel header contains sysplex-related information about the system that the INGPLEX command was executed on. The details are as follows:

- **The System field** shows the name of the system.
- **The Interval field** shows the system failure detection interval in seconds. This interval is the amount of time XCF lets elapse without a status update before assuming that the system failed.
- **The OPNotify field** shows the number of seconds that XCF waits before notifying the operator of a potential system problem.
- **The Maxmsg field** shows the default value for the maximum amount of kilobytes of message buffer space. This default value is used when MAXMSG is not specified on SETXCF START commands.
- **The Cleanup field** shows the number of seconds that XCF waits for cleanup of members.
The **retry** field shows the default value for the retry limit. This value is used when the RETRY keyword is not specified on SETXCF START commands.

The **classlen** field shows the default length (in bytes) of messages allowed for a transport class. This value is used when CLASSLEN is not specified on the SETXCF START CLASSDEF command.

The **max cfi level** field shows the maximum CFLEVEL supported by this system. This system can connect to a coupling facility with a higher CFLEVEL than the value of **max cfi level**, but would not be enabled to use any functions supported by the higher level coupling facility.

The **couplexx** field shows the COUPLExx Parmlib member used for system IPL.

The **SMRBLD** field shows whether (value 1) or not (value 0) system-managed rebuild has been activated in the CFRM couple data set.

The **max sm level** field shows the maximum system-managed process level supported by this system.

The main part of the screen shows information about the primary and alternate CDSs for every CDS type. Press F8 to scroll and display further entries. The **MS** field shows the maximum number of systems that are supported by the CDS.

**Making an Alternate CDS the Primary CDS**

In this example, the alternate LOGR couple data set is made the new primary CDS. A new alternate CDS is automatically generated.

To switch the LOGR couple data set, enter **P** before LOGR on the panel displayed in [Figure 68 on page 190](#) and press Enter. INGPLEX CDS displays the confirmation panel shown in [Figure 69](#).

Use this panel to determine how a new alternate CDS is to be created after the switch. You can either specify your own new alternate CDS or let SA z/OS create it for you. When you specify the new alternate CDS yourself, the data set must
exist and must have been formatted with the XCF formatting tool. Automatic creation requires that spare volumes have been defined for LOGR couple data sets.

Pressing F10 causes SA z/OS to generate the new alternate CDS. After returning to the CDS command dialog, refreshing the panel, and scrolling down with F8, the panel looks as shown in Figure 70.

The previous alternate LOGR CDS has become the primary, and there is a new alternate, which was created by SA z/OS.

**Switching the CFRM Policy**

In this example, the active CFRM policy is switched.

Enter D before CFRM on the panel displayed in Figure 68 on page 190 and press Enter. The panel shown in Figure 71 on page 193 is displayed.
The panel shows information about the names and locations of the CDSs. The panel also shows the parameters that were used by the formatting tool of XCF for the allocation of the CDS. The **POLICY** column, for example, displays the maximum number of policies the CDS can contain. Furthermore, the panel shows information about the policies in the CDS, for example, how many coupling facilities and structures are defined in every policy, and which policy is currently active.

To switch to the HIRPOL policy, enter `S` before this policy and press Enter.

**Displaying the Channel Paths for a CDS Type**

In this example, the channel paths for the CFRM couple data sets are displayed.
Enter C before CFRM on the panel displayed in Figure 68 on page 190 and press Enter. The panel shown in Figure 73 is displayed.

<table>
<thead>
<tr>
<th>System</th>
<th>T</th>
<th>DEVN</th>
<th>CHPIDs</th>
<th>SSID</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEY1</td>
<td>P</td>
<td>260A</td>
<td>E4=E5=E2=E3=+</td>
<td>2600</td>
</tr>
<tr>
<td>A</td>
<td>2610</td>
<td>E4=E5=E2=E3=+</td>
<td>2600</td>
<td></td>
</tr>
<tr>
<td>KEY2</td>
<td>P</td>
<td>260A</td>
<td>E4=E5=E2=E3=+</td>
<td>2600</td>
</tr>
<tr>
<td>A</td>
<td>2610</td>
<td>E4=E5=E2=E3=+</td>
<td>2600</td>
<td></td>
</tr>
<tr>
<td>KEY3</td>
<td>P</td>
<td>260A</td>
<td>E4=E5=E2=E3=+</td>
<td>2600</td>
</tr>
<tr>
<td>A</td>
<td>2610</td>
<td>E4=E5=E2=E3=+</td>
<td>2600</td>
<td></td>
</tr>
<tr>
<td>KEY4</td>
<td>P</td>
<td>260A</td>
<td>13=22=30=94=+</td>
<td>2600</td>
</tr>
<tr>
<td>A</td>
<td>2610</td>
<td>13=22=30=94=+</td>
<td>2600</td>
<td></td>
</tr>
</tbody>
</table>

Figure 73. Channel Path Information for CFRM Couple Data Sets

- The **System** field shows the name of the sysplex members.
- The **T** field (for 'type') indicates whether the CDS is the primary (value 'P') or alternate (value 'A').
- The **DEVN** field displays the number of the device that the CDS resides on.
- The **CHPIDs** field shows the status of the paths to the devices in the format `chipid=status_code`. The codes are those of the operating system. They have the following meaning:
  + The path is logically and physically available and I/O on the path was successful.
  * The path is physically, but not logically available. The subchannel's logical path indicator is off but I/O to the path is successful. You can use the command `VARY PATH (ddd,nn),ONLINE` to make channel path `nn` logically available to device `ddd`.
  - The path is neither logically nor physically available. The subchannel's logical and physical indicators are both off for this channel path. You can use the command `CONFIG CHP(nn),ONLINE` to make the channel path logically available to all devices connected to the channel.
  & The device is reserved to another path. This indicator applies to devices with the dynamic pathing selection feature.
  < The path is installed but not physically available. The start subchannel request received a condition code of 3.
  > The device microcode has detected an error and will not allow I/O to complete on the path.
  B The path is unable to communicate. The device indicates that a busy or reserve condition exists on the path.
  C A controller error occurred while accessing the device.
D  A device error occurred while accessing the device.
I  Intervention is required; the device is not ready.
R  The path is available and the device is reserved to this path/group. This only applies to devices with the dynamic pathing feature.
T  A time out has occurred; there is no response from the device. The cause of the time out is undetermined and this condition is transient.
U  A storage control unit or storage director error occurred while accessing the device.
X  Unable to determine the failing unit.
•  The SSID field displays the storage subsystem that the device belongs to.

**INGPLEX SYStem**

**Purpose**

The SYSTEM function displays the target sysplex name, its GRS mode and its member systems.

**Example**

[Figure 74](#) shows an example of the INGPLEX SYSTEM command dialog panel.

```
AOFX100  SA z/OS - Command Dialogs  Line 1 of 4
Domain ID = IPSFP  ------ INGPLEX SYSTEM ------ Date = 27/01/06
Operator ID = NETOP1 Time = 16:29:12
Sysplex .. . . . . : KEY1PLEX
GRS Mode .. . . . : STAR
Display more info: C CPU E ETR I IPL 0 IOS S STOR/ESTOR
Signalling Path :  D device T structure
Monitor 

<table>
<thead>
<tr>
<th>Cmd</th>
<th>System</th>
<th>Status</th>
<th>Timestamp</th>
<th>INTERVAL</th>
<th>Action</th>
<th>TIME</th>
<th>WEIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>KEY1</td>
<td>ACTIVE</td>
<td>16:28:34</td>
<td>86400</td>
<td>ISOLATE</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>KEY2</td>
<td>ACTIVE</td>
<td>16:28:35</td>
<td>86400</td>
<td>ISOLATE</td>
<td>50</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>KEY3</td>
<td>ACTIVE</td>
<td>16:28:34</td>
<td>86400</td>
<td>ISOLATE</td>
<td>50</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>KEY4</td>
<td>ACTIVE</td>
<td>16:28:36</td>
<td>86400</td>
<td>ISOLATE</td>
<td>50</td>
<td>15</td>
</tr>
</tbody>
</table>
```

[Figure 74. INGPLEX SYSTEM Command Dialog Panel 1](#)

The following command codes are available:

- **C** Displays the online or offline status of one or more processors and any vector facilities, or ICRFs attached to those processors.
- **E** Displays the timer synchronization mode and ETR ports.
- **I** Displays IPL information.
- **O** Displays IOS-related configuration information.
S Displays the number of megabytes of central and expanded storage assigned and available to the system.

D Displays the device number of one or more inbound or outbound signalling paths that XCF can use and information about inbound or outbound XCF signalling paths to this system.

T Displays detailed signalling path information for all coupling facility structures.

The following information is displayed:

- The **Sysplex** field shows the name of the sysplex.
- The **GRS Mode** field shows the GRS mode of the target system. The mode can be either STAR or RING.
- The **Cmd** field allows you to specify command codes. To use one of the command codes shown, type the appropriate letter next to the resource name, and press Enter.
- The **System** column shows the name of the system.
- The **Status** column shows the status of the system.
- The **Monitor Timestamp** column shows the last time stamp recorded for status monitoring on this system.
- The **INTERVAL** column shows the system failure detection interval in seconds. This interval is the time XCF allows to elapse without a status update before assuming that the system failed.

The last three columns contain configuration data of the SFM policy (if applicable).

- The **SSUM Action** field shows the SSUM action. It can be one of the following:
  - ISOLATE
  - DEACTIVATE
  - RESET
  - PROMPT
  - N/A
- The **SSUM TIME** field shows the SSUM interval as specified in the current SFM policy.
- The **SSUM WEIGHT** field shows the SSUM weight specified in the current SFM policy. This value is used in sysplex reconfigurations after a signalling connectivity failure.

Use the PF11 key to view WLM-related information to the right, as shown in Figure 75 on page 197.
The **Timestamp** field shows the last time stamp when capacity data was queried from WLM.

- The **SUs Total** field shows the number of available SUs.
- The **SUs Used** field shows the number of used SUs in the last 10 minutes.
- The **Resource** field shows the number of SU-consuming resources with a desired state of AVAILABLE.
- The **SUs Exp.** field shows the number of free SUs, taking into account the resources that SA z/OS is about to start or stop.

**INGPLEX CONsole**

**Purpose**

The CONSOLE function displays the following information for the target sysplex:

- The name of the master console
- WTO & WTOR buffer utilization
- Number of queued messages (replies) of various types
- Awaiting mounts
- Operator requests and list of consoles (name, status, authority, number of WTOR buffers, UD, device, system, ALTGRP, MSCOPE)
Use one of the following command codes to get more information for the selected console or consoles:

**D**  Displays details for the console

**R**  Displays current requests for the console

The fields on the command dialog panel display the following information:

- The **Sysplex** field shows the name of the sysplex.
- The **Message Buffer Usage** field shows the limit of the number of WTO message buffers allowed outstanding.
- The **Awaiting Replies** field shows a decimal number representing the number of messages awaiting replies.
- The **Immediate Action** field shows a decimal number representing the number of outstanding immediate action messages (with descriptor codes 1 or 2). If the number is greater than 99999, asterisks appear in this field.
- The **Critical Action** field shows a decimal number representing the number of outstanding critical eventual action messages (with descriptor code 11). If the number is greater than 99999, asterisks appear in this field.
- The **Master Console** field shows the name of the master console.
- The **Reply Buffer Usage** field shows the limit of the number of WTOR message buffers allowed outstanding. The maximum value of yyyy is specified by the RMAX parameter in the CONSOLxx parmlib member.
- The **Eventual Action** field shows a decimal number representing the number of outstanding eventual action messages (with descriptor code 3). If the number is greater than 99999, asterisks appear in this field.
- The **Awaiting Mounts** field shows a decimal number representing the number of outstanding mount requests.
- The **Operator Requests** field shows a decimal number representing the number of outstanding requests for operator intervention.
The following details are shown for each MCS console that has been defined. If the MSCOPE information does not fit on the primary screen, the PF10 function key is available for you to toggle between the primary panel and the panel showing MSCOPE information.

- The Console field shows the name of the console as specified in the CONSOLxx parmlib member.
- The Status field shows the status of the console. The following values can occur:
  
  **HARDCOPY**
  Hardcopy log. This condition is only indicated if the console is active on the system where the command processes.

  **ACTIVE**
  Active console

  **ACTIVE-P**
  In the process of becoming an active console. This condition is only indicated if the console is active on the system where the command is processing.

  **MASTER**
  Master console

  **INACTIVE**
  Inactive console

  **INACT-P**
  In the process of becoming a non-active console. This condition is only indicated if the console is active on the system where the command is processing.

  **PROB-DET**
  The active system console is in the problem determination mode. PD is indicated only for the system console.

  **SUBSYS**
  Subsystem-allocatable console

- The AUTH field shows which commands may be entered from this console. The following values can occur:
  
  **ALL**
  Any INFO, SYS, IO, or CONS command may be entered from this console.

  **CONS**
  INFO commands and any commands from the console command group may be entered from this console.

  **INFO**
  Any command from the informational command group may be entered from this console.

  **IO**
  INFO commands and any commands from the I/O Control command group may be entered from this console.

  **MASTER**
  The specified console is authorized to enter any operator command.

  **NONE**
  This console does not have command authority.

  **SYS**
  INFO commands and any commands from the system control command group may be entered from this console.

- The NBUF field shows the number of WTO message buffers currently queued to this console. If nnnn is greater than 9999, asterisks (****) appear in this field.
INGPLEX CONSOLE

- The UD field shows whether this console is receiving messages with the UD attribute.
- The Device field shows the device number of the console as specified in the
  CONSOLxx parmlib member.
- The System field shows the system name of the active console.
- The ALTGRP field shows the alternate group defined for this console.
- The MSCOPE field lists the name of the system or systems that this console is
  receiving unsolicited messages from. Note that these systems might be different
  from the system where this console is physically attached.

INGPLEX IPL

Purpose

With the INGPLEX IPL command you can view and compare the IPL information
of the operating system. The IPL information is collected by the PROCLIB member
(for further information, refer to the installation procedure step "Configure
SYS1.PARMLIB Members" in "Configuring SA z/OS on Host Systems" of IBM
Tivoli System Automation for z/OS Planning and Installation). If a system does not
behave after IPL as expected, the IPL recording function enables you to identify
parameters that were changed, for example, since the last IPL. The recording
function enables you to compare different IPL scenarios. INGPLEX IPL is a tool
that helps to identify and resolve the cause of startup problems. The following
information can be displayed:
- The selected system (or blank)
- The name of the sysplex
- The maximum number of IPLs that are stored for each system
- An indicator showing whether comments in PARMLIB members are ignored
  when collecting information

Example

```
INGLX200 SA z/OS - Command Dialogs Line 1 of 6
Domain ID = IPSFM ------- INGPLEX IPL ------- Date = 02/22/02
Operator ID = NETOP1 Time = 17:59:27

System . . . . . . Max. number of IPL records/system : 10
Sysplex . . . . . . Suppression of PARMLIB comments : N

Cmds: C compare record / D display details / E erase record

System IPL Timestamp Dev Volume OpSys Release FMID
-------- ---------------- ---- ------ ------ -------- --------
KEYA 2002-02-22 13:52 770E 120204 z/OS SP7.0.2 HB87705
" KEYA 2002-02-09 09:28 770E 120204 z/OS SP7.0.2 HB87705
" KEYA 2002-02-08 15:28 770E 120204 z/OS SP7.0.2 HB87705
" KEYA 2001-12-10 14:31 0707 120147 z/OS SP7.0.2 HB87705
" KEYA 2001-12-10 14:53 0707 120147 z/OS SP7.0.2 HB87705
" KEYB 2002-02-22 13:59 770E 120204 z/OS SP7.0.2 HB87705
" KEYB 2002-02-14 16:24 770E 120204 z/OS SP7.0.2 HB87705
" KEYB 2002-02-11 18:46 770E 120204 z/OS SP7.0.2 HB87705
" KEYB 2002-02-11 15:36 770E 120204 z/OS SP7.0.2 HB87705
" KEYB 2002-02-11 14:22 770E 120204 z/OS SP7.0.2 HB87705

Command ===>
F1=Help F2=End F3=Return F6=Roll
F8=Forward F9=Refresh F10=Previous F11=Next F12=Retrieve
```

Figure 77. INGPLEX IPL Main Panel
Use F10 and F11 to scroll through all available columns. SORT by column numbers is supported as well as the FIND and RFind command to locate information on the panel. You can also limit the display to a particular system by specifying the system name in the appropriate entry field.

The following command codes are available:

C  Compares the complete IPL information with another IPL record. A second panel will be displayed where you can select the second record.

D  Displays detailed information about this IPL record.

E  Erases the IPL information records. This action must be confirmed.

• The Sysplex field shows the name of the sysplex.
• The System field shows the name of the system in the sysplex.
• The IPL Timestamp field shows the date and time of the IPL. The format is YYYY-MM-DD HH:MM converted to local timezone.
• The Dev field shows the IPL device number.
• The Volume field shows the volume serial of the IPL device.
• The OpSys field shows the name of the operating system, for example, z/OS or OS/390®.
• The Release field shows the release level of the operationg system.
• The FMID field shows the FMID of the operating system.

For further information about the panel fields refer to the online help.

INGPLEX SDUMP

Purpose

The INGPLEX SDUMP command lets you control the default dump options sysplexwide.

Example

The dump functions can be invoked directly by specifying the commands, or from the dump panel of the INGPLEX command selecting the appropriate command. In addition, you can invoke the dump submenu from the main panel of the INGPLEX command selecting command 7. The panel shown in Figure 78 on page 202 is displayed.
If you select option 1, the panel shown in Figure 79 is displayed.

The following command code is available:

**C change**

Invokes the modification panel by providing the options of the selected system as input:

- The **Sysplex** field shows the name of the sysplex.
- The **System** field shows the name of the system in the sysplex.
- The **Permission** field shows your authorization level.
- The **Dump options** field shows the default SDUMP options of all systems in the sysplex. For each system the following details are displayed:
INGPLEX SDUMP

Q= Shows whether or not SDUMP quiesces the system while dumping the contents of the SQA or CSA.

TYPE= Causes SVC dump to dump the cross memory address spaces that the caller has when SVC dump gets control (XMEM) or when the error causing the dump occurs (XMEME).

BUFFERS= Shows the reserved storage exclusively used by SVC dump. This storage can be used while capturing the contents of the common area storage.

MaxSpace Shows the maximum amount of virtual storage that SVC dump can use to capture volatile virtual storage data, summary dump data, and component-specific data before writing the dump to DASD.

MsgTime Shows how long (mm) the message IEA793A is shown at the console. When the system deletes the message, it also deletes the captured dump.

The FIND and RFIND commands are supported. If you specify command code C, the panel shown in Figure 80 is displayed.

Figure 80. INGPLEX SDUMP Modification Panel

The modification panel allows you to modify all SDUMP options. Furthermore, you can delete SDUMP options. After entering your changes you can set the new options for:
• The selected system
• All systems in the sysplex
• Selected systems in the sysplex

To set the options press the appropriate F-key. If you want to modify selected systems in the sysplex, you are prompted for the systems that the SDUMP options are being changed for. To reset the options to the state when the modification panel was invoked press F5 Undo all.
Note: The user must be authorized to change any SDUMP option. The authorization can be any of those that are used for controlling coupling facilities and couple data sets.

For further information about the panel fields refer to the online help.

INGPLEX SVCdump

Purpose

The INGPLEX SVCDUMP function allows you to issue a multisystem dump of up to 15 address spaces of a single system including their data spaces and structures.

Example

The following systems of sysplex KEY1PLEX are registered to the automation. Use any non-blank character to select one system and then press ENTER.

```
Sel  System
---  ------
   KEY2
   KEY3
   KEY4
```

Figure 81. INGPLEX SVCDUMP Target System Selection Panel

- The Sel field lets you select a system that a multisystem dump is issued from.
- The System field shows the name of the system having joined the same XCF group the operator is logged on to.

For further information about the panel fields refer to the online help. After selecting a system and pressing Enter, the panel shown in Figure 82 on page 205 is displayed.
If, for example, you select the VTAM address space and the WATS address space (which is a user), press Enter and then press F5, the panel shown in Figure 83 is displayed.

Address space VTAM has some data spaces (D), one list structure (L) and some XCF group members (M). TSO user WATS has nothing.

The following command codes are supported:

- **D**  Deselects the previous selection.
- **S**  Selects a local address space, data space, structure, or XCF group member address space for the SVC dump.
If you press F5, the dump option selection panel is displayed, as shown in Figure 84.

The panel shows the default dump options that are set on invocation. After specifying the dump title, press F5 to issue the dump. When the dump is taken, the function returns to the address space selection panel with all selections cleared. The SORT, FIND and RFIND commands are supported for selection panels only. For further information about the panel fields refer to the online help.

Executing this command in line-mode requires the following 5 steps:

1. Execute the command:
   `INGPLEX SVC sysname [REQ=LIST] OUTMODE=LINE`

2. Remove the first 5 lines and the last line from the output. Then select the address spaces that you want to dump and save the entries using a PIPE command:
   `PIPE ... | SAFE INGRX260`

   Keep the same format that is returned in the REQ=LIST output.

3. Execute the command:
   `INGPLEX SVC sysname REQ=COMP OUTMODE=LINE`

4. Again remove the first 5 lines and the last line from the output. Then select the address spaces and the associated components, such as data spaces, XCF group members, and cache and list structures and save these entries using a PIPE command:
   `PIPE ... | LIT \dump title/sdata options/stropt\ |SAFE INGRX260`

   Keep the same format that is returned in the REQ=COMP output. The dump options passed by the LIT stage correspond exactly to the values in Figure 84.

5. Execute the command:
   `INGPLEX SVC sysname REQ=DUMP OUTMODE=LINE`
INGPLEX SLIP

Purpose

With the INGPLEX SLIP command you can display serviceability level indication processing (SLIP) traps being set at all systems in the sysplex. With INGPLEX SLIP you can view, enable, disable, and delete the SLIP trap defined in the sysplex.
Example

The following command codes are available:
+ Shows the settings of the SLIP trap.
- Hides the settings of the SLIP trap.
D Disables the SLIP trap.
E Enables the SLIP trap.
R Deletes the SLIP trap.

The SORT, FIND and RFIND commands are supported.

Note: The user must be authorized to enable, disable, and delete a SLIP trap. The authorization can be any of those that are used for controlling coupling facilities and couple data sets.

For information about the panel fields refer to the online help.

INGRELS

Purpose

The INGRELS command shows the relationships that are defined for the given resource.

Syntax

```
>>INGRELS resource
```

```plaintext
TARGET=system_name
```
```
DOMAID=domain_id
```
```
SYSPLEX=sysplex_name
```
```
OUTDSN=dsname
```
```
OUTMODE=line
```
```
AUTO
OUTLOG
```
Parameters

resource

Specifies the name of the resource to be displayed. The format is
name/type[/system].

TARGET

For information on the TARGET parameter, see “TARGET Parameter” on page 10.

OUTDSN

For information on the OUTDSN parameter, see “OUTDSN Parameter” on page 13.

OUTMODE

For information on the OUTMODE parameter, see “OUTMODE Parameter” on page 12.

WAIT

Specifies the number of seconds to wait before reporting that a timeout
occurred if the automation manager does not provide the requested data. The
maximum time interval is 999 seconds.

If omitted, the time interval is 30 seconds.

SHOW

Specifies whether or not to show just the relationships or also the relationship
conditions. It can be:

EXPANDED

This shows relationships and their conditions. This is the default.

COLLAPSED

This shows just the relationships.

Restrictions and Limitations

None.

Usage

The INGRELS command operates sysplexwide. For an overview see “Overview of
Commands that Operate Sysplexwide” on page 9.

Examples

If you enter inrels and the name of a resource, a panel similar to that in
Figure 87 on page 210 is displayed.
The `Cmd` field lets you specify command codes. To use one of the command codes shown, type the appropriate letter next to the resource name and press Enter. The following command codes are available:

+ Shows the condition details that exist for the selected relationship. This is only applicable if the relationship conditions are collapsed.

- Collapses the condition details so that only the relationship is shown.

E Shows the requests and votes of the resource (it displays the INGVOTE panel for the resource).

F Shows detailed information about the resource (it displays the INGINFO panel for the resource).

J Displays the INGGROUP panel for the resource.

S Focuses on the selected resource. This is equivalent to specifying the selected resource name in the Resource field at the top of the panel.

- The `Name` field shows the name of the resource that the specified resource has a relationship with. This is referred to as the relationship partner and is also called the supporting resource.
- The `Type` field shows the type of the resource.
- The `Target` field shows the name of the system where the resource resides.
- The `Dir` field shows the direction of the relationships as seen from the specified resource.

F  Forward. The relationship exists from the specified resource to the partner resource.

B  Backward. The relationship exists from the partner resource to the specified resource.
The \textit{Relationship} field shows the name of the relationship that exists between the two resources. This defines the dependency that exists between the two resources. The status of all descendants will be considered for a relationship with strong chaining. For weak chaining only the direct descendants are checked. Weak is the default. Strong chaining causes SA z/OS to check all relationships of the involved resources along the dependency graph.

If the relationship name contains 'active', this means that an attempt is made to bring the supporting resource in the required state to satisfy the dependency. Active is the default.

The following PF keys are supported:
\begin{itemize}
  \item Use the PF5 key to collapse the visible relationship conditions for all relationships.
  \item Use the PF11 key to see the relationship condition details for all relationships.
\end{itemize}

If you collapse all condition details, the INGREL\textsl{S} panel looks like \textit{Figure 88}.

![INGREL\textsl{S} Command Dialog Panel with All Details Collapsed](image)

\textbf{INGREQ}

\textbf{Purpose}

The INGREQ command lets you:
\begin{itemize}
  \item Initiate the shutdown process for one or more resources
  \item Initiate the startup process for one or more resources
  \item Cancel a request that was previously made for the specified resource
\end{itemize}
Syntax

To start up or shut down resources:

```
<table>
<thead>
<tr>
<th>INGREQ</th>
<th>resource</th>
<th>REQ</th>
<th>START</th>
<th>startup parms</th>
</tr>
</thead>
<tbody>
<tr>
<td>NETVASIS</td>
<td>ALL</td>
<td></td>
<td>UP</td>
<td>shutdown parms</td>
</tr>
<tr>
<td>PRI=LOW</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRI=FORCE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRI=HIGH</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXPIRE=(date, time)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIMEOUT=(interval)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSG</td>
<td></td>
<td></td>
<td>CANCEL</td>
<td></td>
</tr>
<tr>
<td>REMOVE= (AVAIL, DEGR, SYSGONE, UNKNOWN)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OVERRIDE= (ALL)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRECHECK=YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VERIFY=YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SOURCE= source</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRECHECK=NO</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VERIFY=NO</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TARGET= system_name</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OUTMODE= LINE</td>
<td></td>
<td></td>
<td>AUTO</td>
<td></td>
</tr>
<tr>
<td>APPLPARMS=</td>
<td></td>
<td></td>
<td>NETLOG</td>
<td></td>
</tr>
<tr>
<td>WAIT=YES</td>
<td></td>
<td></td>
<td>CMT= text</td>
<td></td>
</tr>
<tr>
<td>WAIT=NO</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WAIT=nnn</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
**INGREQ**

**Startup parms:**
- SCOPE=ONLY
- TYPE=NORM
- SCOPE=ALL
- TYPE=IMMED
- START=NO
- START=YES
- INTERRUPT=NO
- INTERRUPT=YES

**Shutdown parms:**
- SCOPE=ALL
- TYPE=NORM
- TYPE=IMMED
- RESTART=NO
- RESTART=YES
- INTERRUPT=NO
- INTERRUPT=YES

**To cancel a request:**
- INGREQ resource REQ=CANCEL
- SOURCE=source
- VERIFY=YES
- VERIFY=NO
- WTOR
- WAIT=YES
- WAIT=NO
- WAIT=nnn
- TARGET=system_name
- OUTMODE=LINE
- AUTO
- NETLOG

**Feedback option:**
- FDBK=(MSG,userid,max_time,exp_status)

**Feedback option:**
- FDBK=(MSG,userid,max_time,exp_status)

**Parameters**

**NETVASIS**
Prefix the INGREQ command with NETVASIS if you want to pass the description text in lower or mixed case.

**resource**
Specifies the name of the resource to be processed. The format is `name[/type[/system]]`. Wildcards are supported. In fullscreen mode, if more than one resource matches the wildcard specification, a selection panel is displayed.
In line mode, if more than one resource matches the wildcard specification and MULT=YES is specified, all selected resources are processed. Multiple resource specification in line mode without MULT=YES causes an error message to be displayed.

To shut down all resources for a system, specify name/SYG/name, where name is the system name. Alternatively, you can specify ALL.

**Note:** You need not specify the full name of the resource. This applies to the system as well as to the type. If the resource name is unique within the name space of the automation manager, the system name need not be specified. If the resource name is unique within the appropriate types, the type need not be specified. A valid resource name could be, for example, CICSPROD.

**REQ**
Specifies the request to be carried out. It can be either START or STOP. Alternatively you can use UP or DOWN.

Specify CANCEL if you want to cancel a request.

**PRI**
Specifies the priority given to the request. It can be:

- **FORCE**
  Takes precedence over requests posted with HIGH priority, for example, INGREQ=START PRI=FORCE overrules an INGREQ=STOP PRI=HIGH request.

- **HIGH**
  High priority.

- **LOW**
  Low priority. This is the default.

**EXPIRE**
Specifies the expiry date and time of the request. The request is automatically removed when it expires. The date format is yyyy-mm-dd. The time format is hh:mm.

Alternatively, you can specify a relative time, for example, +04:00. The relative time will be automatically converted into the absolute date/time. The maximum relative time interval is 24:00 hours.

**TIMEOUT**
Specifies the interval used to check whether the request has been successfully completed. Either a message is issued, or the request is cancelled if the request has not been satisfied after that time. The format is mm.

**MSG**
Specifies that a message should be issued telling the notify operator that the request was not satisfied within the expected time interval. The default is to issue a message.

**CANCEL**
Specifies that the request should be canceled.

**JOB**
Indicates the specified resource is a job name. The job name can contain a wildcard.

**REMOVE**
Indicates the condition when the request is automatically removed, regardless of whether the request is satisfied or not. If the specified condition matches the observed status of the resource, the request is deleted.
When starting a resource and specifying REMOVE=AVAILABLE, this causes the request to be removed from the system as soon as the resource becomes available.

As soon as the observed status of the resource becomes DEGRADED, the start request is removed. Thus there will be no requests outstanding for the resource and a INGREQ ALL will go through even with low priority.

If the system where the automation agent runs leaves the sysplex, the observed status of all resources running on that system becomes SYSGONE.

If the automation agent stops, the observed status of all resources controlled by the automation agent becomes UNKNOWN.

Alternatively you can specify an asterisk (*) or NO to reset the installation-defined default.

`OVERRIDE` specifies the overrides to be considered for the request. These can be one or more of the following:

- `NO` specifies that no override is done. For example, a resource will not be started unless its startup flag is set on and its trigger, if used, is satisfied. This is the default.

- `ALL` sets the DPY, FLG, STS, and TRG flags to on. For startup this means that the resource is made available regardless of the automation flag settings, its trigger setting or the state of the start dependencies defined for the resource. For shutdown this means that the resource is made unavailable regardless of its automation flag settings, the trigger state and the state of the stop dependencies.

- `DPY` causes the status of the relevant START/STOP dependencies that are defined for the resource to be ignored.

  Note: This parameter should only be used with STOP SCOPE=ONLY.

- `FLG` the appropriate automation flags will be ignored for the resource.

- `STS` the current observed status of the resource is ignored by the agent during INGREQ precheck processing. Override STS does not affect automation manager processing. Therefore, the automation manager may still reject the request after the agent has forwarded it to the automation manager.

- `TRG` the current trigger settings will be ignored when determining whether or not the resource can be started or stopped.

`PRECHECK` specifies whether or not the startup or shutdown process should pre-validate any actions before actually performing them. The value can be:
YES
  Validation is performed for the entry resources. This is the default.

NO  Validation is not performed before the command is issued.

The following validation is done for each resource that is to be affected by the INGREQ request:
  • START or STOP ability, depending on current status
  • Automation flag checks, depending on the request type
  • Extra feature (CICS or DB2) start or stop ability checks

These checks are made on each automation agent that hosts a specific resource before any automation manager involvement in the request. Any failure that is detected during the precheck phase results in an INGREQ request failure. This is indicated by a non-zero return code and a relevant error message that is issued and delivered as set by the OUTMODE parameter.

VERIFY
  Specifies whether the startup or shutdown process should be verified. The following specifications are valid:
  YES  Depending on the NetView task type that the INGREQ command is running on, the verification is either displayed on a fullscreen panel or, in line mode, shown on the console. The latter is also done when OUTMODE=LINE is coded.

WTOR
  The list of affected resources is displayed before the startup or shutdown begins. The operator is prompted to verify the list via WTOR.

NO  No verification is performed. VERIFY is set to NO for unattended tasks. Otherwise it is set to YES.

The default depends on the type of task that the INGREQ command runs on.

SOURCE
  Specifies the originator of the request. The default is OPERATOR for an OST and AUTOOPS for an autotask. When REQ=CANCEL, this parameter is used as a filter. Only those requests from the named source are cancelled. Wildcards are supported, for example, OP*. An asterisk (*) matches a string of arbitrary length and a percentage sign (%) matches a single character.

WAIT
  Specifies whether or not to wait until the request is complete. The default is YES.

  The variable nun is the number of seconds to wait before giving up and reporting that a timeout has occurred. The maximum time interval is 999 seconds.

CMT
  Specifies descriptive information about the request. The maximum length is 120 characters. If the text contains embedded blanks, it must be enclosed in quotation marks or parentheses.

TARGET
  For information on the TARGET parameter, see "TARGET Parameter" on page 10.
OUTMODE
For information on the OUTMODE parameter, see “OUTMODE Parameter” on page 12.

APPLPARMS
The content of APPLPARMS is put into the &APPLPARMS variable.

MULT
Specifies that more than one resource will be accepted when running in line mode, if the resource wildcard specification results in multiple resources.

SCOPE
Specifies whether the startup or shutdown affects a resource or its descendants, or both. Valid options are:

- **ONLY**
  Specifies that only the specified resource is started or stopped. This is the default for startup.

- **ALL**
  Specifies that the resource and its descendants are started or stopped. This is the default for shutdown.

- **CHILDREN (shutdown only)**
  Specifies that only the descendants of the resource are stopped.

TYPE
Specifies the type of the startup or shutdown. Each type must be defined in the policy database. Valid options are:

- **NORM (startup or shutdown)**
  Specifies a normal startup or shutdown. This is the default.

- **IMMED (shutdown only)**
  Specifies an immediate shutdown.

- **FORCE (shutdown only)**
  Specifies a forced shutdown.

- **user (startup only)**
  Specifies a user-defined startup type.

Note: If no startup commands are defined for the specified start type, the startup commands for start type NORM are issued.

Note: The value specified here will be overridden by the INGSET STARTTYPE/STOPTYPE value if previously specified. However, a stop type of FORCE, wherever specified, will always be honored.

Use the VERIFY option to view the TYPE value that will be used for this request.

Once a shutdown has got to a PROBLEM or DENIED state, you can submit another shutdown request against the resource, specifying a different shutdown type or override parameters or both.

If you want to escalate the shutdown of a child subsystem, you can issue the INGREQ directly against it, but you must use a higher priority than the shutdown request against the parent was made with.
**RESTART (shutdown only)**
Specifies whether the resource should be restarted automatically when it has been shutdown completely. Valid options are:
- **NO** – Specifies that no restart is performed. This is the default.
- **YES** – Specifies that a restart is performed.

**INTERRUPT (shutdown only)**
Specifies whether or not to interrupt the startup or shutdown phase of the resource. Valid options are:
- **NO** – If the resource is starting, the automation manager waits for the resource's UP message before proceeding with the shutdown.
  
  If the resource is shutting down, the automation manager waits until the shutdown is complete. This is the default.
- **YES** – If the resource is starting, the automation manager interrupts the startup process and starts shutting down the resource.
  
  Interrupting the shutdown process means breaking the former stop request that is in a locked state because the agent has not yet indicated that the shutdown is complete. A typical scenario where the INTERRUPT=YES option is used is when escalating to a higher stop type.

**FDBK**
Specifies the feedback parameter that causes the final result of the command to be reported back to the designated instance.

**MSG**
Causes message signaling of the successful or unsuccessful completion of the command to be performed.

The message is sent back to the originator of the command. The originator is determined by the NetView OPID(S) function unless the user ID is specified. Depending on the status, either message ING300I or ING301I is issued. The message text contains the command that was processed successfully or failed.

**userid**
Identifies the NetView user ID of the person to be notified if different to the originating user.

Use the following values to specify when to issue the message:
- **G** – Issue the message only when the resource has reached its expected status.
- **F** – Issue the message only if the resource did not reach its expected state in the specified time interval.
- **B** – Issue the message in all cases.

**WAIT**
To wait for the completion of the command (denoted by the max_time parameter). Note that the WAIT option is rejected when the INGREQ command executes on a work operator.

**max_time**
The time interval in NetView format (mm:ss, :ss, mm or hh:mm:ss) that SA z/OS will wait. If the specified resource has not reached the expected state, the command is considered to have failed and the operation is posted in error.
exp_status

The observed status that the resource should be in to consider command processing to be complete. The status can be abbreviated, for example, AV for Available. More than 1 status can be specified.

Restrictions and Limitations

To use the INGREQ command system operations must be initialized.

The FDBK parameter will only monitor resources within the local sysplex.

Security Considerations

The INGREQ command supports resource level security. If turned on, the following profiles in class SYSAUTO are checked:

<table>
<thead>
<tr>
<th>Profile</th>
<th>Authority</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGT.sysplex.xcfgrp.RES.resname.restype[.ressys]</td>
<td>UPDATE</td>
<td>Unless authority is CONTROL</td>
</tr>
<tr>
<td></td>
<td>CONTROL</td>
<td>When SCOPE, OVERRIDE, or INTERRUPT is used with a value other than the default (IBM supplied or installation default)</td>
</tr>
</tbody>
</table>

If multiple resources are affected multiple profiles are checked. For START and STOP requests the request is carried out or rejected as a whole. CANCEL requests are processed individually.

For further details, refer to IBM Tivoli System Automation for z/OS Planning and Installation.

Usage

The INGREQ command operates sysplexwide. For an overview see “Overview of Commands that Operate Sysplexwide” on page 9.

Examples

If you enter INGREQ a panel similar to Figure 89 on page 220 is displayed.
The Resource field shows the name of the resource to be processed. The format is name/type/system. Wildcard is supported.

The Target field shows the name of the system (system name, domain ID, or sysplex name) that the command should be routed to. Specifying this is only necessary if the resources are not part of the local sysplex.

The Request field shows the request to be carried out. This is either START or STOP. Alternatively you can use UP or DOWN. To remove a previously made request from the same source specify CANCEL.

The Type field shows the type of the startup or shutdown. Valid shutdown types are NORM, IMMED and FORCE as well as any other type defined in the policy database for the resource. Specify a question mark (?) to display the list of defined startup types.

The Scope field shows whether the startup or shutdown affects this resource or its descendants, or both.

The Priority field shows the priority given to the request.

The Expire field shows the expiry date and time when the request will be removed.

The AutoRemove field shows the condition when the request is automatically removed regardless of whether the request is satisfied or not.

The Restart field shows whether the resource should be restarted automatically when it has been shutdown completely. This applies only when requesting a shutdown of a resource.

The Override field shows the overrides to be considered for the request. The parameter can be used to bypass the conditions or settings that would otherwise prevent a resource, or resources, from starting or shutting down.

The Verify field shows whether the startup or shutdown process should be verified. If YES or WTOR is entered, the list of affected resources is displayed before the startup or shutdown begins. You will be prompted to verify the list. If WTOR is specified, you will be prompted via a WTOR. If NO is entered, the startup or shutdown begins immediately. The default is YES.

The Comment field lets you specify descriptive information, for example, why the request was made.
The **ApplParms** field shows modifications or parameters to the START or STOP request.

If you press PF11 a panel similar to Figure 90 is displayed.

![Figure 90. INGREQ Command Dialog Panel 2](image)

- The **Timeout** field shows the time period that the request must complete in. After this time interval either a message is issued, or the request is canceled.
- The **Precheck** field shows whether or not the startup or shutdown process should perform up-front validation.
- The **Interrupt** field shows whether or not a request to shut down a resource should interrupt the startup phase of a resource:
  - **YES**: The automation manager interrupts the startup process as soon as this stop request is submitted.
  - **NO**: If the resource is starting, the automation manager waits for the resource’s UP message before proceeding with the shutdown. This is the default.

---

### INGRPT

**Purpose**

The INGRPT command displays statistical information about the automation agent and some basic information about the automation manager. Two types of reports are produced:

- Summary report.
- Detail report.

**Syntax**

```
-- INGRPT [RESET] RESOURCE=name 
```

Chapter 2. System Operations Commands 221
INRPT

```
TARGET=
Dest_list
  ALL
OUTDSN=dsname
OUTMODE=
  LINE
  AUTO
  NETLOG
WAIT=YES
WAIT=nnn
```

**Report options:**

```
STATS=SUMMARY
STATS=DETAIL
NONE
```

**Destination:**

```
-system_name
-domain_ID
-sysplex_name
```

**Dest_list:**

```
( Destination )
```

**Parameters**

The following parameters are supported:

**RESET**

Resets the counters to 0. This option can be specified with the STATS parameter to get the statistics information and subsequently reset the counters.

**RESOURCE**

Is the name of the subsystem that you want to collect the statistics for, for example, TSO. Wildcard is supported, for example, INRPT STATS=D RESOURCE=abc*. The wildcard can be leading or trailing, for example *abc*.

**TARGET**

For information on the TARGET parameter, see "TARGET Parameter" on page 10.

**OUTDSN**

For information on the OUTDSN parameter, see "OUTDSN Parameter" on page 13.

**OUTMODE**

For information on the OUTMODE parameter, see "OUTMODE Parameter" on page 12.

**WAIT**

Specifies whether or not to wait until the request is complete. The default is YES.

`nnn` is the number of seconds to wait before giving up and reporting that a timeout has occurred. The maximum time interval is 999 seconds.
STATS
Specifies whether to generate a summary report or a detail report. Detail
statistics include information about each resource or subsystem that is
automated by the agent. The following options are valid:

**SUMMARY**
This is the default.

**DETAIL**
Specifies to show detailed information for each resource (subsystem in the
automation agent terminology).

**NONE**
No report is produced. The NONE option is only useful if the reset option
is specified.

Restrictions and Limitations
None.

Usage
Issuing ACF COLD resets the statistic counters.

The statistic counters for a subsystem removed with INGAMS REFRESH are not
erased. However, the statistic counters for a subsystem are cleared if ACF COLD or
INGRPT RESET is issued.

Examples
If you enter INGRPT you will see the start of a summary report for your system
that is similar to [Figure 91].

```
Domain ID = IPUN8 ---------- INGRPT ---------- Date = 10/31/03
Operator ID = NETOP1 System = AOC8 Time = 13:04:14

***************************** Summary Statistics *****************************
------------------------------------------------------------------------------
Begin of statistics : 10/30/03 11:32:25 Domain : IPUN8
End of statistics : 10/31/03 13:04:12 Sysplex : AOC8PLEX
Elapsed time (hours) : 25:32 (1532 min)

Automation Agent Statistics
CPU time used (seconds) : 344.87
Total number of resources defined : 69
Total number of resources managed : 22
Total number of messages automated : 0
Average number of messages per hour : 0.0
```

**Figure 91. INGRPT Command Dialog Panel**

This panel shows statistical information about the automation agent and some
basic information about the automation manager. The report shows information
that has been gathered during the SA z/OS session. The SA z/OS session begins
with either SysOps initialization or the last INGRPT RESET command (whichever
INGRPT

comes last) until the INGRPT STATS=SUMMARY/DETAIL command is issued. The counters are not retained across SysOps sessions.

The summary report for a particular subsystem shows:
• The total number of messages automated for the resource.
• The average number of messages automation per hour.
• The total number of resulting commands.
• The total number of Start commands issued.
• The total number of Stop commands issued.
• The total number of critical threshold reached hits.

INGRUN

Purpose

The INGRUN command lets you:
• Set a runmode for a system
• Add the qualification of one or more resources for any runmode
• Delete runmode qualification for one or more resources

Syntax

```
-- INGRUN --REQ-- SET-- mode-spec --TARGET-- system_name
                   ADD-- resource-spec
                   DEL-- resource-spec

OUTMODE=-- LINE
           AUTO
           NETLOG

WAIT=-- YES
      -- NO
      -- nnn

mode-spec:

--SYSTEM-- system
--RUNMODE-- runmode
--PERSISTENT-- YES
             -- NO

TYPE=-- NORM
      -- IMME
      -- FORCE

PRI=-- LOW
    -- FORCE
    -- HIGH

OVERRIDE=-- NO
          -- ( )
          -- ALL
          -- DPY
          -- FLG
          -- STS
          -- TRG
```
resource-spec:

Notes:
1 Line-mode only

Parameters

REQ
The request to be carried out.

SET Specifies a new runmode is set
ADD Qualifies one or more individual resources for a runmode
DEL Deletes the runmode qualification for the given resource

RUNMODE Specifies the runmode to be activated. The runmode must be specified in the policy database. In full-screen mode, a question mark(?) can be specified to display a selection list. Use *ALL to disable runmode consideration.

PERSISTENT Specifies whether the request should be kept across system IPL. Valid options are:

YES The runmode request is kept in the takeover file. This is the default setting.
NO The runmode request is removed as soon as the system leaves the sysplex. For more information, see REMOVE=SYSGONE of the "INGREQ" on page 211 command.

RUNRES Specifies the resource to be considered qualified for any runmode. The format is name/type[/system]. In full-screen mode, if more than one resource matches the specification, a selection panel is displayed. In full-screen mode, for a DEL request a question mark(?) can be specified to display a selection list of all resources previously specified on an ADD request.

MULT Specifies that more than one resource will be accepted when running in line mode, if the resource wildcard specification results in multiple resources. This command is available in line mode only.

NO The resource specification must resolve to a single resource.
YES Multiple resources can be processed with a single invocation of the INGRUN command.

SYSTEM Specifies the system for which the runmode is set.
INGRUN

TYPE
Specifies the type of shutdown to be used for resources that do not qualify for the new runmode. For more information, see the "INGREQ" on page 211 command.

PRI / PRIORITY
Specifies the priority for the shutdown of the resources that do not qualify for the new runmode. For more information, see the "INGREQ" on page 211 command.

OVERRIDE
Specifies the overrides that are to be considered when shutting down resources that do not qualify for the new runmode. For more information, see the "INGREQ" on page 211 command.

VERIFY
Specifies if the shutdown process of resources that do not qualify for the new runmode should be verified.

CMT / COMMENT
Specifies descriptive information, for example, why the request was made.

TARGET
For information on the TARGET parameter, see "TARGET Parameter" on page 10.

OUTMODE
For information on the OUTMODE parameter, see "OUTMODE Parameter" on page 12.

WAIT
Specifies whether to wait or not for the request to complete. The default is YES.

\textit{nnn} is the number of seconds to wait before giving up and reporting that a timeout has occurred. The maximum time interval is 999 seconds. For more information, see the "INGREQ" on page 211 command.

Restrictions and Limitations
To use the INGRUN command, system operations must be initialized.

Security Considerations
The INGRUN command supports resource level security. If turned on, the following profiles in class SYSAUTO are checked:

<table>
<thead>
<tr>
<th>Profile</th>
<th>Authority</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGT.sysplex</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.xcfgrp.RES.resname.restype[.ressys]</td>
<td>UPDATE</td>
<td>Always</td>
</tr>
</tbody>
</table>

If multiple resources are affected, multiple profiles are checked. For SET requests, this is the SYG resource corresponding to the selected system (system/SYG/system). The request is carried out or rejected as a whole.

For further details, refer to \textit{IBM Tivoli System Automation for z/OS Planning and Installation}. 
Usage

The INGRUN command operates sysplexwide.

Examples

If you enter ingrun a panel similar to Figure 92 is displayed.

--- Parameters for SET request ------------------------------------------
System => System name
Runmode => Runmode name (mode or ?)
Persistent => Keep request across IPL (YES/NO)
Type => Type of processing (NORM/IMMED/FORCE)
Priority => Priority of request (FORCE/HIGH/LOW)
Override => (ALL/NO/TRG/FLG/DPY/STS)
Verify => Check affected resources (YES/NO/WTOR)
Comment =>

--- Parameters for ADD or DEL requests -----------------------------------
Resource => format: name/type/system

Figure 92. INGRUN Command Dialog Panel

Request

Valid request types are SET, ADD, and DEL.

Target

Specifies the name of the system (system name, domain ID or sysplex name) that the command should be routed to. This is only necessary when you want to operate outside of the local sysplex.

The following parameters are valid for SET only:

System

Specifies the name of the system that you want to set a runmode for.

Runmode

Specifies the name of the runmode that you want to set. You can specify a question mark (?) to select a from a list of all available resources.

Persistent

Specifies whether the request should be kept across system IPL. For more information, see REMOVE=SYSGONE of “INGREQ” on page 211 command.

Type

Specifies the type of shutdown to be used for resources that do not qualify for the new runmode. For more information, see the “INGREQ” on page 211 command for further details.

Priority

Specifies the priority for the shutdown of the resources that do not qualify for the new runmode. For more information, see the “INGREQ” on page 211 command for further details.
**INGRUN**

**Override**
Specifications the overrides that are to be considered when shutting down resources that do not qualify for the new runmode. For more information, see "INGREQ" on page 211 for further details.

**Verify**
Specifies whether the shutdown process of resources that do not qualify for the new runmode should be verified. For more information, see "INGREQ" on page 211 for further details.

**Comment**
Specifies descriptive information, for example, why the request was made.

The following parameter is set for ADD and DEL requests:

**Resource**
Specifies the name of the resource for which you want to add or delete the qualification. The format is name/type/system or name/type for sysplex resources. Wildcards and incomplete specifications are supported. For DEL requests a question mark (?) can be specified which lets you choose from a list of resources previously specified on an ADD request.

---

**INGSCHED**

**Purpose**
INGSCHED displays a list of all defined schedules (service periods). It lets you:

- Update the time slots (this is referred to as a schedule override).
- Display the list of resources that are associated with the schedule.
- Delete schedule or resource overrides.

**Syntax**

```
/SM590000/SM590000

INGSCHED schedule schedule options resource resource options
REQ=TIMERS WAIT=YES WAIT=NO WAIT=nnn
/SM590000/SM590000

TARGET= system_name domain_id sysplex_name
OUTDSN=dsname OUTMODE=LINE AUTO NETLOG

schedule options:

-REQ=DISP DATE=curr_date DAYS=1
-REQ=BASE
-DEL date days
-DISP date days
-REPL date priority timeslots
-RESOURCES
```
resource options:

- **REQ=**
- **DISP**
- **DATE=**
- **curr_date**
- **DAYS=**

**date:**

- **DATE=**
- **curr_date**
- **YYYYMMDD**

**days:**

- **DAYS=**
- **1**
- **1..366**

**priority:**

- **PRI=**
- **LOW**
- **HIGH**
- **LOW**
- **HIGH**

**timeslots:**

- **UP=**
- **hhmm-hhmm**
- **DOWN=**
- **hhmm-hhmm**

**Parameters**

- **schedule**
  - Is the name of the schedule to be displayed.

- **resource**
  - Specifies the resource that the schedule overrides are to be displayed for. The format is *name/type/system*.

- **REQ**
  - Specifies the request. The request can be:
    - **BASE**
      - Displays the base schedule information.
    - **DEL**
      - Deletes the overrides for the specified date.
    - **DISP**
      - Displays the base schedule, schedule override or resource overrides starting with the specified date.
    - **REPL**
      - Replaces the schedule or resource override with a new one.
INGSCHED

RESOURCES
Displays the resources that use the specified schedule (service period).

TIMERS
Displays the list of timers that are currently scheduled.

DATE
Specifies the date in yyyymmd format that the delete or replace of the override should be done for.

curr_date
The current date.

DAYS
Specifies the number of days for which the overrides should be displayed or deleted.

PRIORITY
Specifies the priority to be assigned to the override. The first value is the UP priority and the second value is the DOWN priority. Valid values are:

HIGH
High priority.

LOW
Low priority.

Both priorities are optional. The default is LOW. The priority values can be abbreviated.

UP
Specifies the UP start and stop times in the format hhmm-hhmm. Up to five time slots can be specified separated by a comma. The UP time slots define the time windows where the resource must or should be available. A priority of HIGH means that the resource must be available, while a priority of LOW means that the resource should be available.

DOWN
Specifies the DOWN start and stop times. The format is hhmm-hhmm. Up to five time slots can be specified separated by a comma. The DOWN time slots define the time window where the resource must or should be not available (that is, unavailable). A priority of HIGH means that the resource must be unavailable, while a priority of LOW means that the resource should be unavailable. If you specify DOWN, specifying the UP priority becomes mandatory.

WAIT
Specifies whether or not to wait until the request is complete. The default is YES.

nnn is the number of seconds to wait before giving up and reporting that a timeout has occurred. The maximum time interval is 999 seconds.

TARGET
For information on the TARGET parameter, see "TARGET Parameter" on page 10.

OUTDSN
For information on the OUTDSN parameter, see "OUTDSN Parameter" on page 13.

OUTMODE
For information on the OUTMODE parameter, see "OUTMODE Parameter" on page 12.
Restrictions and Limitations

The INGSCHED command can only be used when system operations is initialized.

Service period overrides cannot be modified immediately after a system time change where the system clock was set back, for example, in case of daylight savings. Overrides cannot be modified until the new local time passes the original pre-adjustment local time.

Usage

The INGSCHED command operates sysplexwide. For an overview see Overview of Commands that Operate Sysplexwide on page 9.

The automation agent as well as the automation manager verify each time slot being modified. Due to the delay between these verifications, a time slot of the current day that is still valid when the agent checks it could already be invalid when the automation manager checks it. For example, the agent passes an override request to the manager at 10:59:59. The automation manager scheduler receives the request at 11:00:01 and rejects it because the start time has already elapsed. This is indicated by return code 8 and reason code X'10C4' in message INGY1004. If this error occurs, you can fix it by correcting the time slot in error and reissuing the override.

Examples

If you enter ingsched a panel similar to Figure 93 is displayed.

<table>
<thead>
<tr>
<th>INGSCH0</th>
<th>SA z/OS - Command Dialogs</th>
<th>Line 1 of 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain ID = IPSNM</td>
<td>---</td>
<td>INGSCHED</td>
</tr>
<tr>
<td>Operator ID = NETOP1</td>
<td>Time = 13:56:24</td>
<td></td>
</tr>
</tbody>
</table>

Cmd: A Show Details B Show Overrides C Show Resources

<table>
<thead>
<tr>
<th>Cmd Schedule Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD_SERV Standard Service Period</td>
</tr>
<tr>
<td>TSOPLUS Special service period for TSOPLUS</td>
</tr>
<tr>
<td>ONLINE Schedule for CICS applications</td>
</tr>
</tbody>
</table>

Command ===>

PF1=Help PF2=End PF3=Return PF6=Roll PF9=Refresh PF12=Retrieve

Figure 93. INGSCHED Initial Command Dialog Panel

- The **Cmd** field allows you to specify command codes. To use one of the command codes shown, type the appropriate letter next to the resource name and press Enter. The following command codes are available:

  A  Show Details. Displays the service periods for the selected schedule without any override.
INGSCHED

B  Show Overrides. Displays the overrides for the selected schedule starting with the current day for the next \( n \) days filling one panel.

C  Show Resources. Displays the list of resources or application groups that make use of the schedule. This can then be used to modify the service periods for the resource.

- The Schedule field shows the schedule (service period) defined in the sysplex.
- The Description field shows the description specified for the resource.

Showing Schedule Details

Overrides are shown in white.

The base time slots for a selected schedule are displayed when specifying command code A in front of the schedule. The panel in [Figure 94] is displayed (note that INGSCHED REQUEST=BASE shows a similar panel).

| Day | Pri | Strt/Stop Strt/Stop Strt/Stop Strt/Stop Strt/Stop |
|-----|-----|------------|------------|------------|------------|------------|
| MON | UP  | 0600-1159  |
|     | DN  | 0000-0559  | 1200-2400  |
| TUE | UP  | 0600-1159  |
|     | DN  | 0000-0559  | 1200-2400  |
| WED | UP  | 0600-1159  |
|     | DN  | 0000-0559  | 1200-2400  |
| THU | UP  | 0600-1159  |
|     | DN  | 0000-0559  | 1200-2400  |
| FRI | UP  | 0600-1159  |
|     | DN  | 0000-0559  | 1200-2400  |
| SAT | UP  | L 0000-2400 |
|     | DN  | H 0000-2400 |
| SUN | UP  | L 0000-2400 |
|     | DN  | H 0000-2400 |

Figure 94. INGSCHED Command Dialog Panel Showing Schedule Details

- The Schedule field shows the name of the schedule as defined in the sysplex.
- The Day field shows the day of the week.
- The Pri field shows the priority associated with the day of the week. It can have the following values:
  - H  High priority. The resource must be down or up.
  - L  Low priority. The resource should be down or up.
- The Strt/Stop field shows the begin and end times of the service window.

Showing Schedule Overrides

To display detailed information for a particular schedule, specify command code B in front of the schedule. The panel displayed, as shown in [Figure 95 on page 233], shows the start and stop times starting from the requested date. The default starts from the current day. For each day, two rows are displayed:
The first row (UP) shows the time slots when the associated resources should be up (available).

The second row (DN) shows the time slots when the associated resources should be down (unavailable).

To modify the start and stop times for the various days, overtype them. Start and stop times that have been overwritten are shown in yellow. The overrides for a particular day can be removed with command code D, except for time slots that are affected by the current time. The time slots will then be restored to the original values.

To delete individual overrides, blank out the appropriate time slot.

**Showing Resources Related to a Schedule**

A list of resources that use the selected schedule is displayed when specifying command code C in front of the schedule. A panel similar to Figure 96 on page 234 is displayed.
The Cmd field allows you to display specific information for each system. The following command code is available:

- **S** Shows overrides. Changes service periods for the resource group.

- The **Resource** field shows the name of the resource that uses the specified schedule.
- The **Type** field shows the type of the resource.
- The **System** field shows the name of the system that the resource resides on.
- The **Description** field shows the description specified for the resource.

The FIND/RFIND subcommands are supported. See "Varying the Format of the Command Output" on page 13 for information.

**Showing Resource Overrides**

A panel similar to Figure 97 on page 235 shows schedule overrides for the selected resource.
The panel shows the service windows currently defined for the resource. The service windows are determined as follows:

- Base service periods from the associated schedule, if any are defined.
- Override service periods from the associated schedule, if any are defined.
- Override service periods associated with the resource.
- The **Day** field shows the day of the week.
- The **Pri** field shows the priority associated with the day of the week. It can have the following values:
  - **H** High priority.
  - **L** Low priority.
- The **Strt/Stop** fields show the beginning and end times of the service window.
  - The first row (UP) shows the time slots when the associated resources must be up (available).
  - The second row (DN) shows the time slots where the associated resources must be down (unavailable).

Time slots that are overrides for the resource are shown in pink. Time slots defined for the base schedule are shown in green, while its overrides are shown in yellow. Time slots that are already passed cannot be deleted or overridden. To set an override, type over the appropriate time field.

---

### INGSSEND

**Purpose**

INGSEND routes commands to other domains using NetView RMTCMD.

You should use the TARGET parameter with system operations commands if possible.
INGSEND

Syntax

```
INGSEND [NETVASIS] [RESP=YES|ACK|NO] [OPER=defined/issuing_task_ID, operator_ID] [TO=current target domain] [CMD=command] [CRM=RMTCMD] [CORRWAIT=defined/10|value]
```

Notes:
1. The default value can be defined using the SA z/OS customization dialog. If it has not been defined, SA z/OS provides the default.
2. This value is set by the command routing process. That is, the last target domain that you sent a command to is remembered and used as the default.

Parameters

**NETVASIS**
Prefix the INGSEND command with NETVASIS if you want to pass the command text in lower or mixed case.

**RESP**
Specifies whether you want to receive an asynchronous response or acknowledgement when you issue the command. Valid options are:

- **ACK**
  If you use this option:
  - A message is displayed on the operator's NCCF console stating whether or not the command executed successfully.
  - If the command executed successfully, message AOF672I will be displayed.
  - If the command executed but failed, message AOF671I will be displayed.
  - If the command could not be delivered to the target domain or RMTCMD autotask, or was delivered and failed to begin execution, the appropriate NetView RMTCMD message will be displayed to explain the failure.

- **NO**
  Specifies that you do not want to receive any output from the command.

- **YES**
  Specifies that you want the output from the command displayed on your NCCF console.

**CORR**
Specifies that you want the output from the command to be displayed by
the NetView WINDOW command. See “Restrictions and Limitations” on correlating command output.

**OPER**

Specifies the operator to use to issue the command.

*operator_ID* can be defined using the customization dialog for each domain. If not defined, it will default to the issuing task ID.

**Notes:**

1. If the operator alias GATOPER or BASEOPER is specified, the operator ID will default to the issuing task ID.
2. If orig.operid = target.operid in the OPER keyword, the command will be issued.

**TO**

Specifies the domain that the command is being issued to.

*[netid].domain*

Specifies the domain that the command is being issued to. It can be defined using the customization dialog. Netid is optional.

**FP**

Specifies that the command is to be routed to the current focal point.

**Notes:**

1. When INGSEND is issued on the current focal point, and TO=FP is specified, the command will be issued.
2. If this parameter (TO=) is omitted, the current target domain (that is, the last domain you sent a command to) is used, if one is set. If none is set, error message AOF102I will be issued.

**CRM**

Specifies that the command routing mechanism to be used is the NetView RMTCMD command. Only CRM=RMTCMD is valid. The parameter is for compatibility only.

**CORRWAIT**

Specifies the CORRWAIT value (in seconds) to be used when INGSEND uses NetView PIPES. The Corrwait PIPE stage is necessary to trap asynchronous command output. A default for this value can be predefined for each target system, using the customization dialog. If not predefined, the default is 10 seconds. For more information on the use of this value, see the discussion of the PIPE command in *NetView Operation*.

**CMD**

Specifies the command to be executed. The length of the command can be up to 280 characters.

**Restrictions and Limitations**

- The INGSEND command will not execute under the primary POI task (PPT).
- INGSEND uses the NetView PIPE command to trap the message output from the routed command. Not all commands are PIPE-enabled, which means that the command may not be treated as requested in the RESP keyword. Some examples of commands that are not PIPE-enabled are:
  - MVS commands on a NetView not exploiting EMCS.
  - Commands that produce fullscreen output.
  - A NetView command that is not PIPE-enabled.
INGSEND

For more information on RMTCMD and PIPE commands, see Tivoli NetView for z/OS User's Guide.

Usage

If the command you are issuing contains either a space, a comma, or a delimiter, you should delimit the command.

You can delimit the command you are sending with either single quotation marks, double quotation marks, or '/'. If you want to include an occurrence of the delimiter character within the delimited string, the character must be preceded by another delimiter character. The delimiter character must appear as the first character of the string after the 'CMD='. INGSEND sends the command defined by the start and end delimiter. Examples of INGSEND with commands that use delimiters are shown in "Examples."

Examples

The following examples show how INGSEND is used and correctly delimited:

1. Consider the following example:
   "INGSEND RESP=ACK,TO=CNM01,CMD=START TASK=DSILOG"

   Command START TASK=DSILOG will be routed with the following:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Value</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>TO</td>
<td>CNM01</td>
<td>Specified</td>
</tr>
<tr>
<td>RESP</td>
<td>ACK</td>
<td>Specified</td>
</tr>
<tr>
<td>OPER</td>
<td>OPER4</td>
<td>Predefined</td>
</tr>
<tr>
<td>CRM</td>
<td>RMTCMD</td>
<td>Default</td>
</tr>
</tbody>
</table>

   The following message will be issued to OPER2's NCCF console:
   "AOF672I COMMAND (START TASK=DSILOG) WAS EXECUTED IN DOMAIN CNM01 BY A RMTCMD COMMAND FROM OPERATOR OPER2"

   OPER4 is defined using the customization dialog. The above example does not use delimiters.

2. The following example does not use delimiters as far as INGSEND is concerned even though there are delimiter characters in it:
   INGSEND RESP=ACK,TO=CNM01,CMD=MVS $D'RMF'"

3. The following is correctly delimited:
   INGSEND RESP=ACK,TO=CNM01,CMD='ACT CHP123'

4. The following example is also correctly delimited and contains the delimiter character within the delimiters. This command is equivalent to the second example:
   INGSEND RESP=ACK,TO=CNM01,CMD='MVS $D'RMF'

5. The following example is also correctly delimited and is an alternative way to type the previous example:
   INGSEND RESP=ACK,TO=CNM01,CMD=/MVS $D'RMF'/

6. The following example will be passed as correctly delimited but only 'MVS $D' will be sent because the end delimiter follows the D thus defining the command:
   INGSEND RESP=ACK,TO=CNM01,CMD='MVS $D'RMF' "

238 System Automation for z/OS: Operator's Commands
If you enter INGSEND without specifying any parameters, a panel similar to Figure 98 will be displayed.

The available fields correspond to the parameters of the INGSEND command (see “Parameters” on page 236 for details).

**INGSESS**

**Purpose**

INGSESS displays OMEGAMON® session definitions, the session status and statistical information about the session.

**Syntax**

```plaintext
/SM590000/SM590000
INGSESS
session_name
REQ=DISPLAY
REQ=DETAIL
START
STOP
/SM590000
/SM590000
TARGET=
system_name
domain_ID
*ALL
OUTDSN=dsname
OUTMODE=LINE
AUTO
NETLOG
/SM590000/SM630000
```

**Parameters**

`session_name`

The name of the OMEGAMON session that an action is requested for. This parameter is only useful in line mode.
More than one name can be specified for the REQ=DISPLAY function. Wildcards are supported. The session name is mandatory for the Start, Stop, and Detail request.

**REQ=action**

The action that you want to perform for `session_name`. It can be one of the following:

**DISPLAY**

Returns the list of sessions that match the specified filter criteria. This is the default.

**DETAIL**

Returns details for the named session.

**START**

Creates a new session using the session attributes for `session_name`.

**STOP**

Destroys the named session.

This parameter is only useful in line mode.

**TARGET**

For information on the TARGET parameter, see "TARGET Parameter" on page 10.

**OUTDSN**

For information on the OUTDSN parameter, see "OUTDSN Parameter" on page 13.

**OUTMODE**

For information on the OUTMODE parameter, see "OUTMODE Parameter" on page 12.

**Examples**

**Figure 99** is displayed when you enter the INGSESS command at the NetView command line.

<table>
<thead>
<tr>
<th>CMD Session</th>
<th>System</th>
<th>Type</th>
<th>Status</th>
<th>Appl-id</th>
<th>User id</th>
<th>SessOper</th>
</tr>
</thead>
<tbody>
<tr>
<td>-----------</td>
<td>------</td>
<td>-------</td>
<td>--------</td>
<td>---------</td>
<td>---------</td>
<td>----------</td>
</tr>
<tr>
<td>CICSKY41</td>
<td>KEYA</td>
<td>OMIICS</td>
<td>ACTIVE</td>
<td>IPSPOCO</td>
<td>SAOM</td>
<td>AOFSES01</td>
</tr>
<tr>
<td>D325G64</td>
<td>KEYA</td>
<td>OMIIDB2</td>
<td>ACTIVE</td>
<td>IPSPODC</td>
<td>SAOM</td>
<td>AOFSES02</td>
</tr>
<tr>
<td>D325G18</td>
<td>KEYA</td>
<td>OMIIDB2</td>
<td>SESSFAIL</td>
<td>IPSPODC</td>
<td>SAOM</td>
<td>AOFSES01</td>
</tr>
<tr>
<td>IMS742CR</td>
<td>KEYA</td>
<td>OMIIMS</td>
<td>ACTIVE</td>
<td>IPSPOI0</td>
<td>SAOM</td>
<td>AOFSES02</td>
</tr>
<tr>
<td>IMS743CR</td>
<td>KEYA</td>
<td>OMIIMS</td>
<td>MAINT</td>
<td>IPSPOI0</td>
<td>SAOM</td>
<td>AOFSES01</td>
</tr>
<tr>
<td>JHSY4MVS</td>
<td>KEYA</td>
<td>OMIIMVS</td>
<td>ACTIVE</td>
<td>IPSPO2RC</td>
<td>SAOM</td>
<td>AOFSES02</td>
</tr>
<tr>
<td>JHSY4MV2</td>
<td>KEYA</td>
<td>OMIIMVS</td>
<td>INACTIVE</td>
<td>IPSPO2RC</td>
<td>SAOM</td>
<td>AOFSES01</td>
</tr>
<tr>
<td>OMSY4MVS</td>
<td>KEYA</td>
<td>OMIIMVS</td>
<td>ACTIVE</td>
<td>IPSPO2RC</td>
<td>SAOM</td>
<td>AOFSES01</td>
</tr>
</tbody>
</table>

Command ==> PF1=Help PF2=End PF3=Return PF6=Roll PF9=Refresh PF12=Retrieve

**Figure 99. INGSESS Command Dialog Panel**
The **Cmd** field allows you to display specific information for each system. The following command code is available:

- **B** Creates a new session with the appropriate OMEGAMON, if it does not already exist, that is, the session status is INACTIVE or MAINT. After successful session creation, the session status is ACTIVE. The session remains active until it is stopped, either explicitly (using command code C), or implicitly by NetView on behalf of OMEGAMON or session task termination.

- **C** Destroys the session with the appropriate OMEGAMON. The session status will be changed to MAINT. The session is disabled so as to prevent it from being created implicitly again.

- **D** Displays details for the selected session, such as the session attributes from the session definition, the current status, and statistical information.

- The **Session** column shows the name of the session that represents an OMEGAMON monitor.
- The **System** column shows the system that established the connection to OMEGAMON.
- The **Type** column shows the type of session, that is, the OS or middleware that the OMEGAMON monitor is monitoring.
- The **Status** column shows the status of the session. It can be ACTIVE, INACTIVE, MAINT, SESSFAIL, or AUTHFAIL.
- The **Appl-id** column shows the name of the OMEGAMON VTAM application as defined by the installation during customization of the OMEGAMON product.
- The **User id** column shows the user that is defined to log on to the OMEGAMON application. The user ID is needed to control access to the OMEGAMON application. It may be blank if product level security is not implemented for this OMEGAMON application.
- The **SessOper** column shows the automated function name that was assigned to the session by SA z/OS during initialization.

If command code D is selected, the detail screen as shown in Figure 100 on page 242 is displayed.
The **Session** field shows the name of the session that represents an OMEGAMON monitor.

The **System** field shows the system that established the connection to OMEGAMON.

The **Description** field shows descriptive information for this session as specified in the automation policy.

The **Status** field shows the status of the session. It can be ACTIVE, INACTIVE, MAINT, SESSFAIL, or AUTHFAIL.

The **Session Operator** field shows the automated function name that was assigned to the session by SA z/OS during initialization.

The **Logical Unit** field shows the name of the source LU that was assigned by NetView upon establishing a Terminal Access Facility (TAF) fullscreen session.

The **Application id** field shows the name of the OMEGAMON VTAM application as defined by the installation during customization of the OMEGAMON product.

The **User id** field shows the user that is defined to log on to the OMEGAMON application. The user ID is needed to control access to the OMEGAMON application. It may be blank if product level security is not implemented for this OMEGAMON application.

The **Password** field shows the password in the form of a string of asterisks or 'SAFPW'. This is the password used to logon to the OMEGAMON application.

The **Timeout** field shows the maximum time to wait for a response from OMEGAMON before the request is terminated.

The **Logon data** field shows the data that is sent to the OMEGAMON session during logon.

The **Users** field shows a list of operators or *AUTO that have interacted with the session since it became ACTIVE.
• The **Statistics** section shows:
  - The **Total # Commands** field shows the number of commands that have been issued on this session since the session became ACTIVE. The counter is reset each time the session becomes ACTIVE.
  - The **Total # exception analysis** field shows the number of exception trap analysis commands that have been issued since the session became ACTIVE. The counter is reset each time the session becomes ACTIVE.
  - The **Total # exceptions tripped** field shows the number of exceptions that actually tripped as reported by the session since it became ACTIVE. The counter is reset each time the session becomes ACTIVE.

**INGSET**

**Purpose**

INGSET is a line-mode command that is primarily used in automation REXX scripts. With the INGSET command you can:
- Remove an operator's request
- Set a status attribute or automation flag for a resource or application group
- Clear or set the hold flag for a resource or application group
- Set the start or stop type for a resource

**Syntax**

```
INGSET [CANCEL] resource REQUEST=request SOURCE=source [COMMAND Parm Flags]

KILL resource REQUEST=request SOURCE=source [COMMAND Parm Flags]

SET resource [Status options Command Parm Flags]

EXPIRED=(date ,time)

TARGET=system_name domain_id sysplex_name

OUTMODE=LINE AUTO NETLOG

CONFIRM: VERIFY=YES NO

Status options:

[OBSERVED=status AUTOSTAT=status HEALTH=status]
```

---

**Chapter 2. System Operations Commands**

243
**INGSET**

**Command_Parm:**

STARTPARM=parm

STOPPARM=parm

STARTTYPE=starttype

STOPTYPE=stoptype

**Flags:**

AUTOFLAG=YES

NO

HOLDFLAG=YES

NO

**Parameters**

**CANCEL**

Removes a request.

resource

Specifies the resource that the request is to be canceled for. The format is name/type/system. Wildcard is supported.

**KILL**

Removes a request without verifying. To perform verification specify the VERIFY parameter.

resource

Specifies the resource that the request is to be killed for. The format is name/type/system. Wildcard is supported.

**SET**

Sets one or more resource attributes. The following settings can be made:

- Observed status
- The health status
- Automation status
- Automation flags
- Hold flag
- Start type for next startup
- Stop type for next shutdown

Wildcard is supported for the SET parameter.

resource

Specifies the resource that an attribute is to be set for. The format is name/type/system. Wildcard is supported.

**REQUEST**

Causes the request to be canceled. You can specify one of the following:

- MAKEAVAILABLE
- MAKEAVAILABLESO
- MAKEUNAVAILABLE
- MAKEUNAVAILABLESC
- MAKEUNAVAILABLESO
INGSET

Wildcard is supported, for example, REQUEST=MAKEUN*

A MakeAvailable vote on a resource reflects an INGREQ resource,REQ=UP,SCOPE=ALL. That is, the vote is inherited from the resource that it was issued against by all of its dependent resources.

A MakeAvailableSO vote on a resource reflects an INGREQ resource,REQ=UP,SCOPE=ONLY. That is, the vote is not inherited from the resource that it was issued against by all of its dependent resources, but only by that resource.

A MakeUnavailable vote on a resource reflects an INGREQ resource,REQ=DOWN,SCOPE=ALL. That is, the vote is inherited from the resource that it was issued against by the resource and all of its dependent resources.

A MakeUnavailableSC vote on a resource reflects an INGREQ resource,REQ=DOWN,SCOPE=CHILDREN. That is, the vote is inherited from the resource that it was issued against by all of its dependent resources, but it does not affect the resource itself.

A MakeUnavailableSO vote on a resource reflects an INGREQ resource,REQ=DOWN,SCOPE=ONLY. That is, the vote is not inherited from the resource that it was issued against by all of its dependent resources, but affects only that resource.

SOURCE
Specifies the source of the request, for example OPERATOR, or AUTOOPS. Wildcard is supported, for example, SOURCE=*.

OBSERVED
Specifies the observed status to be assigned to the resource. The status can be abbreviated, for example, to AV for available.

AUTOSTAT
Specifies the automation status to be assigned to the resource. The status can be abbreviated, for example, to ID for idle.

HEALTH
Specifies the health status to be assigned to the resource. The status can be abbreviated, for example, to NO for normal.

EXPIRED
Specifies the expiry date and time of the request. Any request older than or equal to the specified timestamp is deleted when the other filter criteria are matched. The date format is yyyy-mm-dd and the time format is hh:mm. The time specification is optional.

CAUTION:
Care must be taken when using the EXPIRED parameter for a remote sysplex that runs in a different time zone. The local time of the system where INGSET CANCEL/KILL command is issued determines whether or not the request is expired.

WAIT
Specifies whether or not to wait until the request is complete. The default is YES.

nnn is the number of seconds to wait before giving up and reporting that a timeout has occurred. The maximum time interval is 999 seconds.
VERIFY
Specifies whether canceling the request is to be verified. The following specifications are valid:

YES  Depending on the NetView task type that the INGREQ command is running on, the verification is either displayed on a fullscreen panel or, in line mode, shown on the console. The latter is also done when OUTMODE=LINE is coded.

NO   No verification is performed. VERIFY is set to NO for AUTOTASKs. Otherwise it is set to YES.

WTOR  The list of affected resources is displayed before the request is removed. The operator is prompted to verify the list via WTOR.

The default depends on the type of task that the INGREQ command runs on.

TARGET  For information on the TARGET parameter, see “TARGET Parameter” on page 10.

OUTMODE  For information on the OUTMODE parameter, see “OUTMODE Parameter” on page 12.

STARTPARM  Is the START parameter that is passed to the next START command. The value that is specified here is used only once and takes precedence over the PARM value that is specified on the next INGREQ (REQ=START) command. Use * to reset its value.

STOPTYPE  Specifies the type of command to be issued for the resource's next startup. The value specified here is used only once and takes precedence over the TYPE value specified on the next INGREQ (REQ=START) command.

STOPTYPE  Specifies the type of command to be issued for the resource's next shutdown. The value specified here is used only once and takes precedence over the TYPE value specified on the next INGREQ (REQ=STOP) command. However, a stop type of FORCE, wherever specified, will always be honored.

AUTOFLAG  Specifies the automation flag. It can be either YES or NO. It can be abbreviated.

HOLDFLAG  Specifies whether the resource is to be started. It can be either YES or NO. The HOLD flag can only be set if the resource is down, that is if its observed status is either HardDown, SoftDown, SysGone or Unknown.

Security Considerations
The INGSET command supports resource level security. If turned on, the following profiles in SYSAUTO are checked:
Profile Authority Condition

AGT.sysplex.xcfgrp.RES.resname.restype[.ressys]

UPDATE When function is CANCEL or KILL

CONTROL When function option is SET

If multiple resources are affected, multiple profiles are checked. For SET requests, the request is carried out or rejected as a whole. CANCEL and KILL requests are processed for each profile.

For further details, refer to IBM Tivoli System Automation for z/OS Planning and Installation.

Return Codes

The following return codes can occur:
0 The request was processed successfully.
1 An error occurred while processing the request or exit requested.

Usage

Note that if you cancel the verification process of a list of affected resources, SA z/OS may issue a slightly misleading message. Although the verification has been canceled, message AOF099I FUNCTION COMPLETED may be returned.

Examples

To cancel a previously issued request to make a resource available, enter the following:

INGSET CANCEL TSO/APL/KEY1 REQUEST=MAKEAVAILABLE SOURCE=OPERATOR

To set the start type for the next startup of a JES2 resource to COLD, enter the following:

INGSET SET JES2/APL/KEY1 STARTTYPE=COLD

INGSTR

Purpose

The INGSTR command allows you to display and manipulate all the structures that are defined in your active policy and all allocated structures in the sysplex. You can rebuild or delete a selected structure or start and stop the duplexing of a structure, if applicable.

The command supports full mode and line mode but in line mode only the display function is available. Therefore, you cannot start an action when you issue INGSTR from an NCCF console.
Authorizations

The actions that you can initiate with INGSTR depend on your authorizations. You can see your authorization type on the panel. Note that the authorization types see the current function, and that your authorization type can be different for different functions.

The following authorization types exist:

**DISPLAY**
- You cannot initiate any action that affects the sysplex configuration.

**ALL**
- You can initiate all actions from the corresponding panel.

**Note:** The codes or PF keys that you can initiate one of these actions with are only displayed if you are authorized to perform the action.

Syntax

```
/SM590000/SM590000
  INGSTR
    strname[*]
    CONDITION= No
    ALL= No

/SM590000
  TARGET= system_name
domain_ID
  sysplex_name

/SM590000/SM630000
  OUTMODE=LINE
```

Parameters

**strname[*]**
- Specifies a name pattern that limits the display. The default is to show all structures that match the 'ALL' condition.

**CONDITION**
- Defines whether or not the rebuild condition of each structure is to be determined and displayed.

**ALL**
- Defines whether allocated and unallocated structures or just allocated structures are displayed.

**TARGET**
- For information on the TARGET parameter, see “TARGET Parameter” on page 10.

**OUTMODE**
- For information on the OUTMODE parameter, see “OUTMODE Parameter” on page 12.

Structure Conditions

The condition of an allocated structure can be:

**Rebuild is not supported.**
- The structure can neither be rebuilt, nor can its deletion be forced.
- The structure has at least one active connector that does not support user-managed rebuild, and at least one active connector that does not support system-managed rebuild.
System-managed processes not supported.
The structure cannot be rebuilt, nor can its deletion be forced.

System-managed rebuild, which is a system-managed process, is not possible for the following reason:
- The CFRM couple data sets have not been formatted to support system-managed processes (ITEM NAME(SMREBLD) NUMBER(1) was not specified).

Note: In certain rare cases system-managed processes are not supported although the condition that is displayed on the DRAIN panel seems to indicate the contrary. Then, the rebuild will be initiated, but will fail with message IXC367I indicating that system-managed processes are not supported for the structure.

No alternate coupling facility defined or available.
The structure can neither be rebuilt, nor can its deletion be forced.
The structure has an active connector and supports rebuild but does not have an alternate coupling facility defined in its preference list, or the alternate coupling facilities that are defined in the preference list are currently unavailable.

Insufficient space detected for rebuild.
The structure cannot or could not be rebuilt. Its deletion cannot be forced.
No alternate coupling facility has enough space to rebuild the structure.

Preference list is empty.
The structure cannot be rebuilt because its preference list is currently empty. A possible reason for this is a pending policy change; for pending policy changes, see P column.

Structure is pending deallocation.
XES accepted a forced deletion of the structure but does the real deallocation later.

Note: This status can only occur when MVS APAR OW39404 has not been installed.

Structure is being rebuilt.
The structure is being rebuilt to another coupling facility.

Duplex rebuild is being stopped.
Two instances of the structure were maintained on different coupling facilities. The application is being disconnected from that instance that is allocated on the target coupling facility. After disconnecting, the instance is deleted.

No connection exists.
The structure cannot be rebuilt, but you can force its deletion.
The structure does not have any connections and cannot be rebuilt with system-managed rebuild.

No alternate coupling facility for structure with no connections.
The structure cannot be rebuilt, but you can force its deletion.
The structure does not have any connections. It could be rebuilt with system-managed rebuild, but no alternate coupling facility is defined in its preference list or available.
No alternate coupling facility for structure with no active connections.
The structure cannot be rebuilt, but you can force its deletion.

The structure has only DISCONNECTING, FAILED, or FAILED-PERSISTENT connections. It could be rebuilt with system-managed rebuild, but no alternate coupling facility is defined in its preference list or available.

The structure's initial size is less than its actual size.
The SIZE value is greater than twice the INITSIZE value. The structure can be rebuilt, but a SIZE value that is greater than twice the INITSIZE might cause the following:
- It might be impossible to allocate a structure at a size of INITSIZE, because the amount of control storage that is required to support the SIZE value might actually be larger than INITSIZE.
- If the allocation succeeds, it might result in a structure with a proportionally large amount of its storage allotted to structure controls, leaving too few structure objects to be exploited usefully by the associated application.

For example, if you have requested a maximum size that is very much larger than the initial size, the system will attempt to use a proportionally large amount of the allocated storage for its controls. The result could be that the allocated storage contains control structures for the future maximum size of the structure and insufficient storage might remain for the application's initial use.

No active connection exists.
The structure cannot be rebuilt, but you can force its deletion.

The structure has only DISCONNECTING, FAILED, or FAILED-PERSISTENT connections and cannot be rebuilt with system-managed rebuild.

Note: INGCF STRUCTURE accepts a rebuild request for structures with this condition, but deallocates them.

No connections. System-managed rebuild supported.
The structure can be rebuilt.

The structure does not have any connections, but can be rebuilt with system-managed rebuild.

No active connections. System-managed rebuild supported.
The structure can be rebuilt with system-managed rebuild.

User-managed rebuild is not possible for the structure because it has only DISCONNECTING, FAILED, or FAILED-PERSISTENT connections.

System-managed rebuild is supported
The structure can be rebuilt.

The structure has active connectors. At least one active connector does not support user-managed rebuild, but all active connectors support system-managed rebuild.

Duplex rebuild is active.
The application is connected to two instances of the same structure on different coupling facilities.

[No condition]
When no condition is displayed, the structure can be rebuilt.
The structure has at least one active connection, and all its active connectors support user-managed rebuild.

**Structure is awaiting rebuild.**
The structure has been selected for rebuild but has not been processed yet.

**Structure is currently allocated on **\textit{cf\_name}**.
The structure can be rebuilt on the target coupling facility with the POPULATE action of the ENABLE function. It is currently allocated on the \textit{cf\_name} coupling facility, but the target coupling facility precedes \textit{cf\_name} in the structure's preference list. This condition is displayed only in the ENABLE command dialog.

**Structure allocated in \textit{cf\_name} cannot be rebuilt to this CF.**
The structure can probably not be rebuilt on the target CF with the POPULATE action of the ENABLE function. It is currently allocated in the \textit{cf\_name} CF, but the target CF precedes \textit{cf\_name} in the structure's preference list. And, the actual size of the structure is greater than the free space of the target CF. This condition is displayed only in the ENABLE command dialog.

**Examples**

If you enter \texttt{ingstr} without any parameters, a panel with all coupling facilities of the sysplex is displayed, as shown in [Figure 101].

<table>
<thead>
<tr>
<th>Structure P D Old New Pref. Location(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNA_GBP0 U CFA CFB</td>
</tr>
<tr>
<td>DSNA_GBP1 S CFA CFB</td>
</tr>
<tr>
<td>DSNA_GBP2 S CFA CFB</td>
</tr>
<tr>
<td>DSNA_GB32K S CFA CFB</td>
</tr>
<tr>
<td>DSNA_SCA S CFB CFB</td>
</tr>
<tr>
<td>GRPXC1S_ADMIN CFB</td>
</tr>
</tbody>
</table>

**Command ===＞**
\texttt{D display details / F force / P stop duplex / R rebuild / S start duplex}

**Figure 101. The INGSTR Panel without Structure Conditions.**

The structure display shows all structures regardless of their allocation status or the coupling facility. You can limit the display by specifying a structure name pattern.

- The **P** column indicates whether a policy change is pending for the structure.
  Rebuilding the structure, if possible, will remove the pending status.
- The **D** column shows what type of duplexing is supported:
  - **U** Indicates that user-managed duplexing is supported.
  - **S** System-managed duplexing,
Both, where user-managed is preferred when possible.

Note that, for performance reasons, this status does not include a check of the SMDUPLEX flag in the CFRM couple data set. However, this flag is checked when you use command code S.

- The **Old** column shows where the structure is allocated or where it was first allocated when it is being duplexed.
- The **New** column shows where the duplexed structure is allocated.
- The **Pref. Location(s)** column shows the locations where the structure should preferably be allocated. When the structure is allocated to the preferred coupling facility this column is blank. A '*' in front of the coupling facility name (or names) indicates that the structure does not allow XCF to perform a reallocation.

- The following command codes are available:
  - **D** Display details of the selected structure.
  - **F** Force the deletion of the selected structure.
  - **P** Stop duplexing of the selected structure.
  - **R** Rebuild the selected structure.
  - **S** Start duplexing of the selected structure.

When the **Include unallocated** option is set to YES, all structures that are defined in the policy are shown. When the **Include condition** option is set to YES, the structure’s current condition is also shown. Specifying this option increases the response time required to build the display. A panel similar to that shown in Figure 102 on page 253 is displayed.
Depending on the status of the CFs and the systems in the sysplex you can use the PF10 key to perform the XCF REALLOCATE command to move the allocated structures to their preferred location. When you press PF10 to move the structures, a panel asking you to confirm the action is displayed, as shown in Figure 103 on page 254. Press PF10 to confirm the action, or PF11 to cancel the reallocation.

Note: You must be authorized to perform the FORCE, REBUILD, START, STOP, or REALLOC action.
If you enter, for example, the command INGSTR D* COND=Y ALL=Y OUTMODE=LINE, this produces line command output, as shown in Figure 104.

The XCF REALLOCATE process runs asynchronously on the next system in the sysplex that has access to the CFRM couple data set. XCF processes all structures in sequence. Once started use the refresh PF key for getting the current status of the process. When the process has completed but not all structures have been moved to their preferred location look for IXC544I messages in the netlog. This message provides an explanation of the reason why a structure wasn’t adjusted.

Structures: D*
Sysplex...: KEYAPLEX
Structure P D Old New Pref. Location(s) Condition
----------- -------- -------- ----------------- ----------------------------
DSNA_GB0 U CFA CFB CFA CF B Duplex rebuild is active.
DSNA_GB1 S CFA CFB CFA CF B *Structure is not allocated.
DSNA_GB2 U CFA CFB CFA CF B Duplex rebuild is active.
DSNA_GB3 S CFB CFA CFA CF B *Structure is not allocated.
DSNA_SCA S CFA CFB CFA CF B Duplex rebuild is active.
*** End of Display ***

The INGTHRES command displays all thresholds that are defined for a resource. INGTHRES lists the thresholds that were set using the INGTHRES command or that are defined in your automation control file. INGTHRES also lets you set, change or delete the threshold settings for a particular resource. INGTHRES also sets the defaults for all MVS subcomponents, for all subsystems, or for MVSESA components.

Syntax
Thresholds:

- CRIT=(nn,interval)
- FREQ=(nn,interval)
- INFR=(nn,interval)

Destination:

- system_name
  - domain_ID
  - sysplex_name

Dest_list:

- (Destination)

Parameters

If you do not specify a subsystem, the INGTHRES panel is displayed.

**DEFAULTS**
Sets the default settings for all automated resources that: do not have their own threshold settings OR do not belong to a group that has its own threshold settings.

**SUBSYSTEM**
Sets defaults for all subsystems.

**MVSESAA**
Sets defaults for all MVS subcomponents.

**resname**
Specifies a particular resource, for example, WTOBUF. You can specify any automated resource.

**filter**
Is the filter used for the output. Only resource names that match the filter criteria are displayed. Wildcard is supported. The filter consists of the major resource name and, optionally, of a minor resource name separated by a period, for example, CICS*.TRANS*, or *.CEMT. To view major resources specify *. To view minor resources specify xyz.*.

**REQ**
Specifies the request type which can be:

**SET**
Adds or updates the thresholds for the given resource. This is the default setting.

**DEL**
Deletes the thresholds for the given resource.
INGTHRES

CRIT
Specifies values for the critical threshold.

FREQ
Specifies values for the frequent threshold.

INFR
Specifies values for the infrequent threshold.

`nn` Specifies the number of errors before the threshold is reached. You can specify a maximum of 50.

`interval` Specifies the time period before the threshold is reached in the form `hh:mm[:ss]`, where:
- `hh` Hours
- `mm` Minutes
- `ss` Seconds

TARGET
For information on the TARGET parameter, see “TARGET Parameter” on page 10.

OUTMODE
For information on the OUTMODE parameter, see “OUTMODE Parameter” on page 12.

Restrictions and Limitations

The changes you make are active only until the next time the automation control file is loaded.

The INGTHRES command can only be used when SA z/OS is initialized.

Usage

SA z/OS uses threshold settings to determine when to stop trying to recover a component or restart a subsystem. The primary use of thresholds is to track subsystem abends and to make sure that the abend-and-recovery cycle does not become a loop.

When a threshold is crossed, SA z/OS performs the action specified for that threshold in the automation control file. For components, this action is customer-defined: contact your SA z/OS administrator for more information. For subsystems, SA z/OS performs the following actions:

<table>
<thead>
<tr>
<th>Threshold exceeded</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infrequent</td>
<td>Issues warning message</td>
</tr>
<tr>
<td>Frequent</td>
<td>Issues warning message</td>
</tr>
<tr>
<td>Critical</td>
<td>Stops Restart automation and issues a message to that effect</td>
</tr>
</tbody>
</table>

All three threshold values (CRIT, FREQ, and INFR) can be set with one invocation of the command.

The Critical threshold should specify more errors or a shorter time period than the Frequent and Infrequent thresholds.
Examples

To set defaults for all automated resources, enter:

```
ingthres defaults crit=(2,00:14) freq=(2,01:00) infr=(4,04:00)
```

To add or replace the critical threshold for the resource JES2, enter:

```
ingthres jes2 crit=(2,02:00)
```

If you enter `ingthres jes2`, you will see the command dialog shown in Figure 105.

If JES reaches the Critical threshold, Restart automation does not continue. If JES exceeds the Frequent or Infrequent threshold, a message is sent to notification operators to inform them.

The `Cmd` field lets you modify the thresholds of a resource. The following command codes are supported:

- **A**: Add new thresholds for a resource with the same values as the one selected.
- **C**: Change existing thresholds for a selected resource.
- **D**: Delete threshold settings for a selected resource.

To define thresholds for a resource you can either type the ADD command at the command line or specify command code **A** in front of the resource whose thresholds should be used as the default. You can then modify the settings as needed.

- The **System** field shows the name of the system where the resource resides.
- The **Resource** field shows the name of the resource that the thresholds are defined for. The entries are sorted alphabetically.
- The **Critical** field shows the critical threshold defined for the resource.
- The **Frequent** field shows the frequent threshold defined for the resource.
- The **Infrequent** field shows the infrequent threshold defined for the resource.

---

**Figure 105. Display of Threshold Settings for JES (INGTHRES JES)**

If JES reaches the Critical threshold, Restart automation does not continue. If JES exceeds the Frequent or Infrequent threshold, a message is sent to notification operators to inform them.

- The **System** field shows the name of the system where the resource resides.
- The **Resource** field shows the name of the resource that the thresholds are defined for. The entries are sorted alphabetically.
- The **Critical** field shows the critical threshold defined for the resource.
- The **Frequent** field shows the frequent threshold defined for the resource.
- The **Infrequent** field shows the infrequent threshold defined for the resource.

---

**Chapter 2. System Operations Commands  257**
If you specify command code A, a panel similar to Figure 106 is displayed.

Figure 106. INGTHRES Command Dialog Panel to Add Thresholds

### INGTOPO

#### Purpose

The INGTOPO command collects SA z/OS topology information from the automation manager via the target systems and maintains the corresponding objects and their hierarchies within RODM.

#### Syntax

```
--- INGTOPO ---
    INIT
    REBUILD
    SWEEP
    SWEEPSYS

---
```

#### Parameters

**INIT**

INIT mode causes INGTOPO to build or update the SA z/OS objects in RODM in accordance with the automation managers of all sysplexes that are defined in the INGTOPOF file. INIT must be specified as the first mode of INGTOPO. The autotask that the Topology Manager is run under is also required.

Check the return code of message ING063I in the NetView log to provide information about the result of the INIT request.

You can call INGTOPO with INIT any time. INGTOPO then either rebuilds the SA z/OS objects from scratch or just performs the necessary update operations. In the former case all event-based minor resources that existed in
RODM will be lost, because these minor resources do not exist within the automation manager and therefore cannot be built.

**task_name**  
Specifies the autotask that is used for status collection on the focal point system.

**plex**  
Specifies the sysplex or processor operations focal point.

**REBUILD**  
In REBUILD mode, INGTOPO removes all SA z/OS objects in RODM for the specified sysplex and rebuilds them from scratch. Note that after calling INGTOPO with REBUILD, all event-based minor resources that existed in RODM for this sysplex will be lost because these minor resources do not exist within the automation manager and therefore cannot be built.

Note that the INGTOPO INIT command must have been successfully run at least once before you can use the REBUILD option.

**SWEEP**  
All SA z/OS topology manager objects in RODM are erased and all internal information is reset.

**SweepSys plex**  
Removes all SA z/OS objects in RODM for the specified sysplex or processor operations focal point and stops all monitoring services.

**Restrictions and Limitations**

To run the INGTOPO command the Multisystem Manager (MSM) must be active.

**Usage**

The INGTOPO command should be issued from the NetView that provides status updates to NMC.

---

**INGTRIG**

**Purpose**

INGTRIG shows all triggers that are defined and used by resources within the sysplex.

**Syntax**

```plaintext
INGTRIG trigger
   TARGET=system_name
   domain_id
   sysplex_name

OUTMODE=LINE
   AUTO
   NETLOG

WAIT=nnn
```

---

Chapter 2. System Operations Commands 259
Parameters

*trigger*

Specifies the name of a trigger so that all the resources that use the trigger are displayed.

TARGET

For information on the TARGET parameter, see "TARGET Parameter" on page 10.

OUTMODE

For information on the OUTMODE parameter, see "OUTMODE Parameter" on page 12.

WAIT

Specifies the number of seconds to wait before reporting that a timeout occurred if the automation manager does not provide the requested data. The maximum time interval is 999 seconds.

If omitted, the time interval is 30 seconds.

Restrictions and Limitations

Triggers for previous releases of SA z/OS cannot be shown.

Usage

The INGTRIG command operates sysplexwide. For an overview see "Overview of Commands that Operate Sysplexwide" on page 9.

Examples

If you enter `ingtrig` a panel similar to Figure 107 is displayed.

```
INGKTR0   SA z/OS - Command Dialogs  Line 1 of 10
Domain ID = IPSFP ---------- INGTRIG ---------- Date = 10/26/99
 Operator ID = NETOP1  Sysplex = KEYPLEX  Time = 10:55:16

Cmd:  S show associated resources

Command ===>
PF1=Help PF2=End PF3=Return PF6=Roll
PF9=Refresh PF12=Retrieve

Figure 107. INGTRIG Initial Command Dialog Panel
```

This panel shows all triggers that are defined within the scope of the automation manager.
• The **Cmd** field allows you to display specific information for each system. The following command code is available:
  
  **S**  Show associated resources. Displays all resources that use the trigger.

• The **Trigger** field shows the name of the trigger.

• The **Description** field shows the description of the trigger.

The FIND/RFIND and SORT subcommands are supported. See “Varying the Format of the Command Output” on page 13 for information.

**Showing Resources**

Specify command code **S** in front of a trigger, or `ingtrig triggername` to display all resources that are associated with that selected trigger, as shown in Figure 108.

---

**Figure 108. INGTRIG Command Dialog Panel Showing Resources Associated with a Trigger**

---

• The **Cmd** field allows you to specify command codes to display specific information for the trigger. The following command code is available:
  
  **S**  Show trigger details. Initiates the DISPTRG command dialog for the selected resource.

• The **Resource** field shows the name of the resource.

• The **Type** field shows the type of the resource.

• The **System** field shows the name of the system where the resource resides.

• The **Description** field shows the description specified for the resource.

The FIND/RFIND and SORT subcommands are supported. See “Varying the Format of the Command Output” on page 13 for information.
Purpose

The INGTWS command lets you:

- Display Application, Operation, Special Resource, Work Station and Calendar information from the Current Plan.
- Modify Application, Operation, Special Resource, Work Station information in the Current Plan.
- Issue a request against any controller defined to SA z/OS in a sysplex.
- Issue a request against a foreign controller where the local tracker is defined to SA z/OS.
- The output of the INGTWS command is either full screen or in pipeable line mode.

Note: INGOPC is a synonym for INGTWS.

Syntax

```
> INGTWS [ resource ] [ REQ= LIST | MOD ] [ TYPE= APPL | OP | SR | WS | CAL ]

> TARGET=target [ OUTMODE= LINE | AUTO | NETLOG ]

> MOD Options [ LIST Options ]

> APPL Selection Criteria [ OP Selection Criteria | SR Selection Criteria ]

> WS Selection Criteria

MOD options:

```
> UPDATE=( keyword=value )

LIST options:

```
> TWSARM=( keyword=value )

APPL Selection criteria:

```
> AD=appl-id IA=yyyyMMddhhmm

OP Selection criteria:

```
> AD=appl-id IA=yyyyMMddhhmm OPNO=nnnn JOBNAME=jobname STATUS=status
```
SR Selection options:

-SRNAME=special_resource

WS Selection options:

-WSNAME=workstation

Parameters

resource
The resource specifies the TWS controller that is to be queried or modified. Multiple specifications are allowed as well as system and sysplex application groups. Wildcards % and * are supported.

The command attempts to resolve the specification to a single appropriate target resource. In all cases the groups are resolved to their members and wildcards are resolved to specific sets of resources.

The resulting list of resources is scanned to check whether there is a single active controller. If a single active controller is found then it is used. If no active controller is found then the list is scanned to check whether there is a single active tracker (only active trackers with LUNAME policy entries are checked because only these trackers can be used to communicate with TWS via the PIF interface). If a single active tracker is found then it is used.

If no viable resources are found an error message is displayed. If multiple viable resources are found, and INGTWS is running in full screen mode, a selection panel is displayed; when OUTMODE=LINE is specified an error message is displayed.

If an active controller could be found, a command is dispatched to the appropriate system in the sysplex to execute the TWS API on the same system as the active controller. If a tracker was found, the LUNAME parameter of the trackers OPCNTL entry may be used to specify a remote controller. In this case the command is dispatched to the system where the tracker is running and the APPC API is used to connect to the remote controller from that system.

REQ
Specifies the request to be issued to the TWS subsystem. It can be one of the following:

LIST Lists TWS Current Plan resources.

MOD Modifies TWS Current Plan resources.

TYPE Specifies the type of Current Plan resource to be listed or modified. It can be one of the following:

APPL Specifies the Current Plan Application Description resource.

OP Specifies the Current Plan Operation resource.
SR  Specifies the Current Plan Special resource.
WS  Specifies the Current Plan Workstation resource.
CAL Specifies the Current Plan Calendar resource.

UPDATE
Specifies the fields that are to be updated and the new contents of the field. Multiple fields can be specified separated by a semi-colon ";". The names of the fields are the same names as specified in the MODIFY command arguments in TWS for z/OS Programming Interfaces.

An alternative to specifying all the fields to be updated using the UPDATE= parameter is to specify the fields and their contents in the default SAFE. Specify one field per message with the format of <fieldname><blank>=<blank><contents>. The blanks between the fieldname and the = symbol and the = symbol and the contents are required.

TWSPARM
Specifies additional fields to be used to locate a resource during LIST processing. Multiple fields may be specified separated by a semi-colon ";". The names of the fields are the same names as specified in the LIST command arguments described in the TWS for z/OS Programming Interfaces manual. The field names must match the TWS segment being searched. For example, CPOC fields are valid for TYPE=APPL and CPOP fields are valid for TYPE=OP, and so on. A TWS EQQ* message will be issued if the field specification is incorrect.

An alternative to specifying additional LIST fields using the TWSPARM= parameter is to specify the fields and their values in the default SAFE. Specify one field per message using the format <fieldname><blanks>=<blanks><contents>. The blanks are optional.

If the same field names are specified in both the TWSPARM and the default SAFE, the TWSPARM values will be used.

Additional LIST fields can only be specified with OUTMODE=LINE.

AD  Specifies the Application Description selection criteria. For LIST requests, this may contain the trailing "*" wildcard character. For MOD requests, this must be the exact name of the application description to be updated.

IA  Specifies the input arrival date/time of the application. The format is as specified by the system programmer when installing and customizing TWS. The default format is YYYYMMDDHHMM.

OPNO  Specifies the operation number selection criteria. This is the operation number of an operation in an application description.

JOBNAME  Specifies the TWS jobname. This field is used to qualify requests of type OP and is optional for all requests.

STATUS  Specifies the TWS status. This field is used to qualify requests of type OP and is optional for all requests.

ERRCODE  Specifies the TWS error code. This field is used to qualify requests of type OP and is optional for all requests.
GROUP
Specifies the TWS group. This field is used to qualify requests of type OP and is optional for all requests.

OWNER
Specifies the TWS owner. This field is used to qualify requests of type OP and is optional for all requests.

PRIORITY
Specifies the TWS priority. This field is used to qualify requests of type OP and is optional for all requests.

SRNAME
Specifies the Special Resource selection criteria.
For LIST requests, this may contain the trailing "*" wildcard character. For MOD requests, this must be the exact name of the special resource.
If the special resource name contains special characters then it must be enclosed in single quotation marks.

WSNAME
Specifies the workstation name selection criteria. Specifies the workstation name selection criteria but may also be used to qualify TYPE=OP requests.
For LIST requests, this may contain the trailing "*" wildcard character. For MOD requests, this must be the exact name of the workstation.

TARGET
Specifies the name of the system (system, domain, or sysplex) that the command should be routed to. The TWS controller specified in the resource field must be active on this system or the command will return no data. This is only necessary when the resource is not part of the local sysplex.
For information on the TARGET parameter, see “TARGET Parameter” on page 10.

OUTMODE
For information on the OUTMODE parameter, see “OUTMODE Parameter” on page 12.

Restrictions and Limitations
To use the INGTWS command system operations must be initialized.

Usage
The INGTWS command operates sysplexwide. For an overview see “Overview of Commands that Operate Sysplexwide” on page 9.

Examples
If you type INGTWS a panel similar to Figure 109 on page 266 is displayed.
The **Resource** field shows the name of the TWS active controller subsystem to be used for issuing the requests. The format is *name/type/system*. Wildcards are supported.

The **Target** field shows the name of the system (system name, domain ID or sysplex name) that the command should be routed to. Specifying this is only necessary if the resources do not reside on the local sysplex.

The **Request** field shows the request to be carried out. It can be LIST or MODIFY.

The **Type** field shows the type of TWS Current Plan resource to be specified.

The **Application** field specifies the TWS application ID. This field is used to qualify requests of type APPL or OP and is optional for LIST requests but is required for MODIFY requests.

The **IA®(R) Date/Time** field specifies the TWS input arrival time. This field is used to qualify requests of type APPL or OP and is optional for LIST requests but is required for MODIFY requests.

The **Operation #** field specifies the TWS operation number. This field is used to qualify requests of type APPL or OP and is optional for LIST requests but is required for MODIFY requests.

The **Jobname** field specifies the TWS jobname associated with an operation. This field is used to qualify requests of type OP and is optional for all requests.

The **Status** field specifies the TWS status associated with an operation. This field is used to qualify requests of type OP and is optional for all requests.

The **Error Code** field specifies the TWS error code associated with an operation. This field is used to qualify requests of type OP and is optional for all requests.

The **Group** field specifies the TWS group associated with an operation. This field is used to qualify requests of type OP and is optional for all requests.

The **Owner** field specifies the TWS owner associated with an operation. This field is used to qualify requests of type OP and is optional for all requests.

The **Priority** field specifies the TWS priority associated with an operation. This field is used to qualify requests of type OP and is optional for all requests.
The Workstation field specifies the TWS workstation for the operation. This field is used to qualify requests of type OP and type WS and is optional for all LIST requests but is required for type WS MODIFY requests.

The SR Name field specifies the TWS special resource name. This field is used to qualify requests of type SR and is optional for LIST requests but is required for MODIFY requests.

To set the status of the special resource 'testnnnn' to available, enter the following command:

```
ingtws req=mod type=sr srname=testnnnn outmode=line Update=(resavail=y)
```

If you specify INGTWS * REQ=LIST TYPE=APPL a panel similar to Figure 110 is displayed.

```
  INGYSTO SA z/OS - Command Dialogs Line 1 of 65
  Domain ID = IPSFM -------- INGTWS -------- Date = 04/10/02
  Operator ID = AFRANCK Sysplex = KEY1PLEX Time = 16:19:30
  CMD: A Update B Operations / scroll
  Application Occurrence List
  Input Arrival Error
  CMD Application Id Date Time Status Code Description
  --- ---------------- -------- ----- --------- ----- ------------------------
  - JKOPCTST1 02/01/23 00:01 Completed Test Batch Iface
  - JKTEST1 02/01/23 00:08 Completed This is a test
  - JKOPCTST1 02/01/24 00:01 Completed Test Batch Iface
  - JKTEST1 02/01/24 00:08 Completed This is a test
  - IEFBR14 02/01/24 08:01 Completed This is a test
  - JKOPCTST1 02/01/25 00:01 Starting Test Batch Iface
  - JKTEST1 02/01/25 00:08 Completed This is a test
  - IEFBR14 02/01/25 08:01 Completed This is a test
  - JKOPCTST1 02/01/26 00:01 Error Test Batch Iface
  - JKTEST1 02/01/26 00:08 Completed This is a test
  - IEFBR14 02/01/26 08:01 Completed This is a test
  - JKOPCTST1 02/01/27 00:01 Error Test Batch Iface
  - JKTEST1 02/01/27 00:08 Completed This is a test
  - 
```

Figure 110. INGTWS REQ=LIST TYPE=APPL Command Dialog Panel

If you specify INGTWS * REQ=LIST TYPE=OP a panel similar to Figure 111 on page 268 is displayed.
If you specify INGTWS * REQ=LIST TYPE=SR a panel similar to Figure 112 is displayed.

If you specify INGTWS * REQ=LIST TYPE=WS a panel similar to Figure 113 on page 269 is displayed.
If you specify INGTWS * REQ=LIST TYPE=CAL a panel similar to Figure 114 is displayed.

Press PF10 and PF11 to display more information for each resource type. Issuing the command code A Update command against a resource in the CMD field displays a panel that lets you modify the resource. The Application Description list supports the B Operations command code. Issuing this command code against an application resource displays a list of operations for that resource. SORT/FIND/RFIND commands are supported. See “Varying the Format of the Command Output” on page 13 for further information.
Pressing PF5 displays a filter selection panel similar to Figure 115 is displayed.

Figure 115. INGTWS Filter Command Dialog Panel

Specify filter strings in the format field-name op contents where:
- field-name is a valid field name as specified by the MODIFY command arguments in TWS for z/OS Programming Interfaces.
- op can have the following values:
  
  =
  ^=
  <
  <=
  >
  >=

- contents are the desired values to be matched by the op. The trailing wildcard character ‘*’ may be used for op.

The operands must be separated by a blank.

INGVOTE

Purpose

The INGVOTE command displays:
- All pending requests that were entered from the automation manager.
- All pending votes of a specified resource.

Syntax
Parameters

resource
Specifies the resource to be displayed. The format is name/type/system. Wildcard is supported.

STATUS
Specifies whether to display the winning or losing requests or votes only. The default is ALL. The STATUS parameter can have the following values:

WINNING
Displays the winning requests or votes.

NOWINNING
Displays the losing requests or votes.

LOSING
Displays the losing requests or votes.

NOLOSING
Displays the winning requests or votes only.

ALL
Displays all winning and losing requests or votes. This is the default.

The values can be abbreviated. Note that if you specify a resource name the vote status is examined, otherwise the request status is examined.

SOURCE
This is used as a filter. Only those requests that originate from the named source are shown. Wildcards are supported, for example, OP*. An asterisk (*) matches a string of arbitrary length and a percentage sign (%) matches a single character.

TARGET
For information on the TARGET parameter, see "TARGET Parameter" on page 10.

OUTDSN
For information on the OUTDSN parameter, see "OUTDSN Parameter" on page 13.

OUTMODE
For information on the OUTMODE parameter, see "OUTMODE Parameter" on page 12.
**INGVOTE**

**EXCLUDE**
Specifies the resource type to be excluded from the display. More than one value can be specified. For example, INGVOTE EXCLUDE=SVP causes all service period resources to be filtered out from the display. Specify NO or * to reset the installation-provided default.

**USER**
This is used as a filter. Only those requests that originate from the specified user id are shown. Wildcard is supported. An asterisk (*) matches a string of arbitrary length and a percentage sign (%) matches a single character.

**WAIT**
Specifies the number of seconds to wait before reporting that a timeout occurred if the automation manager does not provide the requested data. The maximum time interval is 999 seconds.
If omitted, the time interval is 30 seconds.

**Restrictions and Limitations**
None.

**Security Considerations**
The INGVOTE command supports resource level security. If turned on, the following profiles in SYSAUTO are checked:

<table>
<thead>
<tr>
<th>Profile</th>
<th>Authority</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGT.sysplex.xcfgrp.RES.resname.restype[.ressys]</td>
<td>UPDATE</td>
<td>When issuing CANCEL or KILL against requests from full screen</td>
</tr>
</tbody>
</table>

Usage other than listed in the table does not cause a resource level security check. Multiple requests entered on the panel are processed individually.

For further details, refer to IBM Tivoli System Automation for z/OS Planning and Installation.

**Usage**
The INGVOTE command operates sysplexwide. For an overview see "Overview of Commands that Operate Sysplexwide" on page 9.

**Examples**
If you enter ingvote a panel similar to Figure 116 on page 273 is displayed.
The **Cmd** field allows you to specify command codes. To use one of the command codes shown, type the appropriate letter next to the resource name and press Enter. The following command codes are available:

- **C** Cancels the request.
- **K** Cancels the request without showing the verification panel.
- **S** Shows details about the request.
- **V** Shows all the votes or requests that have been entered for that resource.

The **Type** field shows the type of the resource.

The **Name** field shows the name of the resource that the request was made against.

The **Request Data** field shows details about the request. See the online help for further information.

The subcommands FIND/RFIND are supported. See “Varying the Format of the Command Output” on page 13 for more information.

**Line-mode Output**

The line-mode output of the INGVOTE command shows additional information about the request:

- Timeout option
- Application parameters
- Request options (command type, Restart option, Override option, Autoremove option)

Figure 117 on page 274 shows an example.
MONITOR

**Purpose**

The MONITOR command causes the SA z/OS monitoring routine to be run.

**Syntax**

```
MONITOR
```

**Parameters**

`subsystem`

Is the name of the subsystem you want monitored. It can be a list of subsystems. The default is that all subsystems will be monitored.

You can also specify wildcards. This applies a filter for monitoring. Only resource names that match the filter criteria are monitored. Wildcard is supported. Valid specifications are, for example, *abc, abc*, or *abc*.

**TARGET**

For information on the TARGET parameter, see "TARGET Parameter" on page 10.

**Restrictions and Limitations**

The MONITOR command can only be used when SA z/OS is initialized.

The output of the MONITOR command is always in line mode.

**Usage**

MONITOR can be used whenever you want to validate your current automation agent status information.
Examples

To monitor TSO, VTAM and all subsystems that start with CICS, enter the following:

monitor TSO VTAM CICS*

OPCAQRY

Purpose

The OPCAQRY command displays the status of TWS Automation operations, including all commands that are received via the request interface. It displays all command requests processed within the defined time limit.

Syntax

```
OPCAQRY subsystem
  REQ=DETAIL
  TARGET system_name
    domain_ID
    sysplex_name
    *ALL

OUTMODE LINE
  AUTO
  NETLOG
```

Parameters

subsystem

The name of the subsystem. Unless you specify REQ=DETAIL, more than one subsystem name as well as a wildcard can be specified. The wildcard can be, for example, SAP*, *SAP or *SAP*

Note: This parameter has no effect for requests with a type of ‘COMMAND’.

REQ=DETAIL

Displays TWS-related information for the specified subsystem. The resource name is mandatory when REQ=DETAIL is specified.

Note: This parameter is not applicable for requests with a type of ‘COMMAND’.

TARGET

For information on the TARGET parameter, see "TARGET Parameter" on page 10.

OUTMODE

For information on the OUTMODE parameter, see "OUTMODE Parameter" on page 12.

Restrictions and Limitations

The OPCAQRY command can only be used when SA z/OS is initialized.
Examples

If you enter the OPCAQRY command without the REQ=DETAIL parameter, a panel similar to Figure 118 is displayed. The panel shows information about TWS-controlled subsystems that match the filter criteria.

<table>
<thead>
<tr>
<th>Subsystem</th>
<th>System</th>
<th>Application</th>
<th>Request</th>
<th>Date</th>
<th>Time</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPCAQ1</td>
<td>KEY4</td>
<td>OPCAQTESTAD</td>
<td>STOP</td>
<td>04/28/06</td>
<td>17:00</td>
<td>Complete</td>
</tr>
<tr>
<td>OPCAQ2</td>
<td>KEY4</td>
<td>OPCAQTESTAD</td>
<td>RECYCLE</td>
<td>04/28/06</td>
<td>17:14</td>
<td>Complete</td>
</tr>
<tr>
<td>RMF</td>
<td>KEY4</td>
<td>OPCAQTESTAD</td>
<td>STOP</td>
<td>04/29/06</td>
<td>18:05</td>
<td>In progress</td>
</tr>
<tr>
<td>KEY4</td>
<td>OPCAQTESTAD</td>
<td>COMMAND</td>
<td>04/29/06</td>
<td>18:07</td>
<td>In progress</td>
<td></td>
</tr>
</tbody>
</table>

Figure 118. OPCAQRY Command Dialog Panel

- The **Cmd** field allows you to specify command codes to invoke another command dialog. The following command codes are available:
  - **D** Shows the TWS operation details for the subsystem.
  - **R** Resets the timer and completion flags to a null value, and unlocks a specific subsystem after a user error has been detected and corrected. By resetting the timer and completion flags, SA z/OS again accepts requests from TWS.

- The **Status** field shows the status of the request or command in SA z/OS.
  
  For a request this is either complete, incomplete, timeout, in progress, or no request.
  
  A status of timeout indicates that the operation is marked in error because it did not complete within the time limit set by the system programmer in the OPCA code entry.
  
  A status of incomplete indicates that the operation did not achieve the expected status set by the system programmer in the OPCA code entry.
  
  Complete and no request are considered normal statuses.
  
  For a command, the statuses are:
  - **In progress**
    - The command has been received
  - **In error**
    - The command completed but failed
  - **Complete**
    - The command completed successfully
Timeout
The command did not finish in time

Waiting
The command finished processing but is now waiting for completion

The FIND/RFIND and SORT subcommands are supported. See "Varying the Format of the Command Output" on page 13 for information.

If you enter command code D for a subsystem or specify the REQ=DETAIL option, a panel similar to Figure 119 is displayed.

![Figure 119. OPCAQRY Command Dialog Panel Showing Details for a Command]

If the command is in an error condition, the details that are displayed are similar to those in Figure 120 on page 278
Purpose

The RESTART command checks a specified subsystem or all subsystems that are defined to SA z/OS on a particular system, and sets the HOLD flag to No.

Syntax

```
RESTART subsystem [TARGET=Dest_list]
```

```
OUTMODE=LINE
```

Destination:

```
- system_name
- domain_ID
- sysplex_name
```

Dest_list:

```
{ Destination }
```
**Parameters**

*subsystem*

Specifies the name of the subsystem to be allowed to start. If you do not specify a subsystem name, SA z/OS attempts to restart all subsystems.

**TARGET**

For information on the TARGET parameter, see “TARGET Parameter” on page 10.

**OUTMODE**

For information on the OUTMODE parameter, see “OUTMODE Parameter” on page 12.

**Restrictions and Limitations**

You can only use the RESTART command when SA z/OS is initialized.

---

**RESTART**

---

**RESYNC**

**Purpose**

This command causes SA z/OS to rerun selected parts of its initial status determination routines or to reinitialize parts of its environment setup.

**Syntax**

```
RESYNC
  SUBSYSTEM REGISTER
  AM WTOR XCF SMF TIMER SMF XCF AM
  TARGET= Destination Dest_list
  OUTMODE= LINE AUTO NETLOG
```

**Destination:**

```
  system_name
  domain_ID
  sysplex_name
```
RESYNC

Dest_list:

![Diagram of Destination]

Parameters

SUBSYSTEM
  Reinitializes the status of all subsystems on the local agent and clears the status file from deleted subsystems.

REGISTER
  Registers all subsystem resources and monitor resources on the local agent and cross system to all agents in the sysplex.

AM
  Status synchronization of all subsystem resources and monitor resources with the automation manager.

WTOR
  Retrieves the list of outstanding WTORs from the system, and updates the stored WTOR data in SA z/OS to refresh their display in SDF and NMC.

XCF
  Retrieves information from MVS and updates the stored data in SA z/OS about:
  - The status of the sysplex timer and the IDs of each active side
  - The coupling facilities that are known to MVS
  - The status of the couple data sets that are in use by the sysplex

SMF
  Checks the SMF data sets and, if any of the SMF data sets are found in the DUMP status, issues the commands that are defined to SMFDUMP in the MESSAGES/USER DATA automation policy for the MVSESA entry.

TIMER
  The timers for the general subsystem monitor and the gateway monitor are reset.

NTFYOP
  Resets the hold settings at all notify operators via SETHOLD AUTO and subscribes the receivers of class 8x and 9x messages to the automation manager.

GATEWAY
  The gateway connections to remote systems that are controlled by SA z/OS are restarted.

SDF
  Reinitializes the status information for the local SDF and forwards it to the SDF focal point.

SDFDEFS
  Generates the SDF panels using the advanced automation option (AAO) AOF_AAO_SDFROOT.n for the SDF root names that are to be applied.

CPMSG
  Forwards the captured exceptional messages to SDF. An exceptional message is a message with a severity of Unusual, Important or Critical.
RESYNC

CPMSG_NMC
Forwards the captured exceptional messages to NMC. An exceptional message is a message with a severity of Unusual, Important or Critical.

FP
Resends the status information for the local SDF to the SDF focal point.

EXITs
Reregisters SA z/OS exit routines if applicable. This includes dropping or loading the SA z/OS USS process exit and the SA z/OS ARM exit. In addition, the existence of user exits AOFEXSTA, AOFEXX02, AOFEXX03 and AOFEXX04 is verified and the exits are enabled or disabled as appropriate.

UTCOFFSET
Reevaluates the time zone offset relating to the Coordinated Universal Time (UTC), formerly known as Greenwich Mean Time (GMT).

TARGET
For information on the TARGET parameter, see "TARGET Parameter" on page 10.

OUTMODE
For information on the OUTMODE parameter, see "OUTMODE Parameter" on page 12.

Restrictions and Limitations
The RESYNC command can only be used when SA z/OS is initialized.

Usage
This command should only be used when your statuses have gone out of synchronization with actual statuses. Possible causes of this are SSI task failure, problems with your NetView automation table, or communication problems with the automation manager.

SETHOLD

Purpose
SETHOLD lets you select which AOF messages are held for your operator ID. This command is used by the INGNTFY command but can be invoked separately.

Syntax

```plaintext
SETHOLD CONFIG
AUTO iaedw
```

Parameters

CONFIG
Specifying this option changes your hold settings to match the settings specified for your operator ID in the automation control file. If your operator ID is not defined as a notification operator, no messages will be held.

AUTO
Specifying this option changes your hold settings to match the settings
SETHOLD

INGNTFY has currently defined for your operator ID. If your operator ID is not a notification operator, or its definition as a notification operator has been deleted, no messages will be held.

**i a e d w**

Each character specifies the hold setting for the corresponding type of message. If a Y is specified, messages of that type will be held, if any other character is specified, messages of that type will not be held. A character must be specified for each type.

**Restrictions and Limitations**

The SETHOLD command is only useful on an OST task. If issued on an NNT or RMT task, the message-holding flags in the remote domain will be ignored, because the automation table on the OST task will override whatever message attributes may have been specified in the remote domain.

SETHOLD does not update the INGNTFY globals, so if you use it with anything other than the AUTO parameter, the hold settings shown by INGNTFY QUERY may not reflect reality.

**Usage**

SA z/OS issues SETHOLD at SA z/OS initialization, when you log on to NetView interface, and when INGNTFY is used.

When you log on, an entry in the sample automation table (in member AOFMSGMS) will invoke SETHOLD AUTO for your operator ID. When INGNTFY is used to change an operator’s HOLD settings, SETHOLD AUTO will be issued on the appropriate OST task after the global information has been updated.

A more efficient usage would be to modify your initial REXX scripts to issue SETHOLD with specific settings and to not have any hold settings specified in the automation control file or through INGNTFY. If you choose to do this, you will need to remove the automation table entry for the OST (in member AOFMSGMS) and set the AOFLOCALHOLD advanced automation option to 1. Doing this will stop INGNTFY from using EXCMD to issue SETHOLD AUTO, which can disrupt the hold settings specified from your profiles.

Note that the attributes for held messages are defined as automation table synonyms in the AOFMSGSY member.

It is recommended that you do not hold information messages because SA z/OS can generate quite a few of them. If you do hold them, use the CLRHELD command to remove them. You may want to modify the attributes for held information messages (held in the AOFMSGSY ACF member) so that held information messages will not actually be held.

**Examples**

To hold the messages that INGNTFY says you should have held, enter SETHOLD AUTO. Messages will be held according to the task INGNTFY setting.

To hold all but information messages, enter SETHOLD N Y Y Y. All AOF messages, except informational ones, will be held.
SETSTATE

Purpose

SETSTATE changes the agent automation status of a given subsystem.

Syntax

```
SETSTATE subsystem
```

**Purpose**

- `AUTODOWN`
- `CTLDOWN`
- `ENDED`
- `MOVED`
- `RESTART`
- `UP`

**Scope**

- `SCOPE=ONLY`
- `SCOPE=ALL`

**Children**

- `CHILDREN`

**Start**

- `START=NO`
- `START=YES`

**Override**

- `OVERRIDE=NO`
- `OVERRIDE=ALL`

**Event**

- `EVT`
- `FLG`
- `TRG`

**Target**

- `TARGET= Destination`
- `Dest_list`

**Outmode**

- `OUTMODE=LINE`
- `AUTO`
- `NETLOG`

**Destination:**

- `system_name`
- `domain_ID`
- `sysplex_name`

**Dest_list:**

- `Destination`

Parameters

- `subsystem`
  
  Specifies the subsystem whose status (or whose dependent resource's status) you want to change. You can also specify the subsystem's job name.

- `AUTODOWN`
  
  Changes the status from CTLDOWN, STOPPED, BROKEN, MOVED or FALLBACK to AUTODOWN. It is primarily used to return the resource to automation control after the operator has fixed the problem.

- `CTLDOWN`
  
  Changes the status to CTLDOWN so that the subsystem cannot be started by automation.
# SETSTATE

**ENDED**
Changes the status to ENDED.

**MOVED**
The effect of this parameter depends on how the subsystem is defined:
- If the subsystem does not have any secondary system associations, the effect of the MOVED operand is the same as the effect of the CTLDOWN operand: the status of the subsystem is changed to CTLDOWN.
- If the subsystem does have secondary system associations:
  - If the command is issued on one of the subsystem's primary systems, the subsystem's status is changed to MOVED
  - If the command is issued on one of the subsystem's secondary systems, the subsystem's status is changed to FALLBACK.

**RESTART**
Changes the status to RESTART so that the subsystem can be restarted. Specify this if your Initstart flag is set to No. This can only be used when:
- SCOPE=ONLY or SCOPE=ALL
- The application monitor status of the subsystem is INACTIVE

**UP**
Changes the status to UP/RUNNING if the subsystem's UPMMSG has been missed. This can only be used when:
- SCOPE=ONLY
- The application monitor status of the subsystem is ACTIVE
- If the subsystem is enabled for automatic restart management, its automatic restart management status must be AVAILABLE or AVAILABLE-TO.

**Note:** For transient subsystems, UP becomes RUNNING.

**SCOPE**
Specifies what is affected by this command:
- **ONLY**
  Changes the status of this resource only. This is the default.
- **ALL**
  Changes the status of this resource and its dependent resources.
- **CHILDREN**
  Changes the status of the dependent resources only.

**OVERRIDE**
Specifies the overrides to be considered for the request. It can be one of the following:
- **NO**
  Specifies that no override will be done. This is the default.
- **ALL**
  Sets the FLG, TRG as well as DPY and STS overrides on. For DPY and STS see the [INGREQ](#) command for more details.
- **EVT | TRG**
  The startup or shutdown trigger conditions will be ignored.
- **FLG**
  The appropriate automation flags will be ignored for the resource.

**START**
Specifies whether the resource is started after the status has been updated.
SETSTATE

NO  No request is sent to the automation manager. The automation agent status is updated. This is the default.

YES  A default start request is sent to the automation manager.

Note: The START parameter is deprecated. SA z/OS 3.5 is the last release that supports it.

TARGET  For information on the TARGET parameter, see “TARGET Parameter” on page 10.

OUTMODE  For information on the OUTMODE parameter, see “OUTMODE Parameter” on page 12.

Restrictions and Limitations

The SETSTATE command will not execute under the primary POI task (PPT).

The SETSTATE command can only be used when SA z/OS is initialized.

When the SETSTATE command is used to change the status of a subsystem to UP or AUTODOWN, then this will trigger a call to ACTIVMSG or TERMMSG. When using AUTHCHK=SOURCEID as part of your SAF product authorization checking profile, the operator/task issuing the SETSTATE command will require the authority to issue commands and replies that may subsequently be triggered by ACTIVMSG or TERMMSG. This is due to the fact that the status change process did not originate on the work operator assigned to the subsystem as usual but on the operator/task who issued the SETSTATE command.

Security Considerations

The SETSTATE command supports resource level security. If turned on, the following profiles in SYSAUTO are checked:

<table>
<thead>
<tr>
<th>Profile</th>
<th>Authority</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGT.sysplex.xcfgrp.RES.name resteype[.ressys]</td>
<td>UPDATE</td>
<td>Always</td>
</tr>
</tbody>
</table>

If START=YES is specified another resource level security check is done by the invoked INGREQ command.

For further details, refer to IBM Tivoli System Automation for z/OS Planning and Installation.

Usage

Use SETSTATE to:

- Resume handling an application after a manual operator shutdown or after any other event that causes an application to get into a non-automatable status (for example STOPPED, CTLDOWN, BROKEN, ACTIVE)
- Prevent automation from restarting an application if you specify CTLDOWN.
- Alert the automation that an application that was shut down will be moved to another system.
SETSTATE

- The SETSTATE command issues an automation manager request with a priority of HIGH (unless overridden by the advanced automation global INGREQ_PRI). Because requests entered by the operator are persistent, it may be necessary to manually remove the associated vote before subsequent requests are activated.

Examples

If you enter `setstate TSO` a panel similar to [Figure 121] is displayed.

![Figure 121. SETSTATE Command Dialog Panel](image)

SETTIMER

Purpose

SETTIMER displays all scheduled timers. You can add, change, delete, suspend, and reactivate timers.

Syntax

```
```

Parameters

TARGET

For information on the TARGET parameter, see "TARGET Parameter" on page 10.

OUTMODE

For information on the OUTMODE parameter, see "OUTMODE Parameter" on page 12.
Restrictions and Limitations

The SETTIMER command can only be used when SA z/OS is initialized.

Usage

For instructions on how to add, change, or delete timers, see “Setting Timers” in IBM Tivoli System Automation for z/OS User’s Guide.

Examples

If you enter settimer a panel similar to Figure 122 is displayed.

![Figure 122. SETTIMER Command Dialog Panel: Display of Scheduled Timers](image)

- The **CMD** field allows you to add, change, or delete a timer. Valid values are:
  
  - **A** Add a new timer with the same values as the one selected.
  
  - **C** Change an existing timer.
  
  - **D** Delete a timer.
  
  - **U** Suspend a timer. The timer is deleted from NetView but all its relevant data is kept in common global variables. Only CHRON timers can be suspended.
  
  - **R** Resume (or reactivate) a suspended timer. Timers that are associated with a subsystem have a new timer ID (ITMRxxxx) assigned to them.
  
- The **TIMER-ID** field shows the name assigned to the timer. It cannot be ALL or begin with a number or the words SYS, RST, AOF, or ING. Valid characters are A–Z, 0–9, #, @, $, , !, _ and ?.

- The **SCHEDULED TIME** field shows the next scheduled day and time that this timer is set to expire.

- The **TYPE** field shows the type of timer. There are three types:

  - **After** Set to occur only after the elapsed time that was specified when the timer was scheduled. For example, if at 12:00 am, you specify 2 hours, the timer will expire at 02:00 pm.

  - **At** Set to occur at the specified date and time. Occurs only once.
**SETTIMER**

**Every**  Set to occur at the time intervals indicated in the Interval field, or at a specific time each day or week.

- The **TASK** field shows the operator where the timer runs. This can be a primary programmed operator interface task (PPT), an automation operator, or a human operator. This task must be logged on when the timer is set and when it expires. It can also specify a group name. Group names must begin with a plus (+) sign. If a group name is specified, the command runs on the first task in the group that is active when the timer expires.

**Note:** Not all commands can run under the PPT.

- The **INTERVAL** field is only valid for Every type timers. It indicates the time intervals that the timer is set to occur at. For example, a value of 00:30:00 means that the timer will issue the command specified every 30 minutes.

- The **CLOCK** field specifies the time format. It can be GMT, which means that a local time change does not affect the command processing, or LOCAL, so that the time is to be adjusted automatically when daylight saving time changes, or when the z/OS time is adjusted by an operator for other reasons.

- The **SAVE** field specifies whether the timer is to be saved to an external file. This allows the timer to be restored after a NetView outage. This applies to timers created using either the customization dialog or the operator interface. Valid values are YES or NO. The default is YES.

- The **CTUP** field specifies whether to activate the timer if it occurs at a time when NetView is down. Valid values are Yes or No. If CTUP is set to YES, the command specified in the timer definition is issued when NetView is restarted, if the timer expired while NetView was down.
  
  The field can only be set to YES if:
  - This is an At type timer (it occurs once at a specific time and date).
  - The timer was created using the customization dialog.
  - The Save field is set to YES.

- The **Cmd** field below the timer ID specifies the command to be issued when the timer expires.

Adding a new timer or changing an existing one leads to the Add/Change Timer panel, similar to Figure 123 on page 289.
To set the timer, fill in the fields (note that if you are changing a timer the **Timer ID** field cannot be altered).

Use the following fields on this panel to define when the different types of timers are to expire:

<table>
<thead>
<tr>
<th>Timer Type</th>
<th>Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AFTER</strong></td>
<td><strong>Interval</strong>: The number of days (0–365) to elapse before the timer is run.</td>
</tr>
<tr>
<td><strong>AT</strong></td>
<td><strong>Day of Week</strong>: Specify the day of the week (Monday through Sunday). Only the first three characters are required (for example, FRI for FRIDAY). <strong>Specific date</strong>: Specify the date (in mm/dd/yy notation) and time (in hh:mm:ss notation). Hours and seconds default to zero if not entered. If the time begins with an X or multiple Xs instead of a number, the command is set to begin at the next increment of time.</td>
</tr>
<tr>
<td><strong>EVERY</strong></td>
<td><strong>Interval</strong>: The number of days (0–365) to elapse before the timer is run. <strong>From/To</strong>: You can also specify the date and time that the interval is to start and to end. If omitted, the timer runs from midnight to midnight. <strong>Day of Week</strong>: Specify the day of the week (Monday through Sunday). Only the first three characters are required (for example, FRI for FRIDAY). You can also specify DAILY for every day of the week, WEEKDAY, or WEEKEND.</td>
</tr>
</tbody>
</table>

Use the **CHRON Parameters** field to specify additional CHRON parameters, such as REFRESH or NOTIFY. You can also specify suboptions of the **EVERY** parameter such as REMOVE, REMAFTER, DAYSMON, CALENDAR, MXREPEAT, OFF, or FOR.

Enter the command that is to be run when the timer expires in the **Timer command** field. The command can span the three input fields, which are protected if the timer command does not fit in the fields. You can use the PF4 key to clear the command fields.
SETTIMER
Part 3. SA z/OS I/O Operations Commands

Chapter 3. I/O Operations Commands ........ 293
Using I/O Operations Commands .......... 293
Consensus Processing .................... 293
Safe Switching .......................... 293
FICON Switches .......................... 294
FICON Cascaded Switches ................. 294
ALLOW .................................. 295
BLOCK .................................. 297
CHAIN .................................. 299
CONNECT ................................ 301
DISCONNECT ............................. 303
DISPLAY CHANGECHECK .................. 304
DISPLAY CHP ............................ 304
DISPLAY DEV ............................ 306
DISPLAY HOST .......................... 308
DISPLAY NAME .......................... 311
DISPLAY PORT .......................... 314
DISPLAY RESULTS ....................... 316
DISPLAY SWITCH ....................... 321
DISPLAY TIMEOUT ....................... 325
DISPLAY VARY .......................... 325
GETLOCK ................................ 328
LOGREC ................................. 329
PROHIBIT ................................ 330
REMOVE CHP .............................. 332
REMOVE SWITCH .......................... 334
RESET CHANGECHECK ..................... 336
RESET CHANGECHECK ..................... 337
RESET SWITCH .......................... 340
RESET TIMEOUT .......................... 341
RESTORE CHP ............................ 341
RESTORE SWITCH ........................ 343
SYNC SWITCH ............................ 344
UNBLOCK ................................ 345
UNCHAIN ................................. 347
UNLOCK ................................ 348
WRITE .................................. 349

This part describes SA z/OS I/O operations (I/O-Ops) commands in detail: their purpose, format, and specifics of how to use them.

For general information about SA z/OS commands, see IBM Tivoli System Automation for z/OS User’s Guide.
Chapter 3. I/O Operations Commands

Using I/O Operations Commands

You can issue I/O operations commands by entering the command through:

- ISPF dialogs
- The operator console

Not all commands can be issued in both ways. For detailed information about where the commands can be used and about the authorization levels they require, see Appendix B, “General Considerations for I/O Operations Commands,” on page 559.

Note that the LSN in all output messages will show “- -” when this value is not valid. This can happen only to cascaded switches that have no channel attachment, which is normally the case. For this reason, it is recommended that all switches are defined to Hardware Configuration Definition (HCD) including their device numbers. This allows I/O Operations to resolve the LSN also for the above-mentioned switches.

Consensus Processing

Consensus processing means a connectivity command is routed to all I/O operations applications that are affected by the command and the results are rated and accumulated until all of them have been received or the timeout value has expired whichever occurs first.

Unless the command is being forced any vote against the command causes the issuing I/O operations application to fail the command. A vote against the command is assumed when:

- a remote I/O operations application has not been excluded from consensus processing (see “Reset Host” command) while the communication to it could not be established or is interrupted.
- the timeout value expires and a response is still missing.
- an I/O operations application responds with a negative vote.

For backout processing see “Making Connectivity Changes” on page 543.

Safe Switching

I/O-Ops varies paths online or offline when, because of port manipulation, the path from a channel to a device either becomes valid or is no longer valid.

The term safe-switching means that all vary path offline requests due to an I/O-Ops connectivity command are backed out if one of these requests fails and BACKOUT was specified at command invocation. All requests means those requests on all systems that have access to the switch (or switches) that are affected by the command.

For FICON® switches, safe-switching also includes the entire vary process for connectivity commands that affect Inter-Switch-Link ports (E-ports). Because I/O-Ops does not know the topology between the entry switch and the destination
A switch of a path, paths that go through an ISL link will not be varied when an E-port is the target of a connectivity command.

The following conditions result in the failure of a request:

- A vary path offline request fails when the request would disable the last path to a device that is currently in use.
- If no VTAM connection could be established between two systems that have access to a switch and run I/O-Ops, I/O-Ops on the local system (that is, where the command is entered) assumes that the command fails on the remote system.

To avoid this, exclude this system from consensus processing using the command RESET HOST vtamname PURGE.

- For other reasons refer to the section “Making Connectivity Changes” in the appendix, “Definitions for I/O Operations Commands” in IBM Tivoli System Automation for z/OS Operator’s Commands, page 543.

**FICON Switches**

FICON switches allow imbedded space characters on port names. Consequently, I/O-Ops will not issue message IHVD106I when detecting imbedded blanks in port names of FICON switches.

However, I/O-Ops does not support imbedded blanks on port names, either in the ISPF dialogs or in the console command interface. The reason is that generic names and port names must not contain imbedded blanks when used in I/O-Ops console commands.

**FICON Cascaded Switches**

I/O-Ops supports cascaded switches with some restrictions:

1. For CTC connections on cascaded switches, I/O-Ops can neither display CTC control unit data nor manage CTC devices. The reason for this is that when I/O-Ops attempts to determine the attached NDs of such a device, it can get stuck behind a never-ending channel program on the device.

2. The Block command is not supported on Inter-Switch-Link ports (E_Ports). When an E_Port is affected by the command, it is rejected with return code 8 and reason code X’49’. In addition, the message IHVC913I is issued, showing the first or only port that is affected by the command.

3. All other I/O-Ops commands affecting E_Ports (Allow, Prohibit, Unblock, and WRITEPORT) must specify the command option IGNore when an E_Port is involved. Otherwise the command is rejected with return code 8 and reason code X’49’. In addition, the message IHVC913I is issued, showing the first or only port that is affected by the command.

The IGNore option makes the issuer of the command aware that safe-switching can no longer be guaranteed.

4. If an attached Node Descriptor of a device cannot be determined because the path or channel is offline, the Display Device command does not show any control unit data for the particular channel path id.

5. A dynamic configuration change that results in the allocation or deallocation of a cascaded switch is currently not supported.
Note: It is recommended that all switches are defined to the Hardware Configuration Definition (HCD) including their device numbers. This allows I/O-Ops to also show the LSN for cascaded switches.

**ALLOW**

**Purpose**

Use the ALLOW command to define a dynamic connection between ports on one switch or on all the switches allocated to the issuing I/O operations.

You can allow the dynamic connections that are shown in the following syntax diagram. With the ALLOW command, you can, for example, define partitions within a system to facilitate maintenance. For information on the reciprocal command, see “PROHIBIT” on page 330.

**Syntax**

```
Allow (portaddress_x)
   portname_x
   generic_name

(portaddress_y)
   portname_y
   generic_name
   (*)
   swchdevn

Vary NOForce BAckout NOCheck NOBackout
```

**Parameters**

(portaddress_x)|portname_x|generic_name

Specifies the target port by its port address (enclosed in parentheses), by its individual port name, or by a generic port name. For a discussion, see “Switching Commands” on page 546.

(portaddress_y)|portname_y|generic_name|(*)|*

Specifies one of the following: the port address of a second port (enclosed in parentheses), the port name of a second port, a generic port name, all other port addresses (* enclosed in parentheses), or all other port names (* not enclosed in parentheses). For a discussion, see “Switching Commands” on page 546.

swchdevn|*

Specifies one switch or all switches (* not enclosed in parentheses) that are allocated to the issuing I/O operations. For a discussion, see “Switch Identifiers” on page 539.
ALLOW

Options

ALSO|ONLY
ALSO (default) specifies that the ports affected by the command will connect dynamically to each other, regardless of whether they connect dynamically to other ports on the same switch.

ONLY specifies that the ports affected by the command will connect dynamically to each other, but each port is disallowed a dynamic connection to any other port on the switch.

Ignore
You must specify this option when an Inter-Switch-Link port (E_Port) is involved. Otherwise the command is rejected with return code 8 and reason code X'49'. The reason is I/O operations can no longer guarantee "safe-switching" when an E_Port is involved.

Connectivity options
For a discussion on Vary|NOVary, NOForce|NOCheck|Force, and BAckout|NOBAckout see "Making Connectivity Changes" on page 543.

For a discussion of why it sometimes isn't obvious that the status of a path or a device has changed after an operator has issued a connectivity command, see "Device and Path Status After Connectivity Commands" on page 562.

Usage
• If you specify an individual or generic port name, all the switches involved must have at least 1 port implemented with an applicable port name.
• If you specify the same port in both operands and do not specify a generic name, the command will fail because I/O operations will not allow that port to have a dynamic connection with itself.
• If you specify 1 or 2 generic names and at least 1 port in each operand is unique and valid, the command will succeed. I/O operations will allow the valid dynamic connections and disallow a port to connect to itself.
• Although you cannot use the ONLY option and specify either all port addresses (*) or all port names *, you can specify a generic port name. Assume, for example, that all the ports on switch 100 are named and the names start with F: FC1, FC2, FC3, and so on. Although you cannot specify A * * 100, you can specify A F* F* 100.
• When you specify the ONLY option, all paths to the specified ports are varied offline. The dynamic path between the two ports is varied online.
• If a parallel channel and CV converter are involved, you should consider the information provided in "A Parallel Channel in Block (BL) Mode" on page 555.

Examples
In the following example, port FB is allowed to have a dynamic connection only with port D6 on switch 100. The dynamic connections that port FB had with other ports are now prohibited.

`A (FB) (D6) 100 ONLY
IHVC0001 ALLOW command successful`
In the following example of using a generic port name, all the ports whose port names start with the characters HB.CHP_39 are allowed to have dynamic connections with ports that begin with CU. on switch 200.

```
ALLOW HB.CHP_39 CU.* 200
IHVC0001 ALLOW command successful
```

**Figure 124. ALLOW Command: Sample 1**

In the following example of using a generic port name, all the ports whose port names start with the characters HB.CHP_39 are allowed to have dynamic connections with ports that begin with CU. on switch 200.

```
ALLOW HB.CHP_39 CU.* 200
IHVC0001 ALLOW command successful
```

**Figure 125. ALLOW Command: Sample 2**

### BLOCK

**Purpose**

Use the BLOCK command to prevent data transfer through the specified port on one or on all the switches allocated to the issuing I/O operations.

See "UNBLOCK" on page 345 for information on the reciprocal command.

**Syntax**

![BLOCK Syntax Diagram]
**Parameters**

\[(portaddress)\|portname\|generic\_name\]

Specifies the ports to be blocked by an individual port address, by an individual port name, or by a generic port name. For a discussion, see "Switching Commands" on page 546.

**swchdevn\|**

Specifies one switch or all switches (* not enclosed in parentheses) allocated to the issuing I/O operations. For a discussion, see "Switch Identifiers" on page 539.

**Options**

**Connectivity options**

For a discussion on Vary|NOVary, NOForce|NOCheck|Force, and BAckout|NOBAckout see "Making Connectivity Changes" on page 543.

For a discussion of why it sometimes isn't obvious that the status of a path or a device has changed after an operator has issued a connectivity command, see "Device and Path Status After Connectivity Commands" on page 562.

**Restrictions and Limitations**

The command is rejected with return code 8 and reason code X'49' when an Inter-Switch-Link port (E_Port) is affected. The reason is that I/O operations does not know the paths and devices to be varied offline.

**Usage**

- If a CVC CHPID is involved, "An ESCON-in-Converted-Mode (CVC) Path" on page 555 provides more information.
- If a parallel CHPID is involved, "A Parallel Channel in Block (BL) Mode" on page 555 provides more information.

**Examples**

```
8 (D6) 100
IHVC0001 BLOCK command successful
```

![Figure 126. BLOCK Command: Sample 1](image_url)

In the following example of using a generic port name, all ports whose port names start with the characters CU.F are blocked on all the switches allocated to the issuing I/O operations.
**CHAIN**

**Purpose**

Use the CHAIN command to define a chain in a chained path that traverses two switches that are physically linked to each other.

One switch acts as a fiber repeater and is called the passthru switch. Two ports in the chain are statically connected to each other in the passthru switch. The other switch is called the destination switch because the chain “ends” in that switch.

For more information, see “Chain and Unchain a Switchable Path (ESCON only)” on page 551. See “UNCHAIN” on page 347 for the reciprocal command.

**Note:** The CHAIN command is not applicable to FICON switches.

**Syntax**

```
CHAIN
(portaddress_e) | portname_e
    destination_swchdevn
(portaddress_m) | portname_m
```

```
(portaddress_a) | portname_a
    passthru_swchdevn
```

```
Vary NOForce BAckout
NOCheck NOBackout
Force NOBackout
NOVary Force NOBackout
```

**Parameters**

`(portaddress_e) | portname_e`

Specifies the port ending the chain on the destination switch by its port address (enclosed in parentheses) or by its port name. For a discussion, see “Switching Commands” on page 546.
CHAIN

destination_swchdevn
Specifies the device number of the destination switch, which is the switch that ends the chain. For a discussion, see "Switch Identifiers" on page 539.

(portaddress_m)|portname_m
Specifies middle port by its port address (enclosed in parentheses) or by its port name. For a discussion, see "Switching Commands" on page 546.

(portaddress_a)|portname_a
Specifies the port to which the CHPID or control unit is physically attached to the beginning of the chained path by its port address (enclosed in parentheses) or by its port name. For a discussion, see "Switching Commands" on page 546.

passthru_swchdevn
Specifies the device number of the passthru switch, which is the switch that has the static connection between ports a and m. For a discussion, see "Switch Identifiers" on page 539.

Options
Connectivity options
For a discussion on Vary|NOVary, NOForce|NOCheck|Force, and BAckout|NOBAckout see "Making Connectivity Changes" on page 543.

Usage
- Additional steps may be required. See "Chain and Unchain a Switchable Path (ESCON only)" on page 551.
- If you need to define a chain, consider entering the Chain command when I/O operations is started so that the program can take the definition into account for all vary path processing.
- Chaining can add complexity to I/O operations' vary path processing. See "Chain and Unchain a Switchable Path (ESCON only)" on page 551 for more information.

Examples

CHAIN (F2) 200 (D4) (C1) 100
IHVC0001 CHAIN command successful
**CHAIN**

**Purpose**

Use the CONNECT command to establish a static, or dedicated, connection between two ports on the same switch or those two ports on all switches allocated to the issuing I/O operations.
CONNECT

See “DISCONNECT” on page 303 for the reciprocal command.

Note: The CONNECT command is not applicable to FICON switches.

Syntax

Connect (portaddress_x) (portname_x) (portaddress_y) (portname_y) swchdevn

Parameters

(portaddress_x) | portname_x
Specifies one target port on the switch by its port address (enclosed in parentheses) or by its port name. For a discussion, see “Switching Commands” on page 546.

(portaddress_y) | portname_y
Specifies the other target port by its port address or by its port name. For a discussion, see “Switching Commands” on page 546.

swchdevn | *
Specifies one switch or all switches (* not enclosed in parentheses) allocated to the issuing I/O operations. For a discussion, see “Switch Identifiers” on page 539.

Options

Connectivity options
For a discussion on Vary | NOVary, NOForce | NOCheck | Force, and BAckout | NOBAckout see “Making Connectivity Changes” on page 543.

Usage

The two ports on the passthru switch of a chain are implicitly connected by the Chain command. Under some circumstances, however, you will need to enter the Connect command in addition to the Chain command. See “Chain and Unchain a Switchable Path (ESCON only)” on page 551 for more information.

Examples

In the following example, ports C0 and D6 become statically connected. In the hierarchy of port attributes, a static connection overrides a dynamic connection. Therefore, port EA cannot communicate with port C0 as long as the latter remains statically connected.

C (FB) (D6) 100
IHVC000I CONNECT command successful
DISCONNECT

Purpose

Use the DISCONNECT command to cancel the definition of a static, or dedicated, connection between two ports on the same switch or those same two ports on all switches allocated to the issuing I/O operations.

See “CONNECT” on page 301 for the reciprocal command.

Note: The DISCONNECT command is not applicable to FICON switches.

Syntax

```
<disconNect> (portaddress_x) (portname_x) (portaddress_y) (portname_y) swchdevn
```

Parameters

- `(portaddress_x)|portname_x` Specifies one target port on the switch by its port address (enclosed in parentheses) or by its port name. For a discussion, see “Switching Commands” on page 546.
- `(portaddress_y)|portname_y` Specifies the other target port by its port address or by its port name.
- `swchdevn|*` Specifies one switch or all switches (* not enclosed in parentheses) allocated to the issuing I/O operations. For a discussion, see “Switch Identifiers” on page 539.

Options

Connectivity options

For a discussion on Vary | NOVary, NOForce | NOCheck | Force, and BAckout | NOBAckout see “Making Connectivity Changes” on page 543.
DISCONNECT

Usage

If a CVC CHPID is involved, “An ESCON-in-Converted-Mode (CVC) Path” on page 555 provides more information.

Examples

N (FB) (D6) 100
IHVC0001 DISCONNECT command successful

DISPLAY CHANGECHECK

Purpose

Use the DISPLAY CHANGECHECK command to display the current setting of the option which indicates whether all opened switches (ON) are checked for a data change prior to a connectivity command or only those (OFF) that are actually affected by a connectivity command. The default setting on the startup of I/O Operations is ON.

Syntax

DISPLAY CHANGECHECK

Examples

0 CC
IHVC861I Option CHANGECHECK is set to ON.

DISPLAY CHP

Purpose

Use the DISPLAY CHP command to display data about one or all of the channel path identifiers (CHPIDs) online to the operating system (host) image that the issuing I/O operations is running on.

Syntax

DISPLAY CHP

Parameters

chpid
  Specifies a single channel path identifier (CHPID).
*  All CHPIDs that are known to the issuing I/O operations.
DISPLAY CHP

Usage

For more information on the data that is displayed, see the references under the fields:

IHVC221I Port
IHVC812I Chp Swch Status
IHVC813I Chp Type Devn LSN Port H S C P Port name
IHVC814I

A B C D E F G H

IHVC815I
IHVC816I Chp Type Swch Status
IHVC817I Devn LSN Port H S C P Port name
IHVC818I

A B C D E F G H

IHVC210I Connected to: .. .. . .. . ........................
IHVC211I Chained to: .... .. .. .. . .. . ........................
IHVC224I connected to: .. .. . .. . ........................
IHVC225I chained to:.... .. .. .. . .. . ........................

A Message identifier

Note: The message ID is not shown in the actual display.

B Channel path ID (see “Channel Types” on page 542)

C Channel type (see “Channel Types” on page 542)

D Switch device number (see “Switch Identifiers” on page 539)

E Logical switch number (see “Logical Switch Number (LSN)” on page 540)

F Port address (see “Port Addresses” on page 540)

G Port status codes (see “Switch Port Hardware Status on a Display Command” on page 541)

H Port name (see “Using Port Names as a Tool in Switching Commands” on page 550)

Examples

IHVC999I I/O-OPS display

<table>
<thead>
<tr>
<th>Chp</th>
<th>Swch</th>
<th>Port</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chp Type</td>
<td>Devn LSN Port</td>
<td>H S C P</td>
<td>Port name</td>
</tr>
<tr>
<td>3C</td>
<td>CNC_S</td>
<td>0200 02 F2 C</td>
<td>HA.CHIP_3C</td>
</tr>
<tr>
<td>Chained to:</td>
<td>0100 01</td>
<td>D4 C</td>
<td>C1</td>
</tr>
<tr>
<td>D4 connected to:</td>
<td>C1 C</td>
<td>D4 HA.CHIP_3C</td>
<td></td>
</tr>
</tbody>
</table>

Chapter 3. I/O Operations Commands 305
DISPLAY CHP

I/O Operations
SYSA

3A

3C

HA.CHP_3C

FB

FE

C1

D4

0100

LINK.200

HA.CHP_3C

F2

0200

C5

CU.820

3C, C1, D4, and F2 information is displayed

Figure 131. DISPLAY CHP Command: Sample 1

DISPLAY CHP

I/O Operations
SYSA

3A

HA.CHP_3A

FB

E0

0100

Figure 132. DISPLAY CHP Command: Sample 2

DISPLAY DEV

Purpose

Use the DISPLAY DEV command to display data about one, a range, or all the devices connected to one or all the switches that are allocated to the issuing I/O operations.

Syntax

```
>> Display Dev devn swchdevn
     lower-upper swchdevn
     lower-* swchdevn
```
DISPLAY DEV

Parameters

devn
A specific device number.

lower-upper
The range of device numbers from the lower limit, followed by a hyphen - and the upper limit.

lower-*
The range of device numbers from the lower limit, followed by a hyphen - and an asterisk (* not enclosed in parentheses) to indicate the highest number.

* All the device numbers (* not enclosed in parentheses).

swchdevn
Specifies one switch allocated to the issuing I/O operations.

* All switches allocated to the issuing I/O operations. For a discussion, see “Switch Identifiers” on page 539.

Restrictions and Limitations

If an attached Node Descriptor (ND) of a device on a cascaded switch cannot be determined because the path or channel is offline, no control unit data is shown for the particular channel path ID.

This is always true for CTC connections on cascaded switches. The reason is that I/O operations can get stuck behind a never-ending channel program on a CTC device when attempting to determine the attached NDs of the device.

Usage

For more information on the displayed output, see the references under the fields.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>IHVC824I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IHVC825I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IHVC826I</td>
<td>Devn</td>
<td>Chp</td>
<td>Swch</td>
<td>Devn LSN Port H S C P Port name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
<td>G</td>
</tr>
<tr>
<td>IHVC827I</td>
<td>......</td>
<td>......</td>
<td>......</td>
<td>......</td>
<td>......</td>
<td>......</td>
</tr>
<tr>
<td>[</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IHVC220I</td>
<td>Cntl unit data:....</td>
<td>......</td>
<td>......</td>
<td>......</td>
<td>......</td>
<td>......</td>
</tr>
<tr>
<td>IHVC828I</td>
<td>Connected to:</td>
<td>......</td>
<td>......</td>
<td>......</td>
<td>......</td>
<td>......</td>
</tr>
<tr>
<td>IHVC829I</td>
<td>Chained to:</td>
<td>......</td>
<td>......</td>
<td>......</td>
<td>......</td>
<td>......</td>
</tr>
<tr>
<td>IHVC830I</td>
<td>.. connected to:</td>
<td>......</td>
<td>......</td>
<td>......</td>
<td>......</td>
<td>......</td>
</tr>
<tr>
<td>IHVC831I</td>
<td>.. chained to:</td>
<td>......</td>
<td>......</td>
<td>......</td>
<td>......</td>
<td>......</td>
</tr>
</tbody>
</table>

A  Message identifier

Note: The message ID is not shown in the actual display.

B Channel path ID (see “Channel Types” on page 542)

C Channel type (see “Channel Types” on page 542)

D Switch device number (see “Switch Identifiers” on page 539)

E Logical switch number (see “Logical Switch Number (LSN)” on page 540)

F Port address (see “Port Addresses” on page 540)

G Port status codes (see “Switch Port Hardware Status on a Display Command” on page 541)
DISPLAY DEV

H Port name (see “Using Port Names as a Tool in Switching Commands” on page 550)

Examples

D D F01 *
IHWC9991 I/O-OPS display

<table>
<thead>
<tr>
<th>Chp</th>
<th>Swch</th>
<th>Status</th>
<th>Devn</th>
<th>Chp Type</th>
<th>Devn LSN</th>
<th>Port</th>
<th>H</th>
<th>S</th>
<th>C</th>
<th>P</th>
<th>Port name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0F01</td>
<td>3A</td>
<td>CNC_S</td>
<td>0100</td>
<td>01</td>
<td>FB</td>
<td>HA.CHP_3A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CU.F00</td>
</tr>
<tr>
<td>0F01</td>
<td>3D</td>
<td>CNC_S</td>
<td>0200</td>
<td>02</td>
<td>F7</td>
<td>HA.CHP_3D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CU.F00</td>
</tr>
</tbody>
</table>

Cntl unit data:0100 01 D6 CU.F00
Cntl unit data:0200 02 D6 CU.F00

The paths to device F01 are 3A, FB, D6, F00, and 3D, F7, D6, F00

Figure 133. DISPLAY DEV Command Sample

DISPLAY HOST

Purpose

Use the DISPLAY HOST command to display status about the specified I/O operations(host).

See “DISPLAY RESULTS” on page 316 for the related command, and “Processing a Vary Path Request” on page 556 for more information.

Syntax

Display Host vtamname tcphostname

swchdevn
DISPLAY HOST

Parameters

vtamname|tcphostname|*

Specifies one or all I/O operations hosts known to the issuing I/O operations.

vtamname

Specifies an I/O operations by its VTAM application name (VTAM ID).
You can specify THIS-SYS in uppercase characters if the issuing I/O operations is the target program and any of the following conditions holds:
• The issuing I/O operations does not have a valid VTAM application name and TCP/IP host name.
• The issuing I/O operations' VTAM name and the TCP/IP host name are not available.
• VTAM and TCP/IP are not available.

tcphostname

Specifies an I/O operations by its TCP/IP host name. You can specify THIS-SYS in uppercase characters if the issuing I/O operations is the target program and any of the following conditions holds:
• The issuing I/O operations does not have a valid VTAM application name and TCP/IP host name.
• The issuing I/O operations' VTAM name and TCP/IP host name are not available.
• VTAM and TCP/IP are not available.

* (asterisk)

Specifies all the I/O operations known to the issuing I/O operations.

For a discussion, see “Switching Commands” on page 546.

swchdevn | *

Specifies one switch allocated to the issuing I/O operations. If you do not specify a switch device number, the issuing I/O operations displays status about all its participating hosts. For a discussion, see “Switch Identifiers” on page 539.

Usage

• See the following example for the type of data that is displayed.
• If the DISPLAY RESULTS, LOGREC, SYNC SWITCH, or WRITE command is entered, the data that could be displayed by DISPLAY HOST is refreshed.
• For more information on the displayed output, see the references under the fields:

<table>
<thead>
<tr>
<th>Message identifier</th>
<th>VTAM name (see “VTAM Application Name” on page 557)</th>
<th>TCP/IP host name (see “TCP/IP Host Name” on page 557)</th>
<th>IPv4 or IPv6 address</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
</tr>
</tbody>
</table>
DISPLAY HOST

IHVC805I The following switches are attached:
IHVC806I Swch
IHVC807I Devn LSN Unique id
IHVC200I .... .. ..........................

A  B  C  D

A  Message identifier
B  Switch device number (see “Switch Identifiers” on page 539)
C  Logical switch number (see “Logical Switch Number (LSN)” on page 540)
D  Switch ID number (see “Switch Unique Identifier” on page 540)

IHVC884I Sysplex system(s) not associated with a host name detected:
IHVC885I Sysplex <- Systems ------------------------------------->
IHVC886I ........ ........, ........, ........, ........, ........,

A  B  C

A  Message identifier
B  Sysplex name
C  System name (or names) in the sysplex that are not associated with a VTAM name or a TCP/IP host name

Examples

IHVC999I I/O-OPS display
IHVC803I This Host: IHVAPPL1/ITHVHOST1 Sysplex/System: IHVPLEX1/IHVSY1
IHVC804I Name Level Physical identifier Sysplex System
IHVC202I IHVAPPL1 V3 R2 002084B16IBM020000000A7260 IHVPLEX1 IHVSY1
IHVC206I IHVHOST1 9.123.47.11
IHVC537I VTAM communication is initialized on this host
IHVC537I TCP/IP communication is initialized on this host
IHVC805I The following switches are attached:
IHVC806I Swch
IHVC807I Devn LSN Unique id
IHVC200I 0100 01 009032005IBMS10000000A7260
IHVC200I 0200 02 009032005IBMS10000000A7465
DISPLAY NAME

Purpose

Use the DISPLAY NAME command to display data about the implemented ports on a switch that port names have been assigned to. (See "WRITE" on page 349 for the command used to assign port names.)

Syntax

```
Display Name
```

Figure 134. DISPLAY HOST Command: Sample 1
DISPLAY NAME

Parameters

\textbf{portname|generic\_name|*}

Specifies that data be displayed about a specifically named port, all ports with the specified generic logical port name, or all named addressable ports (* not enclosed in parentheses). If you specify * in this operand, you cannot specify * in the next operand, but you can specify Block or Connect as options. For a discussion, see "Switching Commands" on page 546.

\textbf{swchdevn|*}

Specifies one switch or all switches (* not enclosed in parentheses) allocated to the issuing I/O operations. If you specify *, you must specify either an \textit{individual} or a \textit{generic port name}, but not all port names. For a discussion, see "Switch Identifiers" on page 539.

Options

\textbf{Block|Connect}

Specifies that data be displayed only about those named ports that are blocked or statically connected to another port. If you specify this option, you must also specify * for all port names.

\textbf{NodeDescriptor}

Specifies that you want to display data of the attached hardware. For all ports except the CUP device port, the node descriptor of the attached hardware is displayed on a separate line when the I/O DEF column shows an unambiguous attachment.

Usage

For more information on the displayed output, see the references under the fields.

\begin{tabular}{|l|l|l|l|}
\hline
IHVC844I & Port Name & Swch & Status \\
IHVC845I & Port Name & Devn & LSN \\
IHVC846I & Port Name & Port & H S C P Def \\
IHVC247I & Port Name & Port & H S C P Def \\
\hline
\end{tabular}

A Message identifier

\textbf{Note:} The message ID is not shown in the actual display.

B Port name (see "Using Port Names as a Tool in Switching Commands" on page 550)

C Switch device number (see "Switch Identifiers" on page 539)

D Logical switch number (see "Logical Switch Number (LSN)" on page 540)

E Port address (see "Port Addresses" on page 540)

F Port status codes (see "Switch Port Hardware Status on a Display Command" on page 541)

G I/O definition (see "Switch Ports" on page 540)

H Node descriptor including tag field
### DISPLAY NAME

#### Examples

**DISPLAY NAME Command: Sample 1. Display all port names**

<table>
<thead>
<tr>
<th>Port Name</th>
<th>Devn</th>
<th>LSN</th>
<th>Port Status</th>
<th>I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>HA.CHP_88.CVC</td>
<td>0100</td>
<td>01</td>
<td>C0</td>
<td>E0</td>
</tr>
<tr>
<td>HA.CHP_3C</td>
<td>0100</td>
<td>01</td>
<td>C1</td>
<td>D4</td>
</tr>
<tr>
<td>LINK.200</td>
<td>0100</td>
<td>01</td>
<td>D4</td>
<td>C1</td>
</tr>
<tr>
<td>CU.F00</td>
<td>0100</td>
<td>01</td>
<td>D6</td>
<td>P</td>
</tr>
<tr>
<td>CU.B20.9034</td>
<td>0100</td>
<td>01</td>
<td>E0</td>
<td>C0</td>
</tr>
<tr>
<td>CU.190</td>
<td>0100</td>
<td>01</td>
<td>EA</td>
<td>P</td>
</tr>
<tr>
<td>HA.CHP_3A</td>
<td>0100</td>
<td>01</td>
<td>FB</td>
<td>P</td>
</tr>
</tbody>
</table>

**DISPLAY NAME Command: Sample 2. Display D6 port names**

<table>
<thead>
<tr>
<th>Port Name</th>
<th>Devn</th>
<th>LSN</th>
<th>Port Status</th>
<th>I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>CU.F00</td>
<td>0100</td>
<td>01</td>
<td>D6</td>
<td>P</td>
</tr>
<tr>
<td>CU.F00</td>
<td>0200</td>
<td>02</td>
<td>D6</td>
<td>P</td>
</tr>
</tbody>
</table>

Figure 135. **DISPLAY NAME Command: Sample 1. Display all port names**

Figure 136. **DISPLAY NAME Command: Sample 2. Display D6 port names**
**DISPLAY PORT**

**Purpose**

Use the DISPLAY PORT command to display data about one or all the ports that are implemented and installed on the specified switch.

**Syntax**

```
Display Port (portaddress) swchdevn
```

**Parameters**

- **(portaddress)**
  Specifies that data should be displayed about one addressable port or all (* enclosed in parentheses) the addressable ports. For a discussion, see "Port Addresses" on page 540.

- **swchdevn**
  Specifies the switch device number that the port is located on. The switch must be allocated to the issuing I/O operations. For a discussion, see "Switch Identifiers" on page 539.

**Options**

- **Block|Connect**
  Specifies that you want data displayed about only those ports that are blocked or statically connected. (If a port is both blocked and statically connected, it would be listed if either alternative is chosen.) If you specify this option, you must specify all addressable ports as an operand (* enclosed in parentheses).

- **NodeDescriptor**
  Specifies that you want to display data of the attached hardware. For all ports

---

![Figure 137. DISPLAY NAME Command: Sample 3. Display all port names and node descriptors](image_url)
except the CUP device port, the node descriptor of the attached hardware is displayed on a separate line when the I/O DEF column shows an unambiguous attachment.

Usage

Unlike the Display Name command, Display Port can apply to only one switch at a time.

For more information on the displayed output, see the references under the fields:

<table>
<thead>
<tr>
<th>IHVC844I</th>
<th>IHVC845I</th>
<th>IHVC846I</th>
<th>IHVC247I</th>
</tr>
</thead>
<tbody>
<tr>
<td>PORT</td>
<td>SWCH</td>
<td>STATUS</td>
<td>I/O DEF</td>
</tr>
<tr>
<td>PORT NAME</td>
<td>DEVN</td>
<td>LSN</td>
<td>PORT H S C P DEF</td>
</tr>
<tr>
<td>A B C D E F G</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[ND ......................... ....]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A Message identifier

Note: The message ID is not shown in the actual display.

B Port name (see “Using Port Names as a Tool in Switching Commands” on page 550)

C Switch device number (see “Switch Identifiers” on page 539)

D Logical switch number (see “Logical Switch Number (LSN)” on page 540)

E Port address (see “Port Addresses” on page 540)

F Port status codes (see “Switch Port Hardware Status on a Display Command” on page 541)

G I/O definition (see “Switch Ports” on page 540)

H Node descriptor of the attached hardware including the tag field

<table>
<thead>
<tr>
<th>IHV849I</th>
<th>IHV240I</th>
<th>IHV241I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allow/Prohibit mask for this port:</td>
<td>A/P Port I/O-Def Port name</td>
<td>A B C D E</td>
</tr>
<tr>
<td>IHV240I</td>
<td>A/P Port I/O-Def Port name</td>
<td>A B C D E</td>
</tr>
<tr>
<td>IHV241I</td>
<td>A/P Port I/O-Def Port name / Node descriptor</td>
<td>A B C D E F</td>
</tr>
<tr>
<td>IHV241I</td>
<td>A/P Port I/O-Def Port name / Node descriptor</td>
<td>A B C D E F</td>
</tr>
</tbody>
</table>

A Message identifier

Note: The message ID is not shown in the actual display.

B Allow or prohibit mask (see “Allow or Prohibit Mask of a Port” on page 541)

C Port address (see “Port Addresses” on page 540)

D I/O definition (see “Switch Ports” on page 540)

E Port name (see “Using Port Names as a Tool in Switching Commands” on page 550)

F Node descriptor of the attached hardware including the tag field
DISPLAY PORT

Examples

```
0 P (D6) 100
IHVC999 I/O-OPS display

Port Name      Swch      LSN Port     H S C P Def
              Devn      Port
0100 01 D6 P  CU
```

Allow/Prohibit mask for this port:

<table>
<thead>
<tr>
<th>A/P</th>
<th>Port</th>
<th>I/O-Def</th>
<th>Port name</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>C0</td>
<td>CH</td>
<td>HA.CHP_88.CVC</td>
</tr>
<tr>
<td>P</td>
<td>C1</td>
<td>CH</td>
<td>HA.CHP_3C</td>
</tr>
<tr>
<td>A</td>
<td>C2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>C3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>C4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>FB</td>
<td>CH</td>
<td>HA.CHP_3A</td>
</tr>
<tr>
<td>A</td>
<td>FE</td>
<td>CU</td>
<td></td>
</tr>
</tbody>
</table>

```

Figure 138. DISPLAY PORT Command: Sample 1

0 P (+) 1011 ND
IHVC999 I/O-OPS display

Port Name      Swch      LSN Port     H S C P Def
              Devn      Port
1011 11 04 ISL
ND 006064.001.MCD.01.000001311AB9 3003
1011 11 05 ISL
ND 006064.001.MCD.01.000001311AB9 3002
1011 11 06 +
1011 11 07 +
1011 11 08 +
1011 11 09 +
1011 11 0A N
1011 11 0B L
1011 11 0C L
1011 11 0D L
1011 11 0E L
1011 11 0F L
1011 11 FE CU
```

Figure 139. DISPLAY PORT Command: Sample 2. Display all ports of a switch including the node descriptors.

DISPLAY RESULTS

Purpose

Use the DISPLAY RESULTS command to display the results of the issuing I/O operations' most recently processed connectivity LOGREC, SYNC SWITCH, or WRITE command (except REMOVE and RESTORE Dev commands). The DISPLAY RESULTS command is a useful tool to help you analyze the resulting configuration status, especially in cases where one or more vary path requests failed.
For related commands, see "DISPLAY HOST" on page 308 and "DISPLAY VARY" on page 325.

Syntax

```
Display Results
```

Usage

- In addition to displaying data about its most recently processed connectivity
  command (except Remove and Restore Dev), the issuing I/O operations displays
  any subsequent command up to the next connectivity, Logrec, Sync Switch, or
  Write command.

- To review any additional failed vary path requests resulting from the same
  command, use the "DISPLAY VARY" on page 325.

- For more information on the displayed output fields, see the references under
  the fields:

```
IHVC803I This Host: vtamappl/tcphost Sysplex/System: plexname/sysname
IHVC808I VTAMname/TCP host Sysplex System
IHVC201I vtamnnl/tcpname1 plexnam1 sysname1
IHVCnnnI One or more status messages
IHVC500I Switch devn LSN Unique Id Status Code
IHVC501I . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
A B C D E F
IHVCnnnI One or more status messages
```

A  Message identifier

**Note:** Message IDs in italics are not shown in display.

B  Switch device number (see "Switch Identifiers" on page 539)

C  Logical switch number (see "Logical Switch Number (LSN)" on page 540)

D  Unique identifier (see "Switch Unique Identifier" on page 540)

E  Switch status

F  Switch status code

*IHVCnnnI* can be one of the following:

**IHVC205I**

- Cannot CHAIN a port to itself

**IHVC301I**

- Port name xxxxxxxxxxxxxxxxxxxxxxx already on this switch

**IHVC404I**

- Current switch data unavailable

**IHVC405I**

- Switch data has been changed

**IHVC409I**

- Port xx does not exist

**IHVC410I**

- Commands cannot affect port xx (switch control unit)

**IHVC411I**

- Unable to return this switch to original state
DISPLAY RESULTS

IHVC413I
I/O-Ops internal error

IHVC414I
Attempted to CONNECT port xx to itself

IHVC415I
Port xx already CONNECTed

IHVC416I
Port xx not installed, cannot CONNECT

IHVC417I
No available paths to the switch for the command

IHVC419I
No port names exist which can be affected by this command

IHVC420I
Attempted to ALLOW port xx to itself

IHVC431I
Switch has been REMOVEd

IHVC433I
Switch has been RESTOREd

IHVC434I
Name xxxxxxxxxxxxxxxxxxxxxxx not valid

IHVC435I
Attempted to ALLOW FC port xx to FC port xx

IHVC436I
Cannot connect xx, because it is a fibre channel port

IHVC437I
Settings on switch prevent configuration changes

IHVC502I
Switch not defined as a device

IHVC506I
No failures detected by this switch

IHVC520I
Command not sent to this switch

IHVC522I
Could not communicate with the switch

IHVC523I
Could not write host names to the switch

IHVC524I
Switch data has been changed during command execution

IHVC533I
Port xx cannot be CONNECTed while in maintenance mode

IHVC622I
ESCON® Converter Model 2 paths may require xxxxxxxxxxxx:

IHVC623I
Id Type Mod Serial number Port Logical name / Unique id
DISPLAY RESULTS

IHVC624I
Chained path associated with: xx xxxxxxxxxxxxxxxxxxxxxxx

IHVC625I
Data follows for chained switch: xxxxxxxxxxxxxxxxxxxxxxx

IHVC626I
CU xxxxx xxx xxxxxxxxxxxxx xx xxxxxxxxxxxxxxxxxxxxxxx

IHVC627I
CV xxxxx xxx xxxxxxxxxxxxx xx xxxxxxxxxxxxxxxxxxxxxxx

IHVC641I
SWCH does not support the requested function

IHVC645I
Switch file in use, no access

IHVC646I
Maximum number of files saved at the switch

IHVC647I
File name xxxxxxx does not exist on switch

IHVC649I
Required resources not available to access file

IHVC651I
Connectivity attributes for port xx conflict with port xx

IHVC653I
Incorrect number of implemented ports for switch

IHVC657I
Cannot delete IPL file at the switch

IHVC659I
Switch initialization not complete

IHVC843I
xxxxxxxxxxxxxxxxxxxxxxxx does not exist on switch xxxx

IHVC903I
Port xx already CHAINed

IHVC906I
Cannot xxxxxxxx CHAINed port xx

IHVC908I
Ports xx and xx are not CONNECTed to each other

IHVC909I
Execution of the CHAIN command will form a closed loop

IHVC910I
Port xx is not part of a CHAINed path

IHVC911I
Ports xx and xx are not part of the same CHAINed path

IHVC912I
Ports not in the correct CHAINed path order

Examples
Example 1

As shown in Figure 140, a chain has been defined so that ports CB and D6 are statically connected on the passthru switch 200.

A user entered N (D6) (CB) 200 and the command failed.

```
I/O Operations
  SYSA
    3A 3C
    FB FE C1
  0100

E0-D6-CB are chained
CB is physically-attached port
D6 is the middle port
E0 is the end-of-chain port
Destination switch Passthru switch
```

Figure 140. DISPLAY RESULTS Command: Scenario 1

The user then entered the DISPLAY RESULTS command to analyze the cause of failure.

```
D R
IHVC999I I/O operations display
IHVC803I This Host: IHVAPPLA/TCPHOSTA Sysplex/System: SYSPLEX1/SYSTEM12
IHVC808I VTAMname/TCP host Sysplex System
IHVC201I IHVAPPLA/TCPHOSTA SYSPLEX1 SYSTEM66
IHVC505I No failures detected by this host
IHVC808I VTAMname/TCP host Sysplex System
IHVC201I IHVAPPLB/ SYSPLEX1 SYSTEM22
IHVC505I No failures detected by this host
Swch devn LSN Unique id Status Code
  0100 01 0090320021BM0100000034893 OPERATIONAL 00
  Command not sent to this SWCH
Swch devn LSN Unique id Status Code
  0200 02 0090320021BM0200000093622 OPERATIONAL 00
  Cannot DISCONNECT CHAINed port D6
  Cannot DISCONNECT CHAINed port CB
```

Example 2

The user tried to Block (C3) 200 NOF.

The command fails because an operational CV converter is attached to port C3.
The user entered the DISPLAY RESULTS command to analyze the cause of failure.

**DISPLAY RESULTS**

```plaintext
DR
IHVC999I I/O operations display
IHVC803I This Host: IHVAPPLA/TCPHOSTA Sysplex/System: SYSPLEX1/SYSTEM12
IHVC808I VTAMname/TCP host Sysplex System
IHVC201I IHVAPPLA/TCPHOSTA SYSPLEX1 SYSTEM66
IHVC505I No failures detected by this host

Swch devn LSN Unique id Status Code
0100 01 009032002IBM01000000034893 OPERATIONAL 00
Command not sent to this SWCH

Swch devn LSN Unique id Status Code
0200 02 009032002IBM02000000093622 OPERATIONAL 00
ESCON Converter Model 2 paths may require DISABLEMENT:
Id Type Mod Serial number Port Logical name / Unique id
CU 003990 G03 0000000090046 C5 CU.820
CV 009035 002 0000000020466 C3 HB.CHP_15.CV.820.3990
```

**DISPLAY SWITCH**

**Purpose**

Use the DISPLAY SWITCH command to display data about one or about all the switches allocated to the issuing I/O operations.

**Syntax**

```
Display-Switch swchdevn Block NodeDescriptor Connect DETail
```

---

Figure 141. DISPLAY RESULTS Command: Scenario 2

Chapter 3. I/O Operations Commands 321
DISPLAY SWITCH

Parameters

*|swchdevn
   Specifies one switch or all switches (* not enclosed in parentheses) allocated to
   the issuing I/O operations. For a discussion, see “Switch Identifiers” on page 539.

Options

DETAIL
   Specifies that you want additional hardware information displayed along with
   the default information. This option is for problem determination purposes.

BLOCK|CONNECT
   Specifies that you want data displayed about only those ports that are blocked
   or are part of a static connection. (If a port is both blocked and statically
   connected, it is listed if either parameter is chosen.) You can enter this optional
   parameter only if you also specified an individual switch device number.

NODE_DESCRIPTOR
   Specifies that you want data of the attached hardware displayed. For all ports
   except the CUP device port the node descriptor of the attached hardware is
   displayed on a separate line when the I/O DEF shows an unambiguous
   attachment.

Usage

For more information on the displayed output, see the references under the fields:

| IHVC834I | Swch LSN Unique id | Ins Status | SC |
| A B C D E F G |

| IHVC83B1 | ......................... ............... |

A Message identifier

**Note:** The message ID is not shown in the actual display.

B Switch device number (see “Switch Identifiers” on page 539)

C Logical switch number (see “Logical Switch Number (LSN)” on page 540)

D Unique identifier (see “Switch Unique Identifier” on page 540)

E Installed ports (see HCD Term)

F Switch status

G Switch status code

| IHVC2731 | Swch LSN Unique id | Tag Status | SC Flags |
| A B C D E F G H |

| IHVC2741 | LIC EC level Sense bytes 0-15 |
| I J K |

| IHVC2751 | Ins Imp 1st LP Sense bytes 16-31 |
| L M N O K |

| IHVC2761 | ......................... ............... |

| IHVC2771 | ........ ........ ........ ........ ........ |

| IHVC2781 | ........ ........ ........ ........ ........ |

A Message identifier

**Note:** The message ID is not shown in the actual display.

B Switch device number (see “Switch Identifiers” on page 539)
DISPLAY SWITCH

C Logical switch number (see "Logical Switch Number (LSN)" on page 540)
D Unique identifier (see "Switch Unique Identifier" on page 540)
E Tag of the node element descriptor (NED)
F Switch status
G Switch status code
H Switch internal flags
I Switch licensed internal code (LIC) version and release
J Engineering change (EC) level
K Sense data
L Installed ports (see HCD Term)
M Implemented ports without CUP
N First implemented port
O Logical paths

A Message identifier

Note: The message ID is not shown in the actual display.

B Port number (see "Port Addresses" on page 540)
C Port status code (see "Switch Port Hardware Status on a Display Command" on page 541)
D Port name (see "Using Port Names as a Tool in Switching Commands" on page 550)
E I/O definition (see "Switch Ports" on page 540)
F Node descriptor of the attached hardware including the tag field

Examples
**DISPLAY SWITCH**

**Example 1**

Display all information about switch 100.

```
D S 100 ND
IHVC999I I/O-OPS display
Swch LSN Unique id Ins Status SC
0100 01 009032002IBM01000000034893 61 OPERATIONAL 00

<table>
<thead>
<tr>
<th>Port Status</th>
<th>Port</th>
<th>H S C P</th>
<th>I/O-Def Port name / Node descriptor</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C0</td>
<td>E0</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td>C1</td>
<td>D4</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td>C2</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C3</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D4</td>
<td>C1</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td>FB</td>
<td>P</td>
<td>CH</td>
</tr>
<tr>
<td></td>
<td>FE</td>
<td>CU</td>
<td></td>
</tr>
</tbody>
</table>
```

![Diagram of switch connections]

*Figure 142. DISPLAY SWITCH Command: Sample 1. Display all information about Switch 100*
Example 2

Display detailed information about all switches.

<table>
<thead>
<tr>
<th>IHV D S * DET</th>
<th>IHVC999I I/O-OPS display</th>
</tr>
</thead>
<tbody>
<tr>
<td>Swch LSN Unique id Tag Status SC Flags</td>
<td></td>
</tr>
<tr>
<td>LIC EC level Sense bytes 0-15 Ins Imp 1st LP Sense bytes 16-31</td>
<td></td>
</tr>
<tr>
<td>1E NOT OPEN 1C</td>
<td></td>
</tr>
<tr>
<td>1001 01 009032005IBM510000000A7260 000 OPERATIONAL 00 20000000</td>
<td></td>
</tr>
<tr>
<td>05.04 n/a 00000000 00000004 05040000 00000000 242 248 4 255 00000000 00000000 00000000 00000000</td>
<td></td>
</tr>
<tr>
<td>1002 02 009032005IBM100000000A7465 000 OPERATIONAL 00 20000000</td>
<td></td>
</tr>
<tr>
<td>05.04 n/a NO SENSE DATA 241 248 4 255</td>
<td></td>
</tr>
<tr>
<td>100A 0A 009032005IBM1000000002013 000 OPERATIONAL 00 20000000</td>
<td></td>
</tr>
<tr>
<td>04.03 n/a 00000000 00000002 04030000 00000000 125 124 128 255 00000000 00000000 00000000 00000000</td>
<td></td>
</tr>
<tr>
<td>1011 11 006140001MCD010000013124DA 000 OPERATIONAL 00 00000000</td>
<td></td>
</tr>
<tr>
<td>09.06 00000000 00000045 09060000 00000000 133 144 4 256 00000000 00000000 00000000 00000000</td>
<td></td>
</tr>
<tr>
<td>101A 1A 000032003IBM510000002483 000 OPERATIONAL 00 20000000</td>
<td></td>
</tr>
<tr>
<td>04.03 n/a 00000000 00000002 04030000 00000000 101 124 128 255 00000000 00000000 00000000 00000000</td>
<td></td>
</tr>
<tr>
<td>101B 1B 01000002MCD0100000131191M 000 OPERATIONAL 00 00000000</td>
<td></td>
</tr>
<tr>
<td>09.07 00000000 00000046 09070000 00000000 32 254 0 256 00000000 00000000 00000000 00000000</td>
<td></td>
</tr>
</tbody>
</table>

DISPLAY TIMEOUT

Purpose

Use the DISPLAY TIMEOUT command to display the maximum number of seconds that the issuing I/O operations will wait to receive votes from the I/O operations that form its voting constituency and that have not been reset off with the command described under “RESET HOST” on page 337. If one or more votes is still outstanding when the time interval lapses, the issuing I/O operations declares a negative consensus.

Syntax

```
Display Timeout
```

Examples

```
D T
IHV860I Current timeout value is 60 seconds
```

DISPLAY VARY

Purpose

Use the DISPLAY VARY command to display a specified number (up to 1500) of failed vary path requests that have resulted either from the issuing I/O operations's most recently processed connectivity command or that were built with the Sync Switch command.

The data displayed depends on the type of command that failed and the options that you specify on the Display Vary command.
DISPLAY VARY

See “DISPLAY HOST” on page 308 and “DISPLAY RESULTS” on page 316 for related commands, and “Processing a Vary Path Request” on page 556 for more information.

Syntax

```
 Display Vary vtamname number_of_varys
```

Parameters

**vtamname**

Specifies the participating I/O operations by its VTAM application ID. The failed vary requests will be listed in the output for that I/O operations (host), but only for a command that was initiated by the issuing I/O operations.

You can specify THIS-SYS in uppercase characters if the issuing I/O operations is the target program and any of the following conditions holds:

- The issuing I/O operations does not have a valid VTAM application name and TCP/IP host name.
- The issuing I/O operations’ VTAM name and the TCP/IP host name are not available.
- VTAM and TCP/IP are not available.

**tcphostname**

Specifies the participating I/O operations by its TCP/IP host name. The failed vary requests will be listed in the output for that I/O operations (host), but only for a command that was initiated by the issuing I/O operations.

You can specify THIS-SYS in uppercase characters if the issuing I/O operations is the target program and any of the following conditions holds:

- The issuing I/O operations does not have a valid VTAM application name and TCP/IP host name.
- The issuing I/O operations’ VTAM name and the TCP/IP host name are not available.
- VTAM and TCP/IP are not available.

**number_of_varys**

```
|  *  |
```

Replace `number_of_varys` with a decimal integer from 1 through 1500. The asterisk (*) specifies all (up to the maximum of 1500). The number of failed vary path requests displayed is independent from similar data displayed by other commands. For example, if you had already displayed three vary path requests using the Display Vary command and want to display 15 additional vary path requests with the Display Vary command, specify 18 vary path requests.

Options

**FAIL**

Specifies only the failed vary path requests (if any), or Varies from a connectivity command be displayed.

**SYNC**

Specifies that the vary path requests built with the Sync Switch command be displayed.
DISPLAY VARY

Usage
- Failed vary path requests listed in this command relate to the path from a
  CHPID to a device.
- If a vary path request fails, the path through the port or ports specified in the
  connectivity command could not be brought offline or online, depending on the
  command that was entered.
- One or more of the following messages are displayed:

  **IHVC212I**
  
  (.....)(.....)(.....)(.....)

  **IHVC213I**
  
  .......... caused VARYs by ........ on ........

  **IHVC214I**
  
  ONLINE .... .... (.....,"..........................”)

  **IHVC215I**
  
  BACKOUT .... .... (.....,"..........................”)

  **IHVC231I**
  
  .......... caused CONFIGs by ........ on ........

  **IHVC232I**
  
  Config type Tried Failed 1st .... failures (Chp)

  **IHVC233I**
  
  OFFLINE .... .... (.....,....,....,....,....)

  **IHVC234I**
  
  BACKOUT .... .... (.....,....,....,....,....)

  **IHVC235I**
  
  ONLINE .... .... (.....,....,....,....,....)

  **IHVC244I**
  
  (.....,....,....,....,....)

  **IHVC271I**
  
  SYNC SWITCH detected VARYs by ........ on ........

  **IHVC272I**
  
  SYNC SWITCH detected CONFIGs by ........ on ........

  **IHVC525I**
  
  This host has been RESET off

  **IHVC811I**
  
  Vary type Tried Failed 1st .... failures (Dev,Chp,Msg)

  **IHVC819I**
  
  OFFLINE .... .... (.....,"..........................”)

  **IHVC822I**
  
  (.....,"..........................”)

  **IHVC853I**
  
  Config type Number 1st .... CONFIGs (Chp)

  **IHVC854I**
  
  OFFLINE .... (.....,....,....,....,....,....)

  **IHVC855I**
  
  ONLINE .... (.....,....,....,....,....)
**DISPLAY VARY**

IHVC856I  
CVC CHPIDs may require ENABLEMENT: ...

IHVC857I  
...

IHVC871I  
No failed VARYs on previous command

IHVC872I  
SYNC type Number 1st .... VARYs (Dev,Chp)

IHVC873I  
OFFLINE .... (.....).........

IHVC874I  
ONLINE .... (.....).........

IHVC875I  
No SYNC VARYs detected on previous command

IHVC877I  
No SYNC CONFIGs detected on previous command

IHVC879I  
No failed CONFIGs on previous command

Note that the message IDs are suppressed, otherwise some messages do not fit onto the text line of a multiline message.

**Examples**

```
D V name *
IHVC999I I/O operations display
Vary type Tried Failed 1st 90 failures (Dev,Chp,Msg)
OFFLINE 5 3 (0190,3A)(019E,3A)(019D,3A)(,)
BACKOUT 2 0 (,,)(,,)(,,)
Config type Tried Failed 1st 90 failures (Chp)
No failed CONFIGs on previous command
SYNC type Number 1st 90 VARYs (Dev,Chp)
No SYNC VARYs detected on previous command
Config type Number 1st 90 CONFIGs (Chp)
No SYNC Configs detected on previous command
```

**GETLOCK**

**Purpose**

Use the GETLOCK command to obtain the processing lock for a user ID that enters a command to any instance of I/O operations within the same set of I/O operations. (A set of I/O operations incorporates all the base programs that can participate in consensus processing of a connectivity command initiated by the issuing instance of I/O operations and those that can communicate with one of these participants via VTAM.)

See "UNLOCK" on page 348 for the reciprocal command.

**Syntax**

```
GETLOCK
```
GETLOCK

Usage
- The GETLOCK command is used when a user wants to perform a series of I/O operation commands without interruption.
- You may need to enter the GETLOCK command if another user ID has the lock and will not, or cannot, release it. Such a situation could arise, for example, if the other user’s terminal is inactive.
- Note that default conditions for obtaining the processing lock are provided with the I/O operations the sample ISPF dialog. You can accept these defaults or customize them for your ISPF user ID.
- The GETLOCK command takes control of I/O operations processing and can, therefore, potentially interrupt a series of connectivity commands being entered by another user ID with unpredictable results.
- If your user ID has control authorization required to enter the Getlock command, be sure you have considered all pertinent aspects of the system I/O environment before issuing this command.
- Note that you can release the lock with the Unlock command and should, of course, do so when you are finished entering I/O operations commands for the session.

Examples

```
GETLOCK
IHVC0001 GETLOCK command successful
```

LOGREC

Purpose
Use the LOGREC command to have I/O operations instruct the specified switch or switches to send their log data to the host (system control program) that the issuing I/O operations is running on.

Syntax

```
LOGREC swchdevn
```

Parameters

```
swchdevn | *
```

Specifies one switch or all the switches (* not enclosed in parentheses) allocated to the issuing I/O operations. For a discussion, see "Switch Identifiers" on page 539.

Usage
- The LOGREC command remains in effect across I/O operations sessions, unless the host image becomes dysfunctional.
- If you enter the LOGREC command, it is advisable to do so when the I/O operations base program is started.
- The LOGREC command causes any data that I/O operations has accumulated for display with the DISPLAY RESULTS command to be refreshed.
PROHIBIT

Purpose

Use the PROHIBIT command to prevent data transfer between ports on the same switch or on all switches allocated to the issuing I/O operations.

Unlike the BLOCK command, the PROHIBIT command allows the two ports to have dynamic connections with other ports. You can prohibit data transfer between ports according to the following syntax diagram. See “ALLOW” on page 295 for the reciprocal command.

Syntax

```
Prohibit (portaddress_x) portname_x generic_name (portaddress_y) portname_y generic_name (* *)
```

Parameters

```
(portaddress_x)|portname_x|generic_name
```

Specifies the port to be prohibited by its port address (enclosed in parentheses), by its individual port name, or by a generic port name. For a discussion, see “Switching Commands” on page 546.

```
(portaddress_y)|portname_y|generic_name|(*)|*
```

Specifies the other ports in one of the following ways: a port address (enclosed in parentheses), an individual port name, a generic port name, all other port addresses (* enclosed in parentheses), or all other port names (* not enclosed in parentheses). For a discussion, see “Switching Commands” on page 546.

```
swchdevn|*
```

Specifies one switch or all the switches (* not enclosed in parentheses) allocated to the issuing I/O operations. For a discussion, see “Switch Identifiers” on page 539.

Options

IGNore

You must specify this option when an Inter-Switch-Link port (E_Port) is involved. Otherwise the command is rejected with return code 8 and reason code X’49’. The reason is I/O operations can no longer guarantee “safe-switching” when an E_Port is involved.
Connectivity options

For a discussion on Vary | NOVary, NOForce | NOCheck | Force, and
BAckout | NOBAckout see “Making Connectivity Changes” on page 543.

For a discussion of why it sometimes isn’t obvious that the status of a path or
a device has changed after an operator has issued a connectivity command, see
“Device and Path Status After Connectivity Commands” on page 562.

Usage

• If you specify an individual or generic port name, all the switches involved must
  have at least one port implemented with an applicable port name.
• If a parallel channel path and CV converter are involved, consider the
  information in “A Parallel Channel in Block (BL) Mode” on page 555.

Examples

```
P (FB) (D6) 100
IHVC0001 PROHIBIT command successful
```

```
P HB.CHP_39 CU.* 200
IHVC0001 PROHIBIT command successful
```

Figure 143. PROHIBIT Command: Sample 1

```
P HB.CHP_39 CU.* 200
IHVC0001 PROHIBIT command successful
```

Figure 144. PROHIBIT Command: Sample 2
REMOVE CHP

**Purpose**

Use the REMOVE CHP command to configure the specified CHPID offline to the host image of the target I/O operations in the command. See "RESTORE CHP" on page 341 for information on the reciprocal command.

**Syntax**

```
>> REMOVE Chp chpid vtamname tcphostname
     Vary NOForce BAckout
     NOCheck NOBackout

CONDitional
UNCONDitional
```

**Parameters**

*chpid*

Specify the CHPID as a hexadecimal number in the range X'00' through X'FF'.

*vtamname*

Specify the VTAM application name of the target I/O operations.

The target I/O operations is the program that instructs its host image to configure the specified CHPID offline.

The target system can be any I/O operations host that a VTAM session is established with.

You can specify THIS-SYS in uppercase characters if the issuing I/O operations is the target program and any of the following conditions holds:

- The issuing I/O operations does not have a valid VTAM name and TCP/IP host name.
- The issuing I/O operations' VTAM name and TCP/IP host name are not available.
- VTAM and TCP/IP are not available.

To review a list of applicable VTAM names, you can use the Display Host command. If you enter DH*, you can determine whether the issuing I/O operations calls itself THIS-SYS.

*tcphostname*

Specify the TCP/IP host name of the target I/O operations.

The target I/O operations is the program that instructs its host image to configure the specified CHPID offline.

The target system can be any I/O operations host that a TCP/IP connection is established with.

You can specify THIS-SYS in uppercase characters if the issuing I/O operations is the target program and any of the following conditions holds:

- The issuing I/O operations does not have a valid VTAM name and TCP/IP host name.
- The issuing I/O operations' VTAM name and TCP/IP host name are not available.
Options

**VARY**

- **VARY** is the default and NOVary is *not* an alternative.

  I/O operations issues the appropriate vary path-offline requests for all the paths that have been defined to the specified CHPID in the relevant host I/O Control Data Set (IOCDS).

  Only after all these paths have been varied offline does the target I/O operations request the target host image to configure the specified CHPID offline.

  For a discussion of why it sometimes isn’t obvious that the status of a path or a device has changed after an operator has issued a connectivity command, see “Device and Path Status After Connectivity Commands” on page 562.

**NOForce|NOCheck**

- NOForce is the default. Although you can specify NOCheck, that alternative has no additional effect in this command.

**BAckout|NOBackout**

- If you specify BBackout, I/O operations attempts to vary those paths back online that had been varied offline successfully if either:
  - At least one vary path-offline request failed.
  - All the vary path requests were successful, but the program could not configure the CHPID offline.

**CONDitional**

- This is the default option for both the REMOVE CHP and RESTORE CHP commands. It indicates that no special configure offline or configure online action should be performed.

**UNCONDITIONal**

- For the REMOVE CHP command, this option puts the specified chpids immediately into pending offline status, even if the chpids are currently active, allocated, or reserved.

  For the RESTORE CHP command, this option brings the specified chpids online, even if there are no paths to the chpids, or if the chpids are pending offline and boxed.

Usage

- The CHPID type must be known to the target I/O operations.

- If the REMOVE CHP command causes one or more devices to be removed from the host image that the target I/O operations is running on, I/O operations automatically attempts to delete these devices from its internal configuration map.

- I/O operations does not attempt to configure the channel offline if any of the relevant vary path requests failed.

- I/O operations does not attempt to configure the channel offline if the target I/O operations is reset off or purged from intersystem communication.

- If the target I/O operations is running in an MVS environment and if the program timed out with the state of the CHPID unknown, I/O operations attempts to configure the channel back online.
If you are considering specifying a CVC channel for this command, note that SA z/OS automatically configures a CVC CHPID offline when it processes any of the following commands for a path defined with that CHPID: BLOCK, DISCONNECT, REMOVE SWITCH, and UNCHAIN.

The command is not supported if the target I/O operations is running on a host system that is operating as a guest in a VM environment.

**REMOVE SWITCH**

**Purpose**

Use the REMOVE SWITCH command to vary all logical paths through the specified switch offline and, optionally, to disable the unit.

The switch must be allocated to the issuing I/O operations. See "RESTORE SWITCH" on page 343 for the reciprocal command, and "Remove (Quiesce) a Switch" on page 549 for discussion.

**Syntax**

```
Vary NOForce BAckout
NOCheck NOBackout
Force NOBackout
NOVary Force NOBackout
NODisable
Disable
```

**Parameters**

*swchdevn*

Specifies the switch to be varied offline and optionally to be disabled. The switch must be allocated to the issuing I/O operations. For a discussion, see "Switch Identifiers" on page 539.

**Options**

* NODisable|Disable

NODisable (default) specifies that every path that passes through the switch is varied offline. (The path to the control unit port, or CUP, is not varied offline.) The switch is not deallocated, or detached, from I/O operations.

Disable specifies that every path that passes through the switch is varied offline. (The path to the control unit port, or CUP, is not varied offline.) The switch is then deallocated, or detached, from I/O operations.

**Connectivity options**

For a discussion on Vary|NOVary, NOForce|NOCheck|Force, and BAckout|NOBAckout see "Making Connectivity Changes" on page 543.

For a discussion of why it sometimes is not obvious that the status of a path or a device has changed after an operator has issued a connectivity command, see "Device and Path Status After Connectivity Commands" on page 562.
**Usage**

- If a path through the switch is chained and you intend to remove the switch, physically cancel the definition of the chain first.
- If you specify the Disable option, power to the switch must be turned off and then on before communications can be restored to the unit. See "RESTORE SWITCH" on page 343.

**Attention:** Issuing the REMOVE Switch command with the Disable option will switch power off to the IBM 9032 Model 2 ESCON Director but will just cycle the power (power stays switched on) for the IBM 9032 Model 3 ESCON Director.

- If you specify the Disable option, and if you plan to restore what is physically the same switch using the RESTORE S command later, do not change the switch device offline to MVS or z/OS. However, if you are planning to restore a physically different switch, for example, when replacing a 9032-003 ESCON Director with a 9032-005 ESCON Director, the correct sequence of operations is as follows:
  1. Issue REMOVE S.
  2. Change the switch device offline to the operation system.
  3. Perform the physical replacement of the switch.
  4. Change the switch online to the operating system.
  5. Then issue RESTORE S.

SA z/OS’s REMOVE DEVICE and RESTORE DEVICE can be used to vary the device offline or online to multiple systems as one operation. For further information, see the description of “REMOVE and RESTORE DEV (API)” in IBM Tivoli System Automation for z/OS Programmer’s Reference.

- If the command fails, display pathing status with "DISPLAY HOST" on page 308, "DISPLAY RESULTS" on page 316, or "DISPLAY VARY" on page 325. Consider rerunning the command with the Force and Disable options.

**Examples**

```
REMOVE S 200 D
IHVC000I REMOVE command successful
```
RESET CHANGECHECK

Purpose

Use the RESET CHANGECHECK command to limit the check for data changes in a switch or not when processing a connectivity command. Normally, all opened switches are checked for a data change prior to the execution of a connectivity command. If any switch responds with a data change indication or cannot respond to the check the connectivity command is rejected. Limiting the check to the switches that are actually affected by the connectivity command avoids the rejection of the command by any other switch.

Syntax

RESET CHANGECHECK

Parameters

OFF  Specifies that only those switches are checked for a data change prior to a connectivity command which are actually affected by the command.

ON  Specifies that all opened switches are checked for a data change prior to a connectivity command. This is the default setting on the startup of I/O Operations.

Usage

Prior to a connectivity command all opened switches are checked for a data change. If any switch responds with a data change indication or cannot respond to
the check for whatever reason the connectivity command is rejected. In case the switch causing the rejection of the command is actually not affected by the connectivity command you can turn off the CHANGECHECK option forcing I/O Operations only to check the switches that are affected by the command.

Examples

```
RC CO F F
IHVC000I RESET command successful.
```

## RESET HOST

### Purpose

Use the RESET HOST command to include or exclude any participant in the issuing I/O operations's consensus processing.

With this command, you can avoid having the issuing I/O operations stop the command from completing because, for example, a participant could not return a vote.

See "DISPLAY HOST" on page 308 for the related command.

### Syntax

```
Reset Host vtamname [tcphostname] [ON|OFF|PURGE]
```

### Parameters

- **vtamname**
  
  Specifies the VTAM application name, or VTAM ID, of the I/O operations targeted in the command.

  When I/O operations is started, it writes its VTAM ID and the TCP/IP host name to all the switches allocated to it. Should the I/O configuration have no use or restricted use of switches, dynamic communication between I/O operations may not be achieved, in which case the RESET HOST command is used. The RESET HOST command allows an operator or automated program to declare the VTAM IDs of peers to I/O operations. I/O operations that are participating in consensus processing use these VTAM IDs to communicate with each other.

- **tcphostname**
  
  Specifies the TCP/IP host name of the I/O operations targeted in the command.

  When I/O operations is started, it writes its VTAM application name and the TCP/IP host name to all the switches allocated to it. Should the I/O configuration have no or restricted use of switches, dynamic communication between I/O operations may not be achieved, in which case, the RESET HOST command is used. The RESET HOST command allows an operator or automated program to declare the TCP/IP host name of peers to I/O operations. I/O operations that are participating in consensus processing use these TCP/IP host names to communicate with each other.
**RESET HOST**

**ON**
Is the default. It specifies that the issuing I/O operations will include the target I/O operations in its consensus processing. (The issuing I/O operations can include itself.) Should the VTAM ID or TCP/IP host name not be recognized, I/O operations records the name and attempts to establish a communication session with the peer. The new peer is included in all subsequent consensus processing. If the name resolves to a TCP/IP host name, I/O operations attempts to establish a TCP/IP connection to the host. Otherwise, a VTAM application name is assumed and I/O operations attempts to establish a VTAM session.

**OFF**
Specifies that the issuing I/O operations will exclude the target I/O operations in consensus processing. (The issuing I/O operations can exclude itself.)

**PURGE**
Specifies that the issuing I/O operations will exclude the targeted I/O operations from participating in its consensus processing until the target host re-registers its VTAM name or TCP/IP host name when it sends a command to one of the switches allocated to both the purged I/O operations and the I/O operations that issued the purge, when the purged I/O operations is restarted, or when a RESET HOST command with the ON operand is issued. Use this option to exclude systems that have ended abnormally or when an incorrect VTAM ID or TCP/IP host name was entered using the RESET HOST command with the ON operand. (The issuing I/O operations cannot purge itself.)

**Usage**

The RESET HOST command affects an I/O operations's participation only in consensus processing initiated by the issuing I/O operations. It does not affect consensus vary path processing initiated by any other I/O operations, including a broadcast initiated by a host that has been reset off.

When the VTAM ID or TCP/IP host name is not known command processing for RESET HOST name ON is as follows:

1. The VTAM ID or TCP/IP host name is recorded by I/O operations and is included in subsequent consensus processing. Should the name not be successfully recorded the following messages are issued:
   IHVC002I Reset command failed. RC = 8 Reas= 5000000A
   IHVC001I Reset command successful
2. If the name resolves to a TCP/IP host name an attempt is made to establish a TCP/IP connection. Otherwise, an attempt to establish a VTAM session is made. If the session is successfully established with the peer its VTAM ID or TCP/IP host name is automatically passed to other I/O operations in the domain. Then each I/O operations dynamically establishes its own session to the peer. The following message is issued:
   IHVC001I RESET command successful
3. If the TCP/IP connection or the VTAM session is not successfully established the TCP/IP host name or VTAM ID is retained by I/O operations, but the name is not broadcast to other peers. This can be caused, for example, by the specification of an incorrect TCP/IP host name or VTAM ID. Should this occur the following messages are issued:
   IHVC001I Reset command successful. RC = 4 Reas= 5000000A
   IHVC401I Host(s) unable to process the RESET command
   Additional information is presented in the Display Results output that indicates the VTAM session was not established.
Examples

```
R H IHVAPPLb OFF
IHVC000I RESET command successful

R H IHVAPPLb ON
IHVC000I RESET command successful

R H IHVAPPLb PURGE
IHVC000I RESET command successful
```

Examples of Using RESET HOST to Declare Peer I/O Operations

**No Switch Configuration:**

In the above figure all three I/O operations start without the knowledge of their peers. To establish communication among them, two RESET HOST commands are required, issued on any of the I/O operations. For example, on IHVAPPL1, the following commands are issued:

```
R H IHVAPPL2 ON
IHVC000I RESET command successful

R H IHVAPPL3 ON
IHVC000I RESET command successful
```

This results in all I/O operations establishing a communication session between them.

**Restricted Switch Configuration:**

```
R H IHVAPPL2 ON
IHVC000I RESET command successful
R H IHVAPPL3 ON
IHVC000I RESET command successful
```

In the above figure, I/O operations IHVAPPL1 and IHVAPPL2 dynamically establish a session with one another at startup. IHVAPPL3 and IHVAPPL4 do so as well. However, neither group knows about the other due to a lack of a switch attachment. To establish communication among the two groups, only a single RESET HOST command is required and may be issued on any of the I/O operations. For example, on IHVAPPL1, the following command is issued:
This results in all I/O operations establishing a communication session between themselves.

**RESET SWITCH**

**Purpose**

Use the RESET SWITCH command to update the issuing I/O operations' configuration map about a single switch or all the switches that are allocated to it.

**Syntax**

```
Reset Switch swchdevn
```

**Parameters**

`swchdevn|*`

Specifies one switch or all switches (* not enclosed in parentheses) allocated to the issuing I/O operations.

For a discussion, see “Switch Identifiers” on page 539.

**Usage**

- If you specify `RS *`, thus resetting all the switches, the issuing I/O operations tries to allocate, or attach, all the switches defined to its host image. In this way, you can provide access to a new switch device number to the issuing I/O operations.
- Make sure you enter this command when you add or modify a switch allocated to the issuing I/O operations during operations.
- Issues a RESET SWITCH command if manipulation of CHPDs or Channel Paths is performed outside I/O operations (for example, using the system console).

**Examples**

```
R S 100
IHVC0001 RESET command successful
```

The following data is updated:

- The VTAM application names of the I/O operations that participate in consensus vary-path processing initiated by the issuing I/O operations application.
- Switch device number. (See “Switch Identifiers” on page 539.)
- Port names and port attributes. (See “Switching Commands” on page 546.)
- Chaining information.
RESET TIMEOUT

Purpose

Use the RESET TIMEOUT command to specify the limit of the time interval, in seconds, that any participating I/O operations will wait for votes to be returned when it broadcasts as the issuing I/O operations.

Syntax

```
[Reset Timeout] seconds
```

Parameters

```
seconds
```

Specifies the number of seconds from 0 up to and including 999,999 or the IBM-supplied default of 60 seconds (* not in parentheses).

Usage

- If the issuing I/O operations receives responses from all eligible voters (all voters except any that have been reset off by the issuing I/O operations) before the time limit is reached, and if all the votes are affirmative, command processing continues.

  Unless the command is being forced, either a negative vote or a failure to respond within the time limit causes the issuing I/O operations to fail the command.

Examples

```
R T 120
IHVC000I RESET command successful
```

RESTORE CHP

Purpose

Use the RESTORE CHP command to configure a specified channel online to the host image that the target I/O operations is running on. See “REMOVE CHP” on page 332 for information on the reciprocal command.

Syntax

```
[RESTORE Chp] chpid|vtamname|tcphostname-Vary-NoForce-NoCheck-BAckout
```

```
[CONDITIONAL] [UNCONDITIONAL]
```

**Parameters**

*chpid*

Specify the CHPID to be restored as a hexadecimal number in the range `X'00'` through `X'FF'`.

*vtamname*

Specify the VTAM application name of the target I/O operations. The target program is the I/O operations that instructs its host image to configure the specified CHPID online.

The target system can be any I/O operations host that a VTAM session is established with.

You can specify THIS-SYS in uppercase characters if the issuing I/O operations is the target program and any of the following conditions holds:

- The issuing I/O operations does not have a valid VTAM name and TCP/IP host name.
- The issuing I/O operations' VTAM name and TCP/IP host name are not available.
- VTAM and TCP/IP are not available.

**Note:** Use the Display Host * command to see if the issuing I/O operations calls itself THIS-SYS.

*tcphostname*

Specify the TCP/IP host name of the target I/O operations.

The target I/O operations is the program that instructs its host image to configure the specified CHPID offline.

The target system can be any I/O operations host that a TCP/IP connection is established with.

You can specify THIS-SYS in uppercase characters if the issuing I/O operations is the target program and any of the following conditions holds:

- The issuing I/O operations does not have a valid VTAM name and TCP/IP host name.
- The issuing I/O operations' VTAM name and TCP/IP host name are not available.
- VTAM and TCP/IP are not available.

**Options**

**VARY**

VARY is the default and NOVary is not an alternative.

The target I/O operations configures the CHPID online before issuing the vary path-online requests for all the relevant paths that are defined to the specified CHPID in the host I/O Control Data Set (IOCDS).

For a discussion of why it sometimes isn't obvious that the status of a path or a device has changed after an operator has issued a connectivity command, see "Device and Path Status After Connectivity Commands" on page 562.

**NOForce|NOCheck**

NOForce is the default. Although you can enter NOCheck, it has no additional effect in this command.
RESTORE CHP

BACKOUT | NOBACKOUT
BAckout is the default. Although you can enter NOBackout, I/O operations’s processing is not different for this command.

For both BACKOUT and NOBACKOUT, I/O operations:

- Attempts to configure the channel back offline if the program is running under MVS and timed out with the status of the channel uncertain.
- Does not attempt to vary any paths back offline if one or more relevant vary path-online requests fail.

CONDITIONAL
This is the default option for both the REMOVE and RESTORE CHP commands. It indicates that no special configure offline or configure online action should be performed.

UNCONDITIONAL
For the REMOVE CHP command, this option puts the specified chpids immediately into pending offline status, even if the chpids are currently active, allocated, or reserved.

For the RESTORE CHP command, this option brings the specified chpids online, even if there are no paths to the chpids, or if the chpids are pending offline and boxed.

Usage
- The CHPID type must be known to the target I/O operations.
- If the RESTORE CHP command causes one or more devices to be made available to the host image that the target I/O operations is running on, I/O operations automatically attempts to add these devices to its internal configuration map.
- I/O operations does not attempt to configure the channel online if the target I/O operations is reset off or purged from intersystem communication.
- The target I/O operations cannot bring a channel online if that channel is not available to the host image that the target program is running on.
- If the target I/O operations is running in an MVS environment and if the program timed out with the state of the CHPID unknown, I/O operations attempts to configure the channel back offline.
- If you are considering specifying a CVC channel for this command, note that I/O operations does not automatically attempt to configure a CVC CHPID offline when it processes any of the following commands for a path defined with that CHPID: Chain, Restore Switch, or Unblock. You can, however, use the RESTORE CHP command to configure the CVC CHPID online.
- The command is not supported if the target I/O operations is running on a host system that is operating as a guest in a VM environment.

RESTORE SWITCH

Purpose

Use the RESTORE SWITCH command to restore logical path status through a switch. I/O operations automatically issues any vary path requests that are needed to synchronize path status with the current physical settings stored at the switch.

If the switch is not allocated that will now be done.
RESTORE SWITCH

See "REMOVE SWITCH" on page 334 for the reciprocal command and "Restore (Enable) a Switch" on page 550 for a discussion.

Syntax

```
/>RESTORE Switch swchdevn
/  Vary NOForce BBackout
/  NOCheck NOBackout
/  Novary Force NOBackout
```

Parameters

swchdevn
Specifies the switch to be restored. For a discussion, see "Switch Identifiers" on page 539.

Options

Connectivity options
For a discussion on Vary | NOVary, NOForce | NOCheck | Force, and BBackout | NOBBackout see "Making Connectivity Changes" on page 543.

Usage

• The Restore Switch command restores paths attached to the switch according to the current switch settings-to units.
• If you have previously issued the command "REMOVE SWITCH DISABLE" you must do the following before you issue the "RESTORE SWITCH" command:
  – For ESCON switches, either POWER®-ON or IML the switch.
  – For FICON switches, use the Fabric Manager to set the switch online.
• If you replace a switch that was disabled with the Remove Switch command with another switch, any previous chaining data is made void. Chaining data in the new switch is used, instead. If the replacement switch is physically linked to another switch, you must define the chained path. See "Chain and Unchain a Switchable Path (ESCON only)" on page 551 for more information.

Examples

```
RESTORE S 100
IHVC0001 RESTORE command successful
```

SYNC SWITCH

Purpose

Use the SYNC SWITCH to build and, optionally, to issue vary path requests that would be required to synchronize path status with the current settings maintained at the specified switch.
SYNC SWITCH

Syntax

```
SYNC Switch swchdevn
  Process
  NOProcess
```

Parameters

**swchdevn**
Specifies the switch. For a discussion, see “Switch Identifiers” on page 539.

Options

**Process**
(default) instructs the issuing I/O operations to issue the appropriate vary path requests.

**NOProcess**
instructs I/O operations to build the list, but not to process it.

Usage

- Use the command described in “DISPLAY VARY” on page 325 to display the vary path requests that would be required if you choose the NOProcess option or the failed vary path requests if you chose the process option.
- Consider using the command if you have entered one or more connectivity commands with the Force option, knowing that pathing status will no longer be synchronous with the switch settings.
- Consider using the SYNC SWITCH command if connectivity changes have been made to the switch at the switch console.
- SYNC SWITCH differs from RESET SWITCH in that the latter command synchronizes the issuing I/O operations' internal configuration map with the physical settings of the switch, but does not synchronize path status.

UNBLOCK

Purpose

Use the UNBLOCK command to permit data transfer through a port that had been blocked. See “BLOCK” on page 297 for the reciprocal command.

Syntax

```
UNblock (portaddress)
  swchdevn
  portname
  generic_name
  NOForce
  NOCheck
  NOBackout
  Vary
  NOVary
  NOCheck
  NOBackout
  NOForce
  NOBackout
```

Chapter 3. I/O Operations Commands 345
UNBLOCK

Parameters

(portaddress)|portname|generic_name

Specifies the port to be unblocked by its port address (enclosed in parentheses), by its individual port name, or by a generic port name. For a discussion, see “Switching Commands” on page 546.

swchdevn|*

Specifies one switch or all switches (* not enclosed in parentheses) allocated to the issuing I/O operations. For a discussion, see “Switch Identifiers” on page 539.

Options

IGNore

You must specify this option when an Inter-Switch-Link port (E_Port) is involved. Otherwise the command is rejected with return code 8 and reason code X'49'. The reason is I/O operations can no longer guarantee “safe-switching” when an E_Port is involved.

Connectivity options

For a discussion on Vary|NOVary, NOForce|NOCheck|Force, and BAckout|NOBAckout see “Making Connectivity Changes” on page 543.

For a discussion of why it sometimes isn’t obvious that the status of a path or a device has changed after an operator has issued a connectivity command, see “Device and Path Status After Connectivity Commands” on page 562.

Usage

If a CVC or parallel CHPID is involved, see “A "Partially-ESCON" Path” on page 554 for more information.

Examples

UNBLOCK (D6) 100
IHVC0001 UNBLOCK command successful

Before                                  After

0100 C0                                  0100 C0
FB D4                                    FB D4
EA D6×                                   EA D6
E0                                       E0

D6 is no longer blocked and allows data transfer

Figure 148. UNBLOCK Command: Sample 1

U CU.F00 *
IHVC0001 UNBLOCK command successful
**UNCHAIN**

**Purpose**

Use the UNCHAIN command to cancel the definition of a chain in a chained path. See "Chain and Unchain a Switchable Path (ESCON only)" on page 551 for more information. See "CHAIN" on page 299 for the reciprocal command.

**Note:** The UNCHAIN command is not applicable to FICON switches.

**Syntax**

```plaintext
UNCHAIN (portaddress_e)\_portname_e \_destination\_swchdevn (portaddress_m)\_portname_m
```

```plaintext
Vary NOForce BAckout NOCheck NOBackout
```

**Parameters**

- **(portaddress_e)\_portname_e**
  - Specifies the end port by its port address (enclosed in parentheses) or by its port name. For a discussion, see "Switching Commands" on page 546.

- **destination\_swchdevn**
  - Specifies the device number of the destination switch in the chained path. For a discussion, see "Switch Identifiers" on page 539.

- **(portaddress_m)\_portname_m**
  - Specifies middle port by its port address (enclosed in parentheses) or by its port name. For a discussion, see "Switching Commands" on page 546.

- **(portaddress_a)\_portname_a**
  - Specifies the port in the chained path that the CHPID or control unit is physically attached to by its port address (enclosed in parentheses) or by its port name. For a discussion, see "Switching Commands" on page 546.
UNCHAIN

**passthru_swchdevn**
Specifies the device number of the passthru switch in the chained path. For a discussion, see “Switch Identifiers” on page 539.

**Options**

**Connectivity options**
For a discussion on Vary|NOVary, NOForce|NOCheck|Force, and BAckout|NOBAckout see “Making Connectivity Changes” on page 543.

**Usage**
- Use the Force option to remove a chained path that is supporting the last path to the switch.
- Make sure that you cancel the definition of a chain before you recable either of the chained switches of the chained ports that are affected.
- The definition of a chain is made void if either of the chained switches is replaced.

**Examples**

```
UNCHAIN (F2) 200 (D4) (C1) 100
IHVC0001 UNCHAIN command successful
```

![Diagram of I/O Operations](image)

*Figure 150. UNCHAIN Command Sample*

UNLOCK

**Purpose**
Use the UNLOCK command to release I/O operations processing from your user ID, so that another user ID can use an I/O operations within the same set. (A set of I/O operations consists of the participants in the issuing I/O operations' vary-path processing and those I/O operations that can communicate with any of the participants.)
UNLOCK

See “GETLOCK” on page 328 for the reciprocal command.

Syntax

```
/SM590000/SM590000
UNLOCK
/SM590000/SM630000
```

Examples

```
UNLOCK
IHVC0001 UNLOCK command successful
```

WRITE

Purpose

Use the WRITE command to assign or write a port name to an addressable port on one or on all the switches that are allocated to the issuing I/O operations.

Syntax

```
Write portname (portaddress) swchdevn
```

Parameters

- **portname**
  - Specifies the port name you want to be assigned. To avoid assigning a port name that contains an unprintable character, specify the name in uppercase letters and use only the following special characters: digits, period, underscore. Do not exceed 24 characters. See “Using Port Names as a Tool in Switching Commands” on page 550 for more information.

- **(portaddress)**
  - Specifies the addressable port that the port name is being assigned to.

- **swchdevn**
  - Specifies one switch or all switches (* not enclosed in parentheses) allocated to the issuing I/O operations. For a discussion, see “Switch Identifiers” on page 539.

Usage

- When assigning a name to a port, consider whether you want that port to be targeted by a generic name.
- The Write command refreshes the data that could be displayed with the Display Results command. Therefore, it is advisable to assign port names when I/O operations is started.
- If you are using the sample ISPF dialog, you can assign a switch device name to a switch. The name will be displayed in the dialog’s display of that switch configuration in matrix format.
  - Either name the name of the switch in the SWCH Name field of the matrix, as described in “A Sample Way to Modify a Switch Matrix” in IBM Tivoli System Automation for z/OS User’s Guide.
  - Or, use the Write command to write a port name to the switch’s control unit port (CUP). If you do not know the CUP address, you can find out with the
WRITE

DISPLAY DEV command. (For example, enter D D 0500 * for switch 500. In the displayed output, the CUP is to the right of CONTROL UNIT DATA.)

Examples

```
W HA.CHP_88.CVC (C0) 100
IHVC0001 WRITE command successful
```

```
I/O Operations

SYSA

88

HA.CHP_88.CVC

C0 C1

FB 0100 D4

EA D6 E0

Figure 151. WRITE Command: Sample 1
```

```
WRITE CU.F00 (D6) *
IHVC0001 WRITE command successful
```

```
Both D6 ports are assigned the same name
```

```
Figure 152. WRITE Command: Sample 2
```
Part 4. SA  z/OS Processor Operations Commands

Chapter 4. Using Processor Operations
Commands .................................. 353
General Information .......................... 353
Host-based Commands ......................... 353
Common Commands .......................... 353
Ensemble Commands .......................... 354
PSM Commands .............................. 354

Chapter 5. Host-based Commands .......................... 355
ISQCCMD .................................. 355
ISQCHK .................................. 358
ISQCMON .................................. 359
ISQECMD .................................. 360
ISQEXEC .................................. 362
ISQHELP .................................. 364
ISQIPSWT .................................. 365
ISQOVRD .................................. 367
ISQROUTE .................................. 368
ISQSEND .................................. 370
ISQSNDH .................................. 372
ISQSTART .................................. 374
ISQSTOP .................................. 376
ISQVARS .................................. 377
ISQXCLS .................................. 392
ISQXCON .................................. 393
ISQXDRL .................................. 396
ISQXDST .................................. 397
ISQXIII .................................. 418
ISQXLOC .................................. 420
ISQXMON .................................. 421
ISQXOPT .................................. 424
ISQXPSM .................................. 426
ISQXUNL .................................. 428

Chapter 6. Common Commands .......................... 431
ACTIVATE .................................. 431
CBU .................................. 434
CCNTL .................................. 437
CONDATA .................................. 438
 CPCDATA .................................. 440
CTRLCONS .................................. 442
DEACTIVATE .................................. 443
EXTINT .................................. 446
GETCLUSTER .................................. 447
GETINFO .................................. 448
GETISTAT .................................. 449
GETSDGR .................................. 450
GETSINFO .................................. 452
GETSSTAT .................................. 453
ICNTL .................................. 454
LOAD .................................. 461
OOCOD .................................. 464
POWERMOD .................................. 466
PROFILE .................................. 468
RESERVE .................................. 480
RESTART .................................. 482
SECLOG .................................. 483
START .................................. 484
STOP .................................. 485
STP .................................. 486
STPDATA .................................. 489
SYRESET .................................. 491
TCDATA .................................. 492
TCM .................................. 494

Chapter 7. Ensemble Commands .......................... 497
ACTIVATE .................................. 497
BCDATA .................................. 499
BLDATA .................................. 500
CONDATA .................................. 502
DEACTIVATE .................................. 503
DISCOVERY .................................. 506
DROP .................................. 507
ENDATA .................................. 509
GETBCSTAT .................................. 510
GETBLSTAT .................................. 511
GETESTAT .................................. 512
GETVHSTAT .................................. 513
GETVSSTAT .................................. 515
HMCSWITCH .................................. 516
LIST .................................. 518
SUBSCRIBE .................................. 521
UNSUBSCRIBE .................................. 523
VHDATA .................................. 526
VSDATA .................................. 527
WLDATA .................................. 529
ZBXDATA .................................. 530

Chapter 8. PSM Commands - Special Requests .......................... 533
ISQACT .................................. 533
ISQCLEAR .................................. 533
ISQMSG .................................. 533
ISQPSM .................................. 534
ISQUERY .................................. 534
ISQSTATUS .................................. 534
ISQTRACE .................................. 535
STOPALL .................................. 535

This part describes SA  z/OS processor operations commands in detail — their purpose, their format, and specifics of how to use them.

For general information about the SA  z/OS commands, see IBM Tivoli System Automation for z/OS User’s Guide.

© Copyright IBM Corp. 1996, 2014
Chapter 4. Using Processor Operations Commands

General Information

Processor operations commands consist of the following basic types:

- Host-based commands
- Common commands

You generally issue processor operations commands at a NetView console from the focal-point system. Alternatively, you can use an NMC, customized for SA z/OS exploitation to issue context-sensitive processor operations commands for target hardware or target system objects by supplementing the prepared command templates.

Commands that start with ISQ are issued as shown in the syntax diagrams, starting with the command name.

Except for the common commands, processor operations commands whose names do not start with ISQ are preceded by the ISQSEND command.

Host-based Commands

The ISQSEND command sends the command to a target operating system for processing, using the processor operations connection to the processor hardware. Processor operations connections from the focal-point to the target hardware use either the support element of the addressed target hardware or can be customized to use an HMC, connected to the target hardware or can be customized to use an HMC, connected to the target hardware LAN for communication. Because the connection to a target hardware and the operating systems running on that hardware is available at target system initialization time or system shutdown time, the ISQSEND command can be used to respond to IPL prompt messages or to answer outstanding replies at system shutdown time.

The ISQSEND command addresses the operating system running on a processor hardware. The command ISQCCMD addresses the processor hardware or a processor hardware (logical partition). The processor hardware that can be monitored and controlled by processor operations provide the OCF (operations command facility). This system management interface is called by ISQCCMD to perform hardware commands, for example, ACTIVATE or SYSRESET.

Other host-based processor operations commands control the start and the stop of the processor operations component of SA z/OS, invoke the status display system, control the monitoring of messages from the target systems, and allow the display and modification of processor operations configuration information.

Common Commands

Common commands are preceded by the ISQCCMD command (described in detail on page "ISQCCMD" on page 355). Use common commands in APIs whenever possible because they provide a single product image across various hardware and software implementations. Regardless of the processor type or the operating system running at your target system, the common command is the same. This can
potentially minimize the need for future modifications to your automation routines should you modify or upgrade your processor hardware or operating system type.

**Ensemble Commands**

Ensemble commands are preceded by the ISQECMD command (described in detail in Chapter 7, “Ensemble Commands,” on page 497). They provide a single operator interface and API for the System zEnterprise Blade Extension (zBX) and its resources, like blades or virtual servers.

In a zEnterprise mainframe environment, up to 8 zEnterprise CPCs, with or without a zBX attachment can be defined as zEnterprise ensemble nodes. You can use the ISQECMD to monitor, control, and automate the zBX resources in this environment.

For the zEnterprise CPC itself, its logical partitions, and the operating systems running in the LPARs, use the ISQCCMD common command and ISQSEND commands for monitoring, control and automation.

**PSM Commands**

A special VM service machine, the ProcOps Service Machine (PSM) is required to monitor and control VM guest systems, which are defined as ProcOps target systems. For problem determination and maintenance purposes a set of PSM commands is provided.

If a VM user is logged on as PSM, these commands can be entered from the PSM’s terminal. Because a PSM normally runs in VM disconnected mode, a subset of the PSM commands can be routed from a ProcOps focal point NetView to the PSM, using command ISQXPSM. See “ISQXPSM” on page 426 for more information about the ISQXPSM command.
Chapter 5. Host-based Commands

ISQCCMD

Purpose

The ISQCCMD command issues a processor operations common command (connection types 'SNMP' and 'TCPIP') or an SA z/OS LPAR management command (connection type 'INTERNAL').

Syntax

```
/SM590000/SM590000
ISQCCMD
   target_system_ProcOps_name
   common_command
   -target_hardware_name.LPAR_name
   -group_name
   -subgroup_name
   -target_hardware_name
/SM590000/SM630000
```

Requirements

ProcOps command environment

• Processor operations must be active.
• The addressed target system must be initialized unless the specified common command is ACTIVATE (see "ACTIVATE" on page 497).

LPAR management environment

• SA z/OS must be fully initialized.
• The processor hardware of the addressed target system or target hardware LPAR must be configured for connection type INTERNAL in the SA z/OS customization dialogs.

Parameters

target_system_ProcOps_name
   Is the name of a target system in the ProcOps or LPAR management environment.

target_hardware_name.LPAR_name
   Is the identifier of the target system using its LPAR location on the specified target hardware. The period is used to separate the hardware from the LPAR name.

group_name
   Is the name of a valid group for ProcOps. The group must contain subgroup names or target system names or both.

subgroup_name
   Is the name of a valid subgroup for ProcOps. The subgroup must contain target system names.

target_hardware_name
   Is the name of the target hardware definition that is associated with the target system.
ISQCCMD

common_command
Represents the common command and any associated parameters to be processed. Multiple parameters are separated from each other by a blank space.

Defaults
None.

Usage
Use the ISQCCMD command to issue a HW common command to the specified target object of a ProcOps or LPAR management environment.

Restrictions and Limitations
ProcOps environment
- Target_hardware_name is valid for the common commands CBU, CCNTL, CONDATA, CPCDATA, CTRLCONS, DEACTIVATE, GETCLUSTER, GETSDGR, GETSINFO, GETSSTAT, OOCOD, PROFILE, TCDATA, TCM, STPDATA, STP, and POWERMOD.
- Group_name or subgroup_name must be processor operations-enabled as defined in the SA z/OS customization dialogs to be a valid parameter.

LPAR management environment
- Target_hardware_name is valid for the common commands CCNTL, CONDATA, CPCDATA, CTRLCONS, GETCLUSTER, GETSDGR, GETSINFO, GETSSTAT, PROFILE, TCDATA and STPDATA.

Return Codes
- 0 Command completed execution.
- 8 Command processing has been rejected.
- 12 Command processing has failed.
- 32 The required environment for processor operations or LPAR Management does not exist. The ISQCCMD command was not processed.

Using LPAR management and ProcOps for processors defined with multiple connection protocols
With the SA configuration dialogs you can specify ProcOps SNMP and INTERNAL(BCPii) connections in a single processor entry. If both connections are configured as active, you can no longer use the ISQCCMD target_system_name form for LPAR Management commands; instead you must use the ISQCCMD target_hardware_name.LPAR_name form.

Failure to do so will result in always executing the LPAR Management request over the SNMP connection. If the system name is not defined to SA, message ISQ400I will be issued, informing you that the target system name is not known.

The dual protocol selection allows only using the target hardware names that have to be specified for each protocol, as the differentiator between the protocols, but not the target system name.

If you still want to use the target system name to differentiate between the LPAR management and ProcOps common commands for a single processor, you must
define separate processor entries and associate system entries to them through the SA dialogs. Define one processor-system set for the ProcOps SNMP connection, and the other set for the INTERNAL connection protocol.

**Specifying an SNMP Timeout Value**

The default timeout value can be modified by setting the NetView task global variable ISQ.SNMP.WAIT to another value. The timeout format is hh:mm:ss, where 'hh' represents the hours of the timeout, 'mm' the minutes and 'ss' of the period to wait for hardware command completion before terminating ISQCCMD. The value of the variable is checked each time before a ISQCCMD request is performed. If the timeout variable is not set, or it has an invalid format, the default value of the corresponding command is used.

In the following REXX example a timeout value of 8 minutes and 30 seconds is set prior calling ISQCCMD to perform an EXTERNAL interrupt hardware command:

```rexx
tme = 'ISQ.SNMP.WAIT' // SNMP timeout varname //
mytgt = 'KEY6' // our TGT system name //
Interpret tme"=" '00:08:30'' // Set timeout to use //
'GLOBALV PUTT 'tme // Update task variable//
Say 'SNMP Timeout value set to 'Tglobal(tme) // Send info msg //
'ISQCCMD 'mytgt' 'EXTINT' // External irpt to Sys//
If RC > 0 Then Do // Error occurred //
End // ...more to add //
Else Do // All went well //
End // ...more to add //
```

**Using Immediate ISQCCMD Common Command Responses**

The immediate command responses cannot be captured with TRAP and WAIT in programs that issue the ISQCCMD command. You can retrieve the immediate HW responses by using a PIPE KEEP stage in your programs. Use 'ISQ.SNMP' as the KEEP name. Note that this command set is also the LPAR management command set and that the KEEP ISQ.SNMP is valid for the LPAR management commands:

- "CCNTL" on page 437
- "CONDATA" on page 502
- "CPDATA" on page 440
- "ICNTL" on page 454
- "GETCLUSTER" on page 447
- "GETINFO" on page 448
- "GETIPSW"
- "GETISTAT" on page 449
- "GETSDGR" on page 450
- "GETSINFO" on page 452
- "GETSSTAT" on page 453
- "PROFILE" on page 468
- "STPDATA" on page 489
- "TCDATA" on page 492

In the following REXX example, the maximum weight that is allowed for a system's LPAR is retrieved and the HW response report containing this information is displayed:

```rexx
mytgt = 'KEY4' // our TGT system name //
'ISQCCMD 'mytgt' ICNTL CMD(READ) VAR(PWMX)' // Max weight //
If RC > 0 Then Do // Error occurred //
End // ...more to add //
Else Do // All went well //
'PIPE KEEP ISQ.SNMP | CONS ONLY' // ...display HW report//
End // ...more to add //
```
ISQCHK

Purpose

The ISQCHK command checks whether processor operations is active.

Syntax

```
> ISQCHK command_name
```

Requirements

None.

Parameters

`command_name`

Is the name of an automation procedure.

Defaults

None.

Usage

This command is intended for use in automation procedures to determine whether processor operations is active. It provides a method of exiting from the automation procedure cleanly when processor operations is not active. If you issue ISQCHK from an operator station and processor operations is running, you will receive no response.

The optional `command_name` parameter is placed in the text of the error message generated when processor operations is not active. This identifies the automation procedure that cannot run because processor operations is not active.

For example, place the following statements near the beginning of a processor operations automation procedure that is written in REXX:

```
'ISQCHK STARTALL'
If RC<>0 then; Exit;
```

If you issue the STARTALL command when processor operations is not active, the error message will inform you that "the STARTALL command" cannot be run until processor operations is active, rather than simply informing you "the command" cannot be run until processor operations is active.

You can issue ISQCHK from an operator station task (OST) or from automation.

Restrictions and Limitations

None.

Return Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Processor operations is active.</td>
</tr>
<tr>
<td>32</td>
<td>Processor operations is not active.</td>
</tr>
</tbody>
</table>
ISQCMON

Purpose

The ISQCMON command causes the current NetView task to be added to or removed from the list of tasks to receive processor operations communications management messages.

Syntax

```
ISQCMON ON
```

Requirements

None.

Parameters

ON  Adds the current NetView task to the list of tasks to receive processor operations communications management messages. This is the default.

OFF  Removes the current NetView task from the list of tasks to receive processor operations communications management messages.

Defaults

ISQCMON defaults to ON, adding you to the list.

Usage

You can issue ISQCMON when processor operations is not running to add or delete yourself from the interested operators list for messages relating to the control of processor operations-managed resources. These messages are not associated with a specific target system.

Messages not associated with a specific target system are sent only to operators who used ISQCMON to place themselves on the interested operators list for such messages. These messages are not sent to the ISQXMON interested operator list. You are automatically added to the ISQCMON interested operator list when you issue the ISQSTART command (see “ISQSTART” on page 374).

See IBM Tivoli System Automation for z/OS User’s Guide for further information about these two types of messages and interested operator lists.

Restrictions and Limitations

Do not issue this command with ISQEXEC.

Return Codes

0  The ISQCMON command completed successfully.

8  You used incorrect command syntax.
ISQECMD

Purpose

The ISQECMD command shell issues a Processor operations ensemble command to manage zEnterprise Ensemble and zBX (Blade Center Extension) resources.

Syntax

```
\%ISQECMD -target_hardware_ProcOps_name ensemble_ProcOps_name real_ensemble_name ensemble_command
```

Requirements

- Processor operations must be active.
- Processor operations must be in session with the ensemble HMC that has the targeted resource in its scope. Session initialization to a ensemble HMC can be configured to happen at ISQSTART time automatically, or can be done when processor operations is already active, using the ISQXIII command.

Parameters

- `target_hardware_ProcOps_name`
  - Is the name of a defined target hardware. It must be selected as member in an ensemble definition. For a target hardware only membership to a single ensemble is supported.

- `ensemble_ProcOps_name`
  - Is the entry name of an ensemble definition in the SA z/OS policy data base. The configuration dialogs enforce that the name is unique and in uppercase.

- `real_ensemble_name`
  - Ensemble name as defined using an ensemble HMC. See “Restrictions and Limitations” on page 361 for additional information.

- `ensemble_command`
  - Represents the actual ensemble command and any associated parameter to be processed. Multiple parameters are separated from each other by a blank space.

Defaults

None.

Usage

Use the ISQECMD command to issue an ensemble command to the specified target ensemble. The individual ensemble command may address additional zBX resources that can be managed over the ensemble HMC connection used.

Ensemble commands can be grouped in the following categories:
- zBX resource notification subscription
- zBX resource discovery
- zBX resource query and list and
• zBX resource management

For Online Help information about the available ensemble commands you can use the ISQHELP interface and select: E Access Help panels for ProcOps Ensemble Commands

Restrictions and Limitations

• If a ISQECMD command target or ensemble command parameter value is defined in mixed case on the ensemble zBX environment, you must use the ‘NetVasis ISQECMD’ form to make sure that the data is passed without uppercase translation to the case-sensitive ensemble zBX environment. Failure to do so may result in ISQECMD command or ensemble command errors.

• In the ensemble definitions of the System Automation policy, you must define the real ensemble name exactly as it is defined on the HMC. You cannot use the real ensemble name as ISQECMD target parameter if this name contains any blanks.

• If you have multiple sessions to different ensemble HMCs active on a processor operations FP domain and these HMCs are located in different HW LAN environments, duplicate real ensemble names may be present. The ISQECMD does not recognize this. Always the first occurrence of the real ensemble name satisfies the internal name search.

Return Codes

0  Command completed execution successfully.
8  The command syntax is not correct.
12 Command failed.
32 The processor operations environment does not exist. The ISQECMD command was not processed.

Specifying a Timeout Value

The default timeout value can be modified by setting the NetView task global variable ISQ.ZBX.WAIT to another value. The timeout format is hh:mm:ss, where ‘hh’ represents the hours of the timeout, ‘mm’ the minutes and ‘ss’ the seconds of the period to wait for hardware command completion before terminating ISQECMD. The value of the variable is checked each time before a ISQECMD command request is performed. If the timeout variable is not set, or it has an invalid format, the default value of the corresponding command is used.

In the following REXX example a timeout value of 4 minutes and 30 seconds is set prior calling ISQECMD to perform a DISCOVERY of all virtual servers in this ensemble domain.

```rexx
tme = 'ISQ.ZBX.WAIT' // zBX timeout varname //
myens = 'R35ZBX' // our ENS name //
'PIPE LIT /00:04:30/ | VAR (TASK) ’tme // Our tglb timeout //
Say ’zBX Timeout value set to ’global(tme) // Send info msg //
'ISQECMD ’myens' DISCOVERY VS' // Do the discovery //
If RC > 0 Then Do // Error occured //
  End // ...more to add //
Else Do // All went well //
  End // ...more to add //
```

Using ISQECMD Ensemble Command Responses

Immediate command responses cannot be captured with TRAP and WAIT in programs that issue the ISQECMD. You can retrieve the immediate responses by using a PIPE KEEP stage in your programs. Use 'ISQ.ZBX' as the KEEP name. All
ensemble commands support this PIPE KEEP. In the following REXX example, a blade extension summary is requested from the addressed ensemble. If the information is available, it is displayed using the 'ISQ.ZBX' data KEEP.

```rexx
cmd = 'ISQECMD 'myens' ZBXDATA'
If RC > 0 Then Do
   // Error occurred
Else Do
   // All went well
End
End
```

If you issue a ISQECMD from the NetView console, the immediate command responses are returned as ISQ800I messages in the netlog. If you used the ISQXMON command to add yourself to the interested operator list for messages and notifications from the targeted ensemble, you will receive copies of these messages as ISQ801I messages on the screen.

Asynchronous ensemble command responses (for example, ACTIVATE) will not provide the complete response in the 'ISQ.ZBX' PIPE KEEP. In order to see the asynchronous parts of an ensemble command response you must have subscribed your operator/task to receive at least JOB and STATUS notifications from the ensemble object (for example, VS, BL). At ensemble connection initialization time, the processor operations ISQXIII command subscribes the ensemble target control task (ISQET*) assigned to the targeted ensemble to receive JOB and STATUS notifications automatically. Use the SUBSCRIBE/UNSUBSCRIBE ensemble commands to modify the processor operations standard subscriptions for the ensemble notifications.

You can find notifications as ISQ800I messages in the netlog. If you want to see copies of the notification messages on your NetView screen, use the ISQXMON command to add your operator/task as an interested operator for messages and notifications from the ensemble.

The following ensemble commands are implemented as asynchronous commands: ACTIVATE, DEACTIVATE.

Note, that the ISQECMD command shell issues message ISQ856I as its final message. There may be notification messages outstanding, related to a just executed asynchronous command, arriving after the ISQECMD command completed.

---

**ISQEXEC**

**Purpose**

The ISQEXEC command sends a command to a target control task to be processed. Commands sent by ISQEXEC are processed by the target control task in the order in which they are presented.

**Syntax**

```
ISQEXEC target_system_name target_hardware_name target_ensemble_name command_name
```
Requirements

The requirements for this command to complete successfully are:

- Processor operations must be active.
- The target entity specified with target_system_name or target_ensemble_name must be initialized. If target_ensemble_name was specified, at least one target system running on that hardware must be initialized.

Parameters

- **target_system_name**
  Specifies the name of the target system.
- **target_hardware_name**
  Specifies the name of the target hardware (processor).
- **target_ensemble_name**
  Specifies the name of the target ensemble. (zEnterprise ensemble for zBX management)
- **command**
  Is the name of the user automation command to be processed.

Defaults

None.

Usage

The main advantage of the ISQEXEC command is that it frees the operator station task (OST) for other work because commands issued by ISQEXEC are processed by a target control task rather than in the OST.

You may require that a set of commands to a target connection are processed in the order in which the commands are presented (that is, serially). ISQEXEC is processed by a target control task, which ensures that the commands are processed in the correct order. A set of commands to be processed serially can be packaged in an automation procedure and be called by the ISQEXEC command, for example:

```
ISQEXEC target_system_name console_connection exec_name exec_operands
```

You can write automation procedures that issue ISQSEND commands. ISQSEND sends commands to a target system. You can enter the ISQEXEC command in the NCCF command line to cause an automation procedure to be processed.

Restrictions and Limitations

The following restrictions or limitations apply to this command:

- Do not issue ISQXDST, ISQHELP, or ISQXOPT with ISQEXEC.
- Do not issue any user automation with ISQEXEC that results in a full-screen panel.
- ISQEXEC does not lock a target console connection; the serialized commands can be interrupted by commands from other tasks. See "Usage."
ISQEXEC

Return Codes
0  The ISQEXEC command completed successfully and it caused the command to be queued to a target control task.
8  The command syntax is not correct.
10 The specified target system was not initialized.
12 An internal error occurred.
32 The processor operations environment does not exist. The ISQEXEC command was not processed.

ISQHELP

Purpose
The ISQHELP command provides help on using processor operations commands.

Syntax
```
ISQHELP
   ISQ
   command_name
   ISQCCMD common_command_name
   ISQECMD ensemble_command_name
```

Requirements
The NetView program must be running for this command to complete successfully.

Parameters
ISQ
Indicates the processor operations Main Help Panel is being requested. Not specifying any parameters with the ISQHELP command also presents the processor operations Main Help Panel.

command_name
Is the name of any parameter with which you need help. This can be a processor operations host-based or any parameter valid to the NetView help processor.

ISQCCMD
Indicates that the variable following it is the name of a common command.

common_command_name
Is the name of the common command for which you want help; only the first 3 characters of the common command need to be specified.

ISQECMD
Indicates that the variable following it is the name of an ensemble command.

ensemble_command_name
Is the name of the ensemble command for which you want help.

Defaults
The processor operations Main Help Panel.
ISQHELP

Usage

The ISQHELP command does not require the processor operations environment.

Issue the ISQHELP command to obtain high-level information about processor operations, to obtain help about the syntax of a processor operations command, to obtain help with the output generated by the command, or to see examples of using the command.

See *IBM Tivoli System Automation for z/OS User’s Guide* for further information on the ISQHELP command and for a description of other help functions.

Restrictions and Limitations

The ISQHELP command cannot be issued from a program.

Return Codes

Return codes are not applicable because the ISQHELP command cannot be issued from a program.

ISQIPSWT

Purpose

The ISQIPSWT command is used to switch IP addresses stored in the processor operations variables:

**SEADDR**

The IP address or host name of the SE (or HMC) that the target hardware is connected to. This value is used by processor operations as the primary information to connect to the target hardware.

**HMCADDR**

The IP address or host name that is used by processor operations as alternate information to connect to the target hardware.

This command can be used, for example, to switch to the secondary SE network adapter card if the connection to the primary adapter card of the SE fails.

Syntax

```
ISQIPSWT target-hardware-name
```

Requirements

Processor operations must be active.

For the target hardware, both variables SEADDR and HMCADDR must be set (that is, not be blank).
Parameters

`target-hardware-name`

Is the name of the target hardware.

FORCE

FORCE(NO)

Check the connection to SEADDR; switch IP addresses only if the connection to SEADDR fails.

FORCE(YES)

Switch IP addresses even if the current SEADDR can still be connected.

Defaults

If no FORCE parameter is specified, FORCE(NO) is assumed by the command.

Usage

It is assumed that the processor operations variable HMCADDR contains the IP address or host name of the secondary network adapter card of the SE.

Use the ISQIPSWT command to switch the connection address that is used to connect to the target hardware. By default, processor operations will use the value that is stored in the variable SEADDR to connect to the target hardware. Upon execution, the command will:

- Close active systems running on the target hardware
- If the target hardware was using a connection to a HMC: check if other target hardware entities use the same connection (that is, the same IP address); close all active systems for these target hardware entities, as well
- Check that the target hardware has a connection status of 'NOT_CONNECTED'
- Switch values between processor operations variables SEADDR and HMCADDR
- Re-start the previously closed systems using the alternate IP address
- If the target hardware was using a connection to a HMC: for all target hardware entities that used the same IP address, switch values between processor operations variables SEADDR and HMCADDR for each target hardware entity and try to restart all previously closed target systems.

In order to restore the original settings, simply execute ISQIPSWT command again.

Restrictions and Limitations

The command can be executed only for target hardware connected using SNMP.

The specified target hardware name must be defined for processor operations.

If HMCADDR contains the IP address of a HMC, ISQIPSWT will be executed only if no active connection to that IP address exists for another target hardware using the same IP address.

Return Codes

0 Command executed successfully.
4 Target hardware name blank or invalid.
5 Invalid FORCE option specified.
6 Processor operations variable SEADDR or HMCADDR is (or both are) blank.
7 Alternatc IP address cannot use an already active HMC connection of one or more Target Hardware using the same HMC IP address.
8 Error occurred retrieving a processor operations variable.
9 Connection to SEADDR still active and FORCE(YES) not specified.
10 Connection test to alternate address (HMCADDR) failed.
11 IP address mismatch: this happens if after a connection start, the value of processor operations variable SEADDR was changed using ISQVARS common command.
12 Target hardware name is not SNMP connected.
13 Target hardware still has the status 'CONNECTED' after closure of all active target systems.
14 Reactivate of systems failed.
16 Initialize pending for some target systems of the target hardware.
32 Processor operations not started or active.

ISQOVRD

Purpose

The ISQOVRD command unconditionally removes an entity lock. Entities can be either target system names together with a console designator or ensemble names.

Syntax

```
>>>ISQOVRD target_system_ProcOps_name OC
               target_hardware_name S SC
               ensemble_ProcOps_name
```

Requirements

The requirements for this command to complete successfully are:
• The use of this command usually requires special authorization.
• Processor operations must be active.
• The target_system_ProcOps_name syntax requires that the addressed target system be initialized.

Parameters

target_system_ProcOps_name
  Is the name of the target system.
  OC Specifies the active operator console (not valid for the coupling facility).
  SC Specifies the active system console

target_hardware_name
  Is the name of the target hardware.
  S Specifies the active system console.

ensemble_ProcOps_name
  Is the ProcOps name of a zBX ensemble.
ISQOVRD

Defaults
None.

Usage
The ISQOVRD command removes the entity lock established by any command or automation procedure.

This command is intended to be used for emergencies only. Overriding a reserve can cause processing to fail, with unpredictable results.

Restrictions and Limitations
The following restrictions or limitations apply to this command:
• This command does not establish an entity lock for the issuing operator, it simply removes the lock on the entity.
• This command does not interrupt or cancel any command or automation procedure currently running.

Return Codes
0 The ISQOVRD command completed successfully; the lock has been removed.
4 You used syntax that is not valid, specified an entity name that is not valid, or specified a target system that is not initialized.
8 A NetView CNMLOCK service problem occurred.
32 The processor operations environment does not exist; the ISQOVRD command was not processed.

ISQROUTE

Purpose
The ISQROUTE command sends a command to a control task to be processed. Commands sent by ISQROUTE are processed by the control task in the order in which they are presented.

Syntax

```
---ISQROUTE--entity_name--command_name---
```

System environment
The requirements for this command to complete successfully are:
• Processor operations must be active.

Parameters

entity_name
Target hardware name:
 Specifies the name of the target hardware.

Target system name:
 Specifies the name of the target system.
Ensemble name:
Specifies the name of the ensemble.

command
Is the name of the user automation command to be processed.

Defaults
None.

Usage
The main advantage of the ISQROUTE command is that it frees the operator station task (OST) for other work because commands issued by ISQROUTE are processed by a processor operations control task rather than in the OST.

This is similar to the ISQEXEC command, however ISQROUTE accepts target system, target hardware, or ensemble names for control task selection. In addition, the entity needs not to be in a required status as this is the case for ISQEXEC, where target systems must have a status of INITIALIZED.

You can require that a set of commands to a target entity are processed in the order in which the commands are presented (that is, serially). ISQROUTE is processed by a control task, which ensures that the commands are processed in the correct order. A set of commands to be processed serially can be packaged in an automation procedure and be called by the ISQROUTE command, for example:

  ISQROUTE target_system_name exec_name exec_operands

You can write automation procedures that issue ISQSEND commands. ISQSEND sends commands to a target system. You can enter the ISQROUTE command in the NCCF command line to cause an automation procedure to be processed.

Restrictions and Limitations
The following restrictions or limitations apply to this command:

- Do not issue ISQXDST, ISQHELP, or ISQXOPT with ISQROUTE
- Do not issue any user automation with ISQROUTE that results in a full-screen panel.
- ISQROUTE does not lock a target console connection; the serialized commands can be interrupted by commands from other tasks. See “Usage.”

Return Codes
0   The ISQROUTE command completed successfully and it caused the command to be queued to a control task.
8   The command syntax is not correct or entity name is not defined.
12  An internal error occurred.
32  The processor operations environment does not exist. The ISQROUTE command was not processed.
Purpose

The ISQSEND command sends a command to a target system for execution, using the processor operations target hardware connection and the console integration facility.

Note: This is an Application Programming Interface.

Syntax

```
ISQSEND target_system_ProcOps_name target_hwname.Lparname OC command
```

Requirements

The requirements for this command to complete successfully are:

- Processor operations must be active.
- The addressed target system connection must have been initialized.

Parameters

**NETVASIS**

Prefix the ISQSEND command with NETVASIS if you want to pass the command text in lower or mixed case.

**target_system_ProcOps_name**

The name of the target system.

**target_hwname.Lparname**

The target hardware name and logical partition name, separated by a period.

This is an alternative to the target system name notation, for all operating systems running on a target hardware activated in LPAR mode.

**OC**

Specifies the active operator console function as the target for an operating system command.

**SC**

Specifies the active system console function as the target for an operating management command.

**command**

The command that is to be processed and any parameters that are associated with that command. Use the ISQSEND command to send operating system commands or operations management commands. If you want to pass the command text in lower or mixed case, prefix the ISQSEND command with NETVASIS.

Note that if you omit the command parameter, a nullstring is sent to the hardware, which is received and may be acknowledged by the target operating system or hardware. See your hardware or operating system documentation for more information.

Operating System Commands

This ISQSEND command type is valid for all ProcOps connections. The following operating systems are supported: MVS (z/OS), VM (z/VM), VSE (z/VSE®), LINUX (Linux on System z), zAware (Advanced Workload Analysis Reporter) and CF (Coupling Facility Control Code).
In addition, Standalone Utility programs running on System z or CMOS/390 hardware, such as SADump (Standalone Dump).

Operations Management Commands
This type of ISQSEND commands is valid only for ProcOps TCPIP connections. For ProcOps SNMP connections, ISQCCMD must be used to perform Operations Management functions.

Operations management commands using TCPIP means that VM/CP commands can be sent to the virtual machine of a ProcOps target system, which is running as a VM guest. Only CP Query commands are supported. For all other operations management functions, use the ISQCCMD command.

Defaults
None.

Usage
For target systems running on processors supported by processor operations, operating system commands can be entered through ISQSEND. The commands are transmitted to the operating system through console integration, an interface provided by the operating system and the processor hardware support element (SE) or hardware management console (HMC).

Restrictions and Limitations
The following restrictions or limitations apply to this command:

- With SA z/OS, the ISQSEND command is the only way to send data to the operating system of a target through the console integration connection. The information that is passed to the target system is not interpreted by SA z/OS.
- Successful ISQSEND completion does not indicate successful completion of the command itself. It only indicates that the command was successfully delivered to the receiver. Your automation routines should interpret the asynchronous message responses of the command to determine the command execution results.
- Note that the operating systems supporting console integration may have commands to activate or deactivate this function, or may provide facilities and commands to prevent response messages from being routed to console integration. For MVS operating systems, the VARY CN(*),DEACTIVATE and VARY CN(*),ACTIVATE commands are available for the deactivation and reactivation of console integration.

Return Codes
0  The ISQSEND command completed successfully and the message was sent.
8  Either the message was not sent, the message was sent with errors, or the specified target system was not initialized.
32  The processor operations environment does not exist. The ISQSEND command was not processed.

Specifying an SNMP Timeout Value
For target hardware connections of type SNMP, the default timeout value of one minute can be modified by setting the NetView task global variable ISQ.SNMP.WAIT to another value. The timeout format is hh:mm:ss, where 'hh' represents the hours of the timeout, 'mm' the minutes and 'ss' the seconds of the
ISQSEND

period to wait for HW command completion before terminating ISQSEND. The value of the variable is checked each time a ISQSEND request for an SNMP connected target system is performed. If the timeout variable is not set, or it has an invalid format, the default value of one minute is used.

In the following REXX example a timeout value of 8 minutes and 30 seconds is set prior calling ISQSEND:

tme = 'ISQ.SNMP.WAIT' // SNMP timeout varname //
mytgt = 'KEY6' // our TGT system name //
Interpret tme"= '00:08:30'' // Set timeout to use //
'GLOBALV PUTT 'tme // Update task variable//
Say 'SNMP Timeout value set to 'Tglobal(tme) // Send msg //
'ISQSEND 'mytgt' D T' // External irpt to Sys//
If RC > 0 Then Do // Error occured //
End // ...more to add //
Else Do // All went well //
End // ...more to add //

ISQSNDH

Purpose

The ISQSNDH command sends a high priority command to a target system for execution, using the processor operations target hardware connection and console integration facility.

Note: This is an Application Programming Interface.

Syntax

```
/SM590000/SM590000
NETVASIS ISQSNDH

  target_system_ProcOps_name target_hwname.Lpar_name OC command

/SM590000/SM630000
```

Requirements

The requirements for this command to complete successfully are:

- Processor operations must be active.
- The addressed target system connection must have been initialized.

Parameters

**NETVASIS**

Prefix the ISQSEND command with NETVASIS if you want to pass the command text in lower or mixed case.

**target_system_ProcOps_name**

The name of the target system.

**target_hwname.Lpar_name**

The target hardware name and logical partition name, separated by a period. This is an alternative to the target system name notation, for all operating systems running on a target hardware activated in LPAR mode.

**OC**

Specifies the active operator console as the target for an operating system command.

**command**

Is the command to be processed and any parameters that are associated with
that command. Use the ISQSNDH command to send operating system commands or operations management commands with priority flag set.

If you want to pass the command text in lower or mixed case, prefix the ISQSNDH command with NETVASIS.

Note that if you omit the command parameter, a nullstring is sent to the hardware, which is received and may be acknowledged by the target operating system or hardware. See your hardware or operating system documentation for more information.

Defaults

None.

Usage

For target systems running on processors supported by processor operations, operating system commands can be entered through ISQSNDH. The commands are transmitted to the operating system through console integration, an interface provided by the operating system and the processor hardware support element (SE) or hardware management console (HMC).

Note: This command is supposed to be used to reply to critical Operating System messages (reply ID of Zero). In order to help to detect such messages, the corresponding ISQ900I message contains console identifier ‘OCP’ instead of ‘OC’, for example,

ISQ900I CPC1.LPAR1 OCP +00 IEAxxxx <message text>

Specifying an SNMP Timeout Value:

The default timeout value of one minute can be modified by setting the NetView global task variable ISQ.SNMP.WAIT to another value. The timeout format is hh:mm:ss, where ‘hh’ represents the hours of the timeout, ‘mm’ the minutes and ‘ss’ the seconds of the period to wait for HW command completion before terminating ISQSNDH. The value of the variable is checked each time a ISQSNDH request for an SNMP connected target system is performed. If the timeout variable is not set, or it has an invalid format, the default value of one minute is used.

Restrictions and Limitations

The following restrictions or limitations apply to this command:
• The ISQSNDH command is supported for connection type SNMP only.
• Successful ISQSNDH completion does not indicate successful completion of the command itself. It only indicates that the command was successfully delivered to the receiver. Your automation routines should interpret the asynchronous message responses of the command to determine the command execution results.
• Note that the operating systems supporting console integration may have commands to activate or deactivate this function, or may provide facilities and commands to prevent response messages from being routed to console integration. For MVS operating systems, the “VARY CN(*),DEACTIVATE” and “VARY CN(*),ACTIVATE” commands are available for the deactivation and reactivation of console integration.
Return Codes
0  The ISQSNDH command completed successfully and the message was sent.
8  Either the message was not sent, the message was sent with errors, or the
    specified target system was not initialized or has a THW connection path
    status of SUSPENDED.
32 The processor operations environment does not exist. The ISQSNDH command
    was not processed.

Purpose
The ISQSTART command starts processor operations. It establishes the processor
operations environment with the NetView program and permits the control of
target resources with processor operations commands. Such resources are: Target
systems or virtual systems, mainframe processors or virtual machines, and
zEnterprise BladeCenter Extensions. This command can be issued from an
automation routine.

Syntax
/SM590000/SM590000
ISQSTART
   ACF DEBUG
   ACFCLEAN
   Optionfile

Requirements
If you specify:
  • With parameter ACF or ACFCLEAN, SA z/OS initialization must be complete
    before ISQSTART can be used
  • With the Optionfile parameter, the NetView program must have at least read
    authority to the dataset

You may need special authorization to use this command.

Parameters
ACF
  If specified, processor operations is started, taking the required configuration
  information from the current active automation control file (ACF).

ACFCLEAN
  If specified, the INGCLEAN command is called to perform a clean up of the
  in-storage data model on the processor operations focal point NetView domain,
  before the ISQSTART ACF processing starts.

  The processor operations configuration information is stored in ACF entry
types like: PROCESSOR, SYSTEM, AOF?ENS, and GROUP. Since none of these
entry types is associated with type SUBSYSTEM, a delete or rename of a
processor operations related ACF entry will not result in an automatic cleanup
of the no longer valid associated global variables during a System Automation
refresh.
ISQSTART

This can cause problems or ISQSTART time for processor operations runtime, especially when no longer available HW resources are still present in the processor operations data model.

Using the INGCLEAN before starting processor operations is one way to avoid such problems.

Optionfile
An additional debug or configuration source. This parameter is for IBM Service support only.

DEBUG
Provides additional diagnostics that are written to the NetView log.

Defaults
There are no defaults for this command when processor operations is first started: you must specify the ACF, ACFCLEAN or Optionfile parameter. When restarting, the command defaults to the parameter used for the previous initial start.

Usage
To perform an initial start, use the ACF or ACFCLEAN parameter. For a restart (warmstart), this parameter is not needed.

An initial start loads the processor operations configuration data from the active ACF and starts all processor operations tasks to establish the processor operations task environment. The configuration data also indicates which, if any, target system or ensemble connection should be initialized (not the IPL process).

The initial start with the ACF parameter takes the available configuration data for processor operations from the loaded ACF to build the processor operations runtime environment. In case processor operations configuration data was changed using the configuration dialogs and a new ACF file was built, you first refresh the ACF on the processor operations FP system with the new user configuration, before this new configuration data is available for processor operations. Use the ISQSTOP command to processor operations before issuing the ISQSTART ACF or ISQSTART ACFCLEAN command to activate the changes in the ProcOps runtime environment.

You usually perform a restart after a processor operations task has failed to restart all processor operations tasks to reestablish the processor operations tasking environment. If the processor operations environment has not already been established by a previous ISQSTART command that specified ACF/ACFCLEAN or Optionfile, the restart will fail.

When you issue ISQSTART, it adds you to the interested operators list that is accessed by the ISQCMON command. This list is for messages that are not associated with a specific target system. The messages relate to the control of processor operations-managed resources.

Restrictions and Limitations
The following restrictions or limitations apply to this command:
• If the ACF or ACFCLEAN start parameters are used, SA z/OS on the processor operations focal point system must be fully initialized, otherwise ISQSTART processing will fail.
ISQSTART

Note that this restriction also applies if ISQSTART is issued while an ACF REFRESH is in progress on the processor operations focal point system.

- A restart re-establishes the previous configuration. The processor operations environment must exist (an initial start must have already been performed) before a restart can be performed.
- Processor operations must be stopped before an initial start can be performed.
- The XML_config_file name cannot be DEBUG.
- The specified XML_config_file must be a sequential data set.

Return Codes

0  Command accepted.
4  You used incorrect syntax.
8  An internal command error occurred.

ISQSTOP

Purpose

The ISQSTOP command stops processor operations. It can be issued in an automation routine.

Syntax

```
ISQSTOP [DEBUG]
```

Requirements

You may require special authorization to use this command.

Processor operations must be active.

Parameters

DEBUG

Provides additional diagnostics, which are written to the NetView log.

Defaults

None.

Usage

The ISQSTOP command immediately stops the processor operations environment and all processor operations tasks. If you do not have the shutdown of processor operations automated via SA z/OS, issue ISQSTOP and ISQSTART as part of normal shutdown procedures to restart failed processor operations tasks or to establish a new operating environment (see the "ISQSTART" on page 374 command).

Restrictions and Limitations

The ISQSTOP command does not shut down target systems.
All current processor operations status information is lost when you issue ISQSTOP. You should exit any displays presented by the ISQXDST, or ISQXOPT commands or ProcOps SDF Interface panels showing LPARs, before stopping processor operations.

**Return Codes**

0  The ISQSTOP command completed successfully.
4  You used incorrect syntax.
8  An internal command error occurred.
32 The processor operations environment does not exist; the ISQSTOP command was not processed.

**ISQVARS**

**Purpose**

The ISQVARS command lets you view and change certain SA z/OS values. Table 3 on page 384 identifies the keywords for the values you can view or change with the ISQVARS command.

**Syntax**

```plaintext
ISQVARS
  GET
    TGT target_system_ProcOps_name
    keyword
    OCF ocf_name
    THW target_hardware_name
    LUN luname
    ENS target_ProcOps_ensemble
    GRP group_name subgroup_name
  GETL
  GETT
  GETC
  PUT
    TGT target_system_ProcOps_name
    keyword
    OCF ocf_name
    THW target_hardware_name
    LUN luname
    ENS target_ProcOps_ensemble
    GRP group_name subgroup_name
    'value'
    variable
  PUTL
  PUTT
  PUTC
  LIST
    TGTLIST
    THWLIST
    GRPLIST
    SBGLIST
    ENSLIST
    variable
```

**Requirements**

Processor operations must be active.
ISQVARS

Parameters

GET
Specifies that you want to retrieve the value of a SA z/OS keyword to be displayed or to be placed into a specified NetView local variable.

GETL
Specifies that you want to retrieve the value of a SA z/OS keyword to be displayed or to be placed into a specified NetView local variable. This keyword is the same as the GET keyword. This is the default if you specify GET.

GETT
Specifies that you want to retrieve the value of a SA z/OS keyword to be displayed or to be placed into a specified NetView task variable.

GETC
Specifies that you want to retrieve the value of a SA z/OS keyword to be displayed or to be placed into a specified NetView common variable.

TGT
Indicates that the variable following it is the name of the target system.

target_system_ProcOps_name
Is the name of the target system.

OCF
Indicates that the variable following it is the name of the target system’s secondary operations command facility (OCF). A secondary OCF is used on an OCF-based processor in LPAR mode to represent a logical partition and to process commands for that CPC image (logical partition). The name of the secondary OCF is specified on the “Target System Description” panel for OCF-based processors in the configuration dialogs. On a System/390® microprocessor cluster, the name of the secondary OCF is the same as the name of the logical partition and of the Image profile used to activate the partition.

ocf_name
Is the name of the target system’s secondary OCF.

THW
Indicates that the variable following it is the name of the target hardware.

target_hardware_name
Is the name of the target hardware.

LUN
Indicates that the variable following it is the logical unit name.

luname
Is the logical unit name.

GRP
Indicates that the variable following is a group or subgroup name.

group_name
Is the name of the group of which the subgroup is a member.

subgroup_name
Is the name of the subgroup of which the target system is a member.

ENS
Indicates that the variable following it is the name of an ensemble.
**target_ProcOps_ensemble**

Is the name of the ensemble.

**keyword**

Is the name of the SA z/OS value to be retrieved or set. Table 3 on page 384 identifies the SA z/OS values that can be read or changed by the ISQVARS command.

**MSG**

Indicates the SA z/OS value is to be displayed on the console as line-mode output.

**variable**

Is the name of the NetView variable whose value is to be set to that of the specified SA z/OS keyword. Local variable names are most commonly used when values are being read and written from within automation procedures. The names of the SA z/OS values are defined in Table 3 on page 384.

**PUT**

Specifies that you want to set the value of a SA z/OS keyword to either the current value of a specified NetView local variable or to a value specified in the command string.

**PUTL**

Specifies that you want to set the value of a SA z/OS keyword to either the current value of a specified NetView local variable or to a value specified in the command string. This keyword is the same as the PUT keyword. This is the default if you specify PUT.

**PUTT**

Specifies that you want to set the value of a SA z/OS keyword to either the current value of a specified NetView task variable or to a value specified in the command string.

**PUTC**

Specifies that you want to set the value of a SA z/OS keyword to either the current value of a specified NetView common variable or to a value specified in the command string.

**TGT**

Indicates that the variable following it is the name of the target system.

**target_system_ProcOps_name**

Is the name of the target system.

**THW**

Indicates that the variable following it is the name of the target hardware.

**target_hardware_name**

Is the name of the target hardware.

**LUN**

Indicates that the variable following it is the logical unit name.

**luname**

Is the logical unit name.

**GRP**

Indicates that the variable following is a group or subgroup name.

**group_name**

Is the name of the group of which the subgroup is a member.
**ISQVARS**

F

`subgroup_name`

Is the name of the subgroup of which the target system is a member.

**ENS**

Indicates that the variable following it is the name of an ensemble.

`target_ProcOps_ensemble`

Is the name of the ensemble.

**keyword**

Is the name of the SA z/OS keyword whose value is to be retrieved or set. [Table 3 on page 384] identifies the values that can be read or changed by the ISQVARS command.

**value**

Is the value to be assigned to the SA z/OS keyword. The straight single quotation marks are required.

**variable**

Is the name of the NetView variable which contains the value to which the SA z/OS keyword is to be set (PUT). Local variable names are most commonly used when SA z/OS values are being read and written from within automation procedures. The keywords used for SA z/OS values are defined in Table 3 on page 384.

**LIST**

Indicates the keyword following it identifies which list is being referenced. This keyword is valid from automation programming only.

**TGTLIST**

Refers to the list of defined target system names.

**THWLIST**

Refers to the list of defined target hardware names.

**GRPLIST**

Refers to the list of defined target group names.

**SBGLIST**

Refers to the list of defined target subgroup names.

**ENSLIST**

Refers to the list of defined target ensemble names.

**variable**

Is the name of the NetView variable whose value is to be set to that of the specified SA z/OS keyword. Local variable names are most commonly used when values are being read and written from within automation procedures. The names of the SA z/OS values are defined in [Table 3 on page 384].

**Defaults**

None.

**Usage**

There are several ways to specify the desired information:

**Target system name**

get|put TGT target_system_ProcOps_name keyword...

**Target hardware name**

get|put THW target_hardware_name keyword...
You can display (on the NetView console) the value of a NetView variable by using the MSG keyword.

The ISQVARS command allows access to SA z/OS values. Values relating to target systems or ports require the name of the component (target system processor operations name or port name). The keyword is a processor operations-defined character string identifying the name of the processor operations value.

If issued from a program, a NetView variable is specified to contain the SA z/OS value.

- In the GET case, the specified NetView variable (variable) is set to the specified SA z/OS value (keyword).
- In the PUT case, the specified SA z/OS value is set to the current value of the specified NetView variable, or to the value specified within straight single quotation marks.

For example, assume target system names system1, system2, and system3 have been defined. The following command:

```
ISQVARS GET LIST TGTLIST TARGETS
```

will place the following character string into the NetView local variable targets:

```
system1 system2 system3
```

**Restrictions and Limitations**

The following restrictions or limitations apply to this command:

- SA z/OS values are defined through the use of configuration panels during installation. Not all of these values can be changed by the ISQVARS command. Table 3 on page 384 identifies the SA z/OS values that can be changed by the ISQVARS command.
- Any changes you make to a SA z/OS value with the ISQVARS command remain in effect only for the current session of SA z/OS. Stopping and then starting SA z/OS resets the variables.
- GET LIST is valid only from within automation programming. You cannot issue GET LIST from an OST.
- GET LIST has the following length restrictions:
The NetView variable is restricted to 32767 bytes when a REXX command processor calls the ISQVARS command using the NetView local variable form (GETL LIST). The ISQVARS command truncates the data to this length if necessary.

The NetView variable is restricted to 255 bytes when a REXX command processor, high-level language, or NetView command list calls the ISQVARS command using the NetView task variable form or NetView common variable form (GETT | GETC LIST). The ISQVARS command truncates the data to this length if necessary.

- Updates for 'Path Poll Frequency' (pollfreq) or 'Path Poll Retries' (pollrtry) have no impact on the polling task itself; to effectively change these values, the update must be performed in the SA z/OS Customization Dialog Processor Information panel (AOFGCPD0); afterwards, rebuild and reload the automation control file and restart processor operations.

Caution: Use ISQVARS with care. Value changes are made immediately and may seriously affect the running of SA z/OS.

Return Codes
0 Processed without errors.
30 The processor operations environment does not exist; the ISQVARS command was not processed.
32 The processor operations environment does not exist; the ISQVARS command was not processed.
131 You did not enter enough parameters.
132 The second argument you specified is unknown.
133 You specified a form of the command that is not valid with the list option.
134 msg and list options are mutually exclusive.
135 Neither get nor put were specified.
136 You used an uneven number of single straight quotation marks.
137 You used single straight quotation marks other than for the last argument.
138 Unable to write to the specified variable.
139 Unable to read the specified variable.
141 msg is a keyword and cannot be used as a variable name.
142 Variable answer could not contain all of the data, variable truncated to 255 bytes.
143 Variable answer could not contain all of the data, variable truncated to 32767 bytes.
144 Addressed port not initialized to a target system.
145 System console not defined for addressed target system.
161 You specified a target hardware name that is not valid.
162 You specified an unknown keyword.
163 You specified a target system name that is not valid.
166 You specified a keyword restricted to LPAR systems.
167 The specified port is not connected to an operator console.
168 The specified port is not connected to a system console.
169 IOCDS cannot be located, system is in LPAR mode.
170 The port definition and the target hardware definition in the SA z/OS configuration panels conflict with one another — the system console cannot be found.
171 The port definition and the target hardware definition in the SA z/OS configuration panels conflict with one another — the operator console cannot be found.
172 You specified a keyword that is not valid for the PUT function.
173 NetView CNMLOCK failure.
Internal locking sequence error.
Unable to obtain lock for PUT function.
You specified a PUT function with non-hexadecimal characters for a keyword field that can contain only hexadecimal characters; the command is ignored.
You specified a PUT function with non-integer characters for a keyword field that can contain only integer characters; the command is ignored.
Unknown operating system type for specified target system.
The specified keyword only valid if the addressed target system is running the MVS or VM operating systems.
You tried to write (PUT) values to a keyword field that are not valid values for that field.
You tried to write (PUT) a value that is too large for the keyword field to which it is being written.
You tried to write (PUT) a time offset value that is not valid.
You specified a parameter that cannot be used with the keyword you specified (the parameter values that can be specified are list, tgt, ens or thw).
It is not valid to change the internal variable you specified from its current status value to the status value you specified.
It indicates that the status of the target system or logical partition cannot be identified.
You specified a logical partition name that is not valid.
You specified a variable that requires a qualified name (such as target.lpar), but the qualified name you provided is not valid.
Logical partition name missing. You specified a target hardware name, but it was not followed by a logical partition name (such as target.lpar).
The memory requested is larger than the maximum that can be installed.
The system name entered is not a valid target system name, but it is a valid target hardware name.
You specified a target ensemble name that is not valid.

ISQVARS Keywords

SA z/OS provides keywords you can use to define your configuration and to determine the status of the elements in your configuration.

When you develop automation routines, you need to read these keyword values and perhaps write to them as well. For example, a routine may change the primary IPL address used by automation. Assuming the old address was 03FE, the following command changes the address to 0123:

```
ISQVARS PUT TGT SILVER pipla '0123'
```

This section explains the keywords and the valid values each keyword accepts. It also identifies the SA z/OS messages that trigger SA z/OS automation to update the keyword values. You may want to use these status messages to begin your own automation routines.

**Variable Value Lengths:** In addition to using the correct form of ISQVARS for the variable type, you must be aware of the length limits of variables for your particular programming environment. This is a particular concern when you are retrieving long data strings such as lists of defined target systems. If you are programming in REXX, the variable value length limit is 32 KB. For other programming environments supported by NetView, the variable value length limit is 256 bytes.
**ISQVARS**

**Keyword Valid Values:** This column shows the values of SA z/OS keywords. You can retrieve or change these values with the ISQVARS command if they are valid values.

In Table 3, the SA z/OS values are sorted according to the contents of the Name column, which identifies the keyword name of the value. The Access column indicates the type of access allowed for the keyword through the ISQVARS command. A keyword may have either read-only access, designated by *read* (GET), or read/write access, designated by *write* (PUT).

The ISQVARS column indicates which command keyword you should use after the GET or PUT keyword of the ISQVARS command to access the SA z/OS value.

**SA z/OS Keywords and Status Messages:** Table 3 lists the ISQVARS keywords sorted by keyword name.

### Table 3. ISQVARS Keywords Sorted by Keyword Name

<table>
<thead>
<tr>
<th>Keyword Description</th>
<th>Name</th>
<th>Access</th>
<th>ISQVARS</th>
<th>Valid Values and Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last activated THWNAME</td>
<td>acthw</td>
<td>write</td>
<td>LUN</td>
<td>Allows the obtaining or setting of the last target hardware (THWNAME) to be activated for a specified LUNAME corresponding to a NetView Connection Path. The setting of the THWNAME needs to be done before the THWNAME can be obtained. If PUT specified, input must be equal to a valid THWNAME.</td>
</tr>
<tr>
<td>THW: active target system name</td>
<td>atsname</td>
<td>read</td>
<td>OCF or THW</td>
<td>When OCF, the name of the active target system that is associated with OCF. When THW, the name of the active target system that initialized on the specified target hardware; one name for XA, ESA, and 370, multiple names for LPAR-mode target hardware.</td>
</tr>
<tr>
<td>Backup focal point</td>
<td>backfp</td>
<td>read</td>
<td>THW</td>
<td>Backup focal point NetView domain ID.</td>
</tr>
<tr>
<td>Number of retry steps for calling the Hardware Interface</td>
<td>cmdretry</td>
<td>write</td>
<td>THW</td>
<td>Processor Operations commands will call the Hardware Interface; this call may fail at first, and, for some return codes Processor Operations tries to re-execute the call. This variable defines how often a retry should be executed. The value is set to 10 at Processor Operations start time (ISQSTART). Valid number range is 00 to 99. <strong>Note:</strong> This variable is only used for SNMP connections.</td>
</tr>
<tr>
<td>Number of retry steps when calling the Ensemble to execute a command or function</td>
<td>cmdretry</td>
<td>write</td>
<td>ENS</td>
<td>Processor Operations commands will call the Ensemble Interface; this call may fail in the first place, and for some return codes Processor Operations tries to re-execute the call. This variable defines how often a retry should be executed. Valid number range is 00 to 99.</td>
</tr>
</tbody>
</table>
Table 3. ISQVARS Keywords Sorted by Keyword Name (continued)

<table>
<thead>
<tr>
<th>Keyword Description</th>
<th>Name</th>
<th>Access</th>
<th>ISQVARS</th>
<th>Valid Values and Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wait time between retry steps for the Hardware Interface</td>
<td>cmdwait</td>
<td>write</td>
<td>THW</td>
<td>Processor Operations commands will call the Hardware Interface; this call may fail at first, and, for some return codes Processor Operations tries to re-execute the call. This variable defines the wait time in seconds between 2 retries. The value is set to 01 at Processor Operations start time (ISQSTART). Valid number range is 00 to 99. <strong>Note:</strong> This variable is only used for SNMP connections.</td>
</tr>
<tr>
<td>Wait time between retry steps for calling the Ensemble Interface function</td>
<td>cmdwait</td>
<td>write</td>
<td>ENS</td>
<td>Processor Operations commands will call the Ensemble Interface; this call may fail in the first place, and for some return codes Processor Operations tries to re-execute the call. This variable defines the wait time in seconds between 2 retries. Valid number range is 1 to 600.</td>
</tr>
<tr>
<td>Target hardware connection type</td>
<td>ctype</td>
<td>read</td>
<td>THW</td>
<td>Type of processor operations SNMP to focal-point NetView.</td>
</tr>
<tr>
<td>IPL message: response to duplicate volume IEA213A IPL message</td>
<td>dvolar</td>
<td>write</td>
<td>TGT</td>
<td>Allowed values are 'Y' (required) or 'N' (not required).</td>
</tr>
<tr>
<td>Alternate Ensemble Hardware Management Console Address</td>
<td>ensaddr</td>
<td>read</td>
<td>ENS</td>
<td>IP address or hostname of the Ensemble HMC, defined as the alternate one.</td>
</tr>
<tr>
<td>Ensemble anchor address information</td>
<td>ensanch</td>
<td>read</td>
<td>ENS</td>
<td>Returns the address of shared memory used to store internal data model and connection information in hexadecimal characters.</td>
</tr>
<tr>
<td>Blade center list</td>
<td>ensbcl</td>
<td>read</td>
<td>THW</td>
<td>For target THW, a list of the discovered zBX blade centers is returned in the form: <code>bcname/serial</code>. Multiple <code>bcname/serial</code> pairs are separated by one or more blanks. This list is filled at ISQXIII time for the ensemble where the target member is a member. If the Ensemble Management Flag for the THW is set to NO or blank, or no Blade Center information could be discovered, a blank is returned.</td>
</tr>
<tr>
<td>Target ensemble connection status</td>
<td>enscs</td>
<td>write</td>
<td>ENS</td>
<td>Connection status to the ensemble HMC, used as the SPOC for the targeted ensemble. Valid values are: ACTIVE, ENS SESSION PROBLEM or blank. For read, you may also get the following values in addition to the values you can write: CLOSED, DORMANT, SUSPENDED, CONNECTING.</td>
</tr>
<tr>
<td>Status panel error message</td>
<td>ensmgs</td>
<td>write</td>
<td>ENS</td>
<td>Status panel error message line.</td>
</tr>
<tr>
<td>Ensemble list</td>
<td>enlist</td>
<td>read</td>
<td>GET LIST</td>
<td>List of the defined ensemble entry names in the current ProcOps configuration. The names are separated by one or more blanks.</td>
</tr>
<tr>
<td>Ensemble lock</td>
<td>enslockn</td>
<td>read</td>
<td>ENS</td>
<td>Name of operator who issued the lock; blank if not set.</td>
</tr>
<tr>
<td>Keyword Description</td>
<td>Name</td>
<td>Access</td>
<td>ISQVARS</td>
<td>Valid Values and Comments</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-------</td>
<td>--------</td>
<td>---------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Ensemble management flag</td>
<td>ensflag</td>
<td>write</td>
<td>THW</td>
<td>This flag controls the management of the Ensemble Blade Extension (zBX) for the specified THW. Allowed values are 'YES', 'NO' or blank. If the value is 'YES', ProcOps will try to discover zBX resources like Blade Centers and Blades of the specified THW and allow commands to manage the discovered resources. If the value is 'NO' or blank, no resource discovery is done for the zBX resources of the specified THW at ISQXIII time and subsequent ProcOps cmds to manage the zBX resources for this THW will fail.</td>
</tr>
<tr>
<td>Ensemble entry name</td>
<td>ensname</td>
<td>read</td>
<td>THW</td>
<td>Returns the ensemble entry name if the specified THW is defined as a member node of an ensemble. If no ensemble membership is defined, a blank value is returned.</td>
</tr>
<tr>
<td>List ensemble node information</td>
<td>ensnodes</td>
<td>read</td>
<td>ENS</td>
<td>Returns a list of all Processors (THW) that are part of the ensemble. List entries that are separated by one or more blanks.</td>
</tr>
<tr>
<td>Primary ensemble hardware management console address</td>
<td>enspaddr</td>
<td>read</td>
<td>ENS</td>
<td>IP address or hostname of the Ensemble HMC, defined as the primary one.</td>
</tr>
<tr>
<td>Password to be used for automatic logon</td>
<td>enspw</td>
<td>read</td>
<td>ENS</td>
<td>Password for Ensemble HMC user id to be used for automatic logon. Note that the ISQVARS response message may not contain a password in readable format. If the ISQVARS calling application specifies a target variable for the PW query result, it is this application’s responsibility to ensure that the security guidelines that may be in place, are met.</td>
</tr>
<tr>
<td>Real ensemble name</td>
<td>ensrname</td>
<td>write</td>
<td>ENS</td>
<td>Name of the zEnterprise Ensemble as defined in the zManager environment. This name may contain mixed case characters and can be up to 16 characters long. This variable is filled if the ISQXIII processing was successful.</td>
</tr>
<tr>
<td>Userid used for automatic logon to Ensemble HMC</td>
<td>ensuser</td>
<td>read</td>
<td>ENS</td>
<td>This userid automatically logs on at the Ensemble HMC at ISQXIII time and logs off again when an ISQXCLS is performed for the ensemble.</td>
</tr>
<tr>
<td>Group name list</td>
<td>grpplist</td>
<td>read</td>
<td>GET LIST</td>
<td>A list of group names. The names are separated by one or more blanks.</td>
</tr>
</tbody>
</table>
Table 3. ISQVARS Keywords Sorted by Keyword Name (continued)

<table>
<thead>
<tr>
<th>Keyword Description</th>
<th>Name</th>
<th>Access</th>
<th>ISQVARS</th>
<th>Valid Values and Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware management console address</td>
<td>hmcaddr</td>
<td>write</td>
<td>THW</td>
<td>IP address or hostname of a HMC where the THW is defined. Before changing the value of this variable, make sure that all target systems that share this address as their target hardware connection have been closed using the ISQXCLS command. The modified address value will only be used if no more connections are active from the ProcOps FP to the HMC. Issue command ISQXIII to restart the connections with the changed address.</td>
</tr>
<tr>
<td>Target hardware processor type</td>
<td>htype</td>
<td>read</td>
<td>TGT or THW</td>
<td>Mainframe</td>
</tr>
<tr>
<td>LPAR: partition initial state</td>
<td>ilpars</td>
<td>read</td>
<td>TGT</td>
<td>ACT1VE or INACTIVE (LPAR target systems only).</td>
</tr>
<tr>
<td>IPL message: response to master catalog action IEA347A IPL message</td>
<td>iplmsgmc</td>
<td>write</td>
<td>TGT</td>
<td>75 characters (MVS only).</td>
</tr>
<tr>
<td>IPL message: response to IEA101A IPL message</td>
<td>iplmsgsp</td>
<td>write</td>
<td>TGT</td>
<td>75 characters (MVS only).</td>
</tr>
<tr>
<td>IPL message: VM IPL start message</td>
<td>iplmsgvm</td>
<td>write</td>
<td>TGT</td>
<td>75 characters (VM only).</td>
</tr>
<tr>
<td>Name of the IP stack for the SNMP connection</td>
<td>ipstack</td>
<td>write</td>
<td>THW</td>
<td>The name of the IP stack on the ProcOps FP system that is to be used for this SNMP connection. If defined, the ProcOps FP TCP/IP must be configured for multiple IP stacks. If blank, the default IP stack is used, regardless of the TCP/IP stack configuration. Note that with SNMP connect over a HMC, the IPSTACK definition of the first CPC that is connected is used for all other CPCs of this HMC connection, regardless of the other IPSTACK definitions that may exist for the other CPCs. The ipstack value used must be a valid z/OS jobname, representing the alternate IP stack. Only length checking is performed in ISQVARS.</td>
</tr>
<tr>
<td>Name of the IP stack for the ensemble connection</td>
<td>ipstack</td>
<td>write</td>
<td>ENS</td>
<td>The name of the IP stack on the ProcOps FP system to be used for this ensemble connection. If defined, the ProcOps FP TCP/IP must be configured for multiple IP stacks. If blank, the default IP stack is used, regardless of the TCP/IP stack configuration. The ipstack value used must be a valid z/OS jobname, representing the alternate IP stack. Only length checking is performed in ISQVARS.</td>
</tr>
<tr>
<td>Keyword Description</td>
<td>Name</td>
<td>Access</td>
<td>ISQVARS</td>
<td>Valid Values and Comments</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-------</td>
<td>--------</td>
<td>---------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LPAR: partition name in use?</td>
<td>lparact</td>
<td>read</td>
<td>THW</td>
<td>Null (not logically partitioned) or name of the initialized target system that has the specified LPAR name, in the form target_hardware_name.LPAR_name. The variable contains the name of the initialized target system that has LPAR name or a null value if none of the target systems initialized to the target hardware have that LPAR name.</td>
</tr>
</tbody>
</table>
| LPAR scope                                  | lparscop| read   | THW     | ALL (default) or DEFONLY  
ALL: sessions will discover all detectable partitions on that target hardware.  
DEFONLY: sessions will discover only partitions defined as target systems for that target hardware. |
| NetView connection path LUNAME              | nplu  | read   | TGT or THW | $NA LUNAME for connection to an OCF-based processor support element. |
| NetView connection path NETID               | npnetid| read   | TGT or THW | $NA NETID for connection to OCF-based processor support element. |
| NetView connection path Target Hardware Names | npthw | read   | LUN     | Returns all defined target hardware names for an SNMP connection. |
| NetView connection status summary           | nvcs  | write  | THW     | Valid values are: ACTIVE, SNMP SESSION PROBLEM or blank.  
For read, you may also get CLOSED and DORMANT in addition to the values that you can write. |
| Secondary OCF name                          | ocf2nd| read   | TGT     | Returns the value of the secondary OCF name. |
| Operator console lock                       | ocllockn| read   | TGT     | The name of operator who issued the lock. Blank if it has not been set. |
| Operator console status                     | ocstat| read   | TGT     | ACTIVE, CLOSED, UNKNOWN, LINK ERROR, UNDEFINED, and CONSOLE LOST |
| Target operating system type                | ostype| read   | TGT     | MVS, VM, LINUX, VSE, ZAWARE, or CF |
| IPL: primary IPL address                    | pipla | write  | TGT     | Specifies the device address to be used for the IPL. |
| IPL: primary IPL CP address                 | piplcpua| write | TGT     | Specifies the central processor address (no longer used). |
| IPL: primary IPL parameter                  | piplp | write  | TGT     | 8 characters allowed. |
| LPAR: partition status                      | pnstatus| write | TGT     | 1 (inactive) or A (active). |
| Path poll frequency for the target hardware connection | pollfreq| write | THW     | Defines how often (in minutes) SA z/OS Processor Operations should poll the support element for its status. Valid values are 0 to 99 minutes. Specify 0 to deactivate polling.  
**Note:** Changing the value has no impact on the polling frequency of the polling task itself, see “Restrictions and Limitations” on page 381 for details. |
Table 3. ISQVARS Keywords Sorted by Keyword Name (continued)

<table>
<thead>
<tr>
<th>Keyword Description</th>
<th>Name</th>
<th>Access</th>
<th>ISQVARS</th>
<th>Valid Values and Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path poll frequency for the ensemble connection</td>
<td>pollfreq</td>
<td>write</td>
<td>ENS</td>
<td>Defines how often (in minutes) SA z/OS Processor Operations should poll the ensemble HMC connection. Valid values are 0 to 359 minutes. Specify 0 to deactivate polling. <strong>Note:</strong> Changing the value has no impact on the polling frequency of the polling task itself - see “Restrictions and Limitations” on page 381 for details.</td>
</tr>
</tbody>
</table>
| Path poll option                                         | pollopt | write  | THW     | CPC (default) or CONN  
CPC: Issue CPCDATA request to validate the console application for a target hardware connection.  
CONN: Issue ISQXCON and GETSSTAT to verify only target hardware connection.                                                                                                                                                                                                                   |
<p>| Path poll retries for the target hardware connection     | pollrtry | write  | THW     | Defines the number of attempts SA z/OS Processor Operations should make to poll the support element for its status before notifying the operator that the connection failed. Valid values are 0 to 99 retries. <strong>Note:</strong> Changing the value has no impact on the polling retries of the polling task itself - see “Restrictions and Limitations” on page 381 for details. |
| Path poll retries for the ensemble connection             | pollrtry | write  | ENS     | Defines the number of attempts SA z/OS Processor Operations should try and poll the HMC connection for status before notifying the operator that the connection failed. Valid values are 0 to 99 retries. <strong>Note:</strong> Changing the value has no impact on the polling retries of the polling task itself - see “Restrictions and Limitations” on page 381 for details. |
| Primary focal point                                      | primfp | read   | THW     | Primary focal-point NetView domainid.                                                                                                                                                                                                                                                                                                                                 |
| Force an IPv4 connection to PSM server                   | psmipv4 | write  | THW     | Allowed values are ’Y’ or ’N’. If set to ’Y’, a IPv4 connection will be used although IPv6 could be tried. This is useful in environments where the IPv6 support is not fully enabled or IPv6 related connection problems may exist. In IPv4 only environments the setting of the flag is ignored. <strong>Note:</strong> This flag is only tested if the PSM server is defined using the Hostname address form.                          |
| Subgroup name list                                       | sbglist | read   | GET LIST | A list of subgroup names. The names are separated by one or more blanks.                                                                                                                                                                                                                                                                                  |
| System console lock                                      | sclockn | read   | TGT     | The name of operator who issued the lock. Blank if it has not been set.                                                                                                                                                                                                                                                                                       |
| System console status                                    | scstat  | read   | TGT     | SNMP SESSION PROBLEM, ACTIVE, CLOSED, DORMANT, and blank.                                                                                                                                                                                                                                                                                                      |</p>
<table>
<thead>
<tr>
<th>Keyword Description</th>
<th>Name</th>
<th>Access</th>
<th>ISQVARS</th>
<th>Valid Values and Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support Element address</td>
<td>seaddr</td>
<td>write</td>
<td>THW</td>
<td>IP address or hostname of THW’s Support Element. Before changing the value of this variable, make sure that all target systems that share this address as their target hardware connection have been closed using the ISQXCLS command. The modified address value will only be used if no more connections are active from the ProcOps FP to the HMC. Issue command ISQXIII to restart the connections with the changed address.</td>
</tr>
<tr>
<td>Target hardware site info</td>
<td>site</td>
<td>write</td>
<td>THW</td>
<td>Location information, if provided with the processor for the target hardware.</td>
</tr>
<tr>
<td>Use SNMPv3 protocol</td>
<td>snmpv3</td>
<td>read</td>
<td>THW</td>
<td>YES: indicates that the THW requires SNMPv3 authentication. NO (or empty): default SNMPv2c authentication using SNMP community.</td>
</tr>
<tr>
<td>Target system attention</td>
<td>tattn</td>
<td>write</td>
<td>TGT</td>
<td>Blank, DCCF, EXCEPTIONS, DISABLED WAIT, SERIOUS ALERT, ALERT, and ENV ALERT.</td>
</tr>
<tr>
<td>Authorization token</td>
<td>tauth</td>
<td>read</td>
<td>TGT or THW</td>
<td>Authorization token or SNMP community name of the processor support element or HMC.</td>
</tr>
<tr>
<td>Target system description</td>
<td>tdesc</td>
<td>read</td>
<td>TGT</td>
<td>Target system description which corresponds to the System Short Description in System Automation's Customization Dialog.</td>
</tr>
<tr>
<td>Desired Image profile</td>
<td>tdipf</td>
<td>write</td>
<td>TGT</td>
<td>Desired Image profile name for the specified target system name.</td>
</tr>
<tr>
<td>Desired Load profile</td>
<td>tdlpf</td>
<td>write</td>
<td>TGT</td>
<td>Desired Load profile name for the specified target system name.</td>
</tr>
<tr>
<td>Desired Reset profile</td>
<td>tdrpf</td>
<td>write</td>
<td>TGT</td>
<td>Desired Reset profile name for the specified target hardware name.</td>
</tr>
<tr>
<td>Status Panel Error Message</td>
<td>temsg</td>
<td>write</td>
<td>TGT or THW</td>
<td>Status panel error message line.</td>
</tr>
<tr>
<td>Target System Group name</td>
<td>tgrp</td>
<td>read</td>
<td>TGT or GRP</td>
<td>Target system names within a group and subgroups associated with this group or the group names for a target system.</td>
</tr>
<tr>
<td>Target system name list</td>
<td>tgtlist</td>
<td>read</td>
<td>GET LIST</td>
<td>A list of target system names. The names are separated by one or more blanks.</td>
</tr>
<tr>
<td>Target hardware name list</td>
<td>thwlist</td>
<td>read</td>
<td>GET LIST</td>
<td>A list of target hardware names. The names are separated by one or more blanks.</td>
</tr>
<tr>
<td>Target hardware name</td>
<td>thwname</td>
<td>read</td>
<td>TGT</td>
<td>Name of target hardware definition associated with the specified target system.</td>
</tr>
<tr>
<td>Target hardware status</td>
<td>thwstat</td>
<td>write</td>
<td>TGT</td>
<td>Blank, LPAR DEFINITION PROBLEM, TARGET HARDWARE PROBLEM, POWERED OFF, POWER-ON RESET REQUIRED, SERVICE, POWERSAVE, SERVICE REQUIRED, DEGRADED, UNKNOWN or OK.</td>
</tr>
</tbody>
</table>
### Table 3. ISQVARS Keywords Sorted by Keyword Name (continued)

<table>
<thead>
<tr>
<th>Keyword Description</th>
<th>Name</th>
<th>Access</th>
<th>ISQVARS</th>
<th>Valid Values and Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target time offset</td>
<td>timeoff</td>
<td>write</td>
<td>TGT</td>
<td>A number in the range of -23 through +23 the TGT time deviates from FP.</td>
</tr>
<tr>
<td>Status Panel Info Message</td>
<td>timsg</td>
<td>write</td>
<td>TGT or THW</td>
<td>Status panel informational message line.</td>
</tr>
<tr>
<td>Target system initialized?</td>
<td>tininit</td>
<td>read</td>
<td>TGT</td>
<td>YES or NO</td>
</tr>
<tr>
<td>LPAR: partition name</td>
<td>tpar</td>
<td>read</td>
<td>TGT</td>
<td>Null or LPAR name for the specified target system if the target system is not in an LPAR. Use the TGT form of the command.</td>
</tr>
<tr>
<td>Target system mode</td>
<td>tmode</td>
<td>read</td>
<td>TGT</td>
<td>ESA, LINUX, ZWARE, or CF.</td>
</tr>
<tr>
<td>Target system console status summary</td>
<td>toca</td>
<td>read</td>
<td>TGT</td>
<td>Blank.</td>
</tr>
<tr>
<td>Target system console status summary</td>
<td>tocb</td>
<td>read</td>
<td>TGT</td>
<td>Blank.</td>
</tr>
<tr>
<td>Target system console status summary</td>
<td>tsca</td>
<td>read</td>
<td>TGT</td>
<td>Blank.  SNMP SESSION PROBLEM and ACTIVE</td>
</tr>
<tr>
<td>Target system console status summary</td>
<td>tscb</td>
<td>read</td>
<td>TGT</td>
<td>Blank.</td>
</tr>
<tr>
<td>Target system name</td>
<td>tsname</td>
<td>read</td>
<td>THW or OCF</td>
<td>When THW, names of all defined target systems on specified target hardware name, whether initialized or not (one name for XA, ESA*, and 370, multiple names for LPAR-mode target hardware). When OCF, the target system name associated with a secondary OCF name.</td>
</tr>
<tr>
<td>Target system status</td>
<td>tstat</td>
<td>write</td>
<td>TGT</td>
<td>UNDECIDABLE, UNKNOWN, INITIALIZED, IPL COMPLETE, LOAD FAILED, IPL FAILED, CLOSED, WAITING FOR IEA101A MESSAGE, WAITING FOR IEA347A MESSAGE, WAITING FOR VM START MESSAGE, STAGE-1 ACTIVATE COMPLETE, NOT OPERATING, and NOT ACTIVE.</td>
</tr>
<tr>
<td>Target System SubGroup name</td>
<td>tsubgrp</td>
<td>read</td>
<td>TGT or GRP</td>
<td>Target system names within a subgroup or the subgroup names for a target system.</td>
</tr>
<tr>
<td>Group processor operations-enabled indicator</td>
<td>vgrp</td>
<td>read</td>
<td>GET</td>
<td>Y or N.</td>
</tr>
<tr>
<td>Subgroup processor operations-enabled indicator</td>
<td>vsubgrp</td>
<td>read</td>
<td>GET</td>
<td>Y or N.</td>
</tr>
<tr>
<td>SNMPv3 Password</td>
<td>v3pwd</td>
<td>read</td>
<td>THW</td>
<td>Password for SNMPv3 authentication.</td>
</tr>
<tr>
<td>SNMPv3 Username</td>
<td>v3user</td>
<td>read</td>
<td>THW</td>
<td>Username for SNMPv3 authentication.</td>
</tr>
</tbody>
</table>

**ISQVARS Resource Type:** The ISQVARS command syntax also includes a resource type. Use the correct resource type (TGT, THW, ENS, LIST, PUT LUN, or PUT GRP) for the requested keyword.

**Automation from Status Messages:** When SA z/OS processor operations detects a status change for any element of your SA z/OS configuration, it updates the appropriate status table variables and may generate a message or an alert. You can create a routine and a NetView automation table entry to automate the operator.
**ISQVARS**

response to these status change messages. For the recommended operator response to each of these messages, see IBM Tivoli System Automation for z/OS Messages and Codes.

**ISQXCLS**

**Purpose**

End communication with a target system (console) or ensemble HMC and set the target system or ensemble connection status to CLOSED. For a target system, the communication with the SE/HMC of the target system's processor is also terminated if this was the last INITIALIZED target system on that processor.

When closing target hardware, ISQXCLS closes the console connections to all target systems defined, running on the specified hardware. This implicitly terminates the communication with the target hardware processor's SE/HMC.

**Syntax**

```
ISQXCLS target_system_ProcOps_name
     OC
     SC
     ensemble_ProcOps_name
     target_hardware_ProcOps_name
```

**Requirements**

The requirements for this command to complete successfully are:

- Processor operations must be active.
- The `target_system_ProcOps_name` syntax requires the addressed target system to be initialized.
- No console connection to be closed can be locked by another task.
- No ensemble connection to be closed can be locked by another task.

**Parameters**

- `target_system_ProcOps_name`  
  Is the name of the target system to be closed.
  
  - **OC**  
  Specifies the active operator console connection.
  
  - **SC**  
  Specifies the active path for an SNMP connection.

- `ensemble_ProcOps_name`  
  Is the name of ensemble to be closed.

- `target_hardware_ProcOps_name`  
  Is the name of the target hardware. This is a required parameter, and must be a valid target hardware name.

**Defaults**

None.
ISQXCLS

Usage

You can close a target system, target ensemble, or target hardware to end the communication between the target entity and focal point over the processor operations path.

Closing communication for a Target System: Issue the following to close a target system:

```
ISQXCLS target_system_ProcOps_name
```

Closing a target system sets its status to CLOSED.

Closing communication for an ensemble HMC: Issue the following to close an ensemble:

```
ISQXCLS ensemble_ProcOps_name
```

Closing an ensemble connection sets its status to CLOSED.

Closing communication for a Target Hardware: Issue the following to close console communication to all target systems on a target hardware and finally end communication between the target hardware processor’s SE/HMC and ProcOps:

```
ISQXCLS target_hardware_ProcOps_name
```

Using this command form allows you to close ProcOps console communication with all target systems running on a defined target hardware in one step. The individual ISQXCLS commands for the target systems are scheduled to run on the ProcOps control task of the target hardware.

Restrictions and Limitations

The following restrictions or limitations apply to this command:

- The close of a target system does not end the processing of that target system. Use the common command DEACTIVATE to perform this task.
- Do not close a system console on a logically partitioned system if any logical partition on that target hardware is active.

Return Codes

- **0** The ISQXCLS command completed successfully, no errors occurred.
- **8** You used syntax that was not valid.
- **12** A NetView or processor operations service problem occurred or the THW connection path status is SUSPENDED.
- **16** A severe error occurred.
- **32** The processor operations environment does not exist; the ISQXCLS command was not processed.

ISQXCON

Purpose

This command is used to manage Processor operations connections during events such as the planned maintenance of processor Support Elements (SEs) or Hardware Management Consoles (HMCs) (that are used by Processor operations). In order to prevent error retries and recovery actions by Processor operations, the SUSPENDED status for a target hardware connection and its associated target...
systems can be set while the SEs/HMCs are not available. After maintenance, a RESUME function allows normal Processor operations connection processing to continue.

In addition, the STATUS function provides a consolidated summary of important connection-related information in the multi-line message ISQ350I. This is an alternative to the ISQXDST status dialog panels, which provide similar information, available in different panel views.

Syntax

```
ISQXCON entity_name
```

Requirements

The requirements for this command to complete successfully are:

- Processor operations must be active.
- The SUSPEND/RESUME functions of this command make use of the NetView DSISVRT VSAM DB to save status variables. If your NetView is not customized to use this facility, the ISQXCON SUSPEND/RESUME functions cannot be used, only the STATUS function will work.

Parameters

- `entity_name`
  
  Is the name of a configured target hardware (processor) in your active Processor operations policy.
  
  or

  Is the name of a configured zEnterprise ensemble (zBX) in your active Processor operations policy.

  You can use the '*' wildcard as the last character in your `entity_name` specification to let ISQXCON select all target hardware names, which match the partial string preceding the '*' for processing.

  Using only the '*' wildcard character as `entity_name` selects all configured target hardware and ensemble names.

- `STATUS`

  Generates an ISQ350I multi-line message status report for the selected target entity name(s).

- `SUSPEND`

  For the selected entity name(s), the active connections are first CLOSED to end communication. Then the connection path status is changed to SUSPENDED. This status is saved in the DSISVRT to keep it available across NetView recycles. For target hardware, each configured target system of that target hardware is also set to SUSPENDED. In a SUSPENDED state, Processor operations commands that use this connection will not work. Use the ISQXCON RESUME function to re-enable the entity name connection again for normal Processor operations.
RESUME
For the selected entity name(s), the connection path status is first checked to see if it is currently SUSPENDED. If so, it is set to DORMANT. For target hardware connections the configured target system status is set to CLOSED. This enables normal Processor operations for them. In order to re-establish communication to an entity name, use the ISQXIII command.

FORCE(opt)
The parameter of the SUSPEND function request.

The supported options are YES | NO. With Force Option NO, the SUSPEND request is only processed if no active target system sessions exist to the target hardware over the connection path to be suspended. If Force Option YES is specified, existing target sessions are closed before the connection path is suspended.

For ensembles, force option YES must be specified to suspend an active connection.

MSGKEEP
The Message Output option. If specified, the message output of the command is written to a PIPE KEEP with the name ISQXCON. For compatibility reasons with other Processor operations commands, the successful completion response ISQ017I is not written to the message KEEP. Use a PIPE with KEEP ISQXCON as its first stage to analyse the message output. Each time the command ISQXCON is executed with the MSGKEEP option, the PIPE KEEP is re-initialized. If the MSGKEEP option is not specified, all ISQXCON messages are written to the NetView console.

Defaults
If no parameters are specified for ISQXCON, this will result in a STATUS request for all configured entity name connections. The resulting messages are written to the NetView console.

If the SUSPEND function is used, the FORCE(NO) option is in effect to prevent an unintended close of active target system or ensemble sessions.

Please note that for SUSPENDED connections, no Processor operations session polling is limited.

Usage
When using ISQXCON to manage a single or multiple entity name connections, it is recommended to use the ISQXCON STATUS function first, before suspending or resuming a connection.

Please note that for SUSPENDED connections, no Processor operations session polling is done.

Restrictions and Limitations
The SUSPEND and the RESUME functions of ISQXCON currently accept only the SNMP connection protocol for target hardware names and the HTTP connection protocol for ensemble names. TCP/IP connections to PSMs are not supported.

If multiple target hardware connections are processed in a request, the ISQXCON return code is always 0.
**ISQXCON**

**Return Codes**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The command completed successfully. If multiple entity name connections are processed, but one or more connections have set return codes 6,8,10 and 16, use the responded messages ISQ357I - ISQ359I to determine which of the connections failed and the reason.</td>
</tr>
<tr>
<td>2</td>
<td>Function/option parameter invalid or missing. The ISQXCON command is not executed.</td>
</tr>
<tr>
<td>4</td>
<td>The entity name parameter is not specified or could not be resolved. The ISQXCON command is not executed.</td>
</tr>
<tr>
<td>6</td>
<td>Session is active, FORCE (YES) option is required.</td>
</tr>
<tr>
<td>8</td>
<td>Connection protocol is not supported.</td>
</tr>
<tr>
<td>10</td>
<td>Status is not SUSPENDED.</td>
</tr>
<tr>
<td>16</td>
<td>Suspend/Resume HW command failed.</td>
</tr>
<tr>
<td>32</td>
<td>The Processor operations environment does not exist; the ISQXCON command was not processed.</td>
</tr>
</tbody>
</table>

**ISQXDRL**

**Purpose**

The ISQXDRL command displays the outstanding MVS requests from a target system; the command is rejected for operating systems other than MVS. ISQXDRL issues the following MVS command to the active operator console port of the target system, and sends the response to the requesting NetView operator:

```
DR,L,CN=(ALL)
```

**Syntax**

```
ISQXDRL target_system_ProcOps_name
```

**Requirements**

The requirements for this command to complete successfully are:

- Processor operations must be active.
- The addressed target system must have a status of INITIALIZED.
- The addressed target system must be running MVS.
- The operator console must not be locked by another task.
- You must be on the interested operators list for the operator console to see the reply messages.

**Parameters**

`target_system_ProcOps_name`

Is the name of the target system. It is a required parameter, and must be a valid target system name.

**Defaults**

None.

**Usage**

The ISQXDRL command provides a convenient alternative to using the passthrough facility to issue the MVS DR,L command.
Each line of the MVS response to the DR,L command is displayed as a single-line message. Processor operations does not trap the responses. There is no prefix to the message lines and the last line is followed by a line of dashes. Other messages to the operator may be interspersed with these messages, especially for the last line of dashes. The line of dashes may be issued some time after the last real line of output from the MVS command.

Restrictions and Limitations

The ISQXDRL command works only when issued to systems running MVS.

Return Codes

0  The ISQXDRL command completed successfully.
4  You used syntax that is not valid, specified a target system that is not running MVS, or specified a target system that is not initialized.
8  A problem occurred during the processing of the “ISQSEND” on page 370 command or the console is locked by another task.
32 The processor operations environment does not exist; the ISQXDRL command was not processed.

ISQXDST

Purpose

The ISQXDST command displays status panels, which provide the following information:
• Status summary (all target systems)
• Status summary (all blades of a zBX)
• Individual ensemble connection status
• Individual target hardware status
• Individual target hardware connection status
• Individual target system status
• Interested operator list

Syntax

```
ISQXDST /target_entity_name
```

Requirements

Processor operations must be active.

Parameters

```
target_entity_name
```
This can be:

```
target_system_name
```
Is the name of a valid target system for which you want to see individual target system status. If omitted, the summary status of all target systems is displayed.

```
target.hardware_name
```
Is the name of a valid target hardware for which you want to see
individual hardware information in ProcOps. If the hardware has a zBX attached, you can enter the zBX blade view from here.

**ensemble_name**
Is the name of a valid zEnterprise ensemble for which you want to see individual information and connection status.

**Defaults**

If your ProcOps configuration has target systems and the associated target hardware (processors) defined, issuing the ISQXDST command without a target entity parameter will show the overall status of all target systems.

If your ProcOps configuration has only ensemble definitions, but no target systems and no associated target hardware definitions, issuing the ISQXDST command without a target entity will show the ensemble selection panel.

While in the ISQXDST status system, the NetView® OVERRIDE=NO option is in effect, which suppresses immediate messages sent to your NCCF screen, so the panel navigation is not interrupted. Note, that if you perform the NetView ROLL function, the DISPLAY=NO option is temporarily changed to DISPLAY=DEFAULT.

If you invoke the ProcOps status system using the ISQXDSTI command, the display option currently in effect for you, is not changed. As a result messages directed to your operator ID may interrupt the panel navigation flow.

**Usage**

Issue the ISQXDST command from an operator console to access the processor operations status facility. If you issue the command to access the status summary panel, you will be able to use the status facility to view multiple panels for multiple ProcOps entities without exiting the status facility. Refer to *IBM Tivoli System Automation for z/OS Defining Automation Policy* and *IBM Tivoli System Automation for z/OS User’s Guide* for more information about how to set up and use the status facility.

To display the Status Summary, issue:

```
ISQXDST
```

This will invoke the target system summary.

To display the status of a specific target entity, issue:

```
ISQXDST target_entity_name
```

If you enter the status facility by issuing the ISQXDST command with no other parameters, all of the status panels give you access to the other status panels that are supported by the status facility, including the zBX blade and ensemble views. Some of these status panels cannot be directly accessed by command. Refer to *IBM Tivoli System Automation for z/OS User’s Guide* for further information about how to use the status panels.

To display the status of a specific target entity, issue:

```
ISQXDST ensemble_name
```

From the displayed ensemble status screen, you can navigate to the views of the other ensembles you may have configured.
Restrictions and Limitations

The following restrictions or limitations apply to this command:

- Do not issue this command with ISQEXEC, ISQROUTE or from an automation procedure.
- You can issue the ISQXDST command only from an operator console.

Return Codes

Return codes are not applicable because the ISQXDST command cannot be issued from a program.

Examples

Figure 153 shows an example of the ProcOps Ensemble view panel ISQEENS that is displayed when you enter the command ISQXCMD and you have ensembles defined in your ProcOps configuration, but no target systems.

“How to Monitor and Control Ensemble connections and zBX resources” on page 408 shows an example of the ProcOps Ensemble view panel ISQEENS that is displayed when you enter the command ISQXDST and you have ensembles defined in your ProcOps configuration, but no target systems.

The following values can be displayed on the ISQXDST panel to describe the status of target systems:

CLOSED
(Green) The target system has been closed.

DCCF
(Red) A target system attention status indicating that a DCCF message has been detected.

DISABLED WAIT
(Red) A target system attention status indicating that a disabled wait condition has been detected.
INITIALIZED
(Green) The target system has been successfully initialized.

IPL COMPLETE
(Green, yellow, or red) The meaning of this status depends on the operating system:
• For a TPF or VSE system, the Load or Load Clear process started.
• For a z/OS system, processor operations automation responded to the IEA347A message.
• For a VM system, processor operations automation responded to the VM start message.

If this condition is displayed in red, it means that something is wrong (such as processor operations having lost its communication link with the target system) rather than that the IPL process failed. This condition reverts to green when the problem is corrected.

IPL FAILED
(Red) z/OS only. Error messages were generated during the initial program load process.

LOAD FAILED
(Red) The Load or Load Clear operation did not complete successfully.

NOT ACTIVE
(Yellow) Activate response was received from the target and did not indicate that stage 1 activation was completed. If the target is running in an LPAR, the logical partition found is not activated.

NOT OPERATING
(Red) Target hardware indicates a not operating condition for the image. No CP status is available for problem determination.

POWER-ON RESET REQUIRED
(Red) A POWER-ON RESET of the target system hardware is required before a BCP load.

POWERED OFF
(Red) The target hardware is physically powered off.

POWERSAVE
(Red) SNMP Path Only: Power® utility for the target hardware failed. Target hardware put in power save state.

SERIOUS ALERT
(Red) SNMP Path Only: Target system attention status indicating a serious alert has been detected.

SERVICE
(Red) SNMP Path Only: Service status enabled for the target hardware.

SNMP SESSION PROBLEM
(Red) SNMP Path Only: An attempt to use the SNMP Connection returned an error.

STAGE-1 ACTIVATE COMPLETE
(Yellow) The support element power-on reset has occurred but the load is not yet complete.

SUSPENDED
(Yellow) The connection path to the target hardware of the target system has been suspended. All defined target systems using the suspended
connection path cannot be operated with Proc-Ops until the connection path is resumed again. For suspend/resume operation you can use the command characters of the ISQXDST panel ISQESUM or the ISQXCON command.

UNDECIDABLE
(Yellow) SNMP Path Only: The target hardware is not communicating with the support element or the support element is not communicating with the HMC.

UNKNOWN
(Green) No attempt has been made to initialize or activate this target system. This status can be set only by a cold start of processor operations.

WAITING FOR IEA101A START MESSAGE
(Yellow or red) MVS only: ACTIVATE, ISQXIPL, or ISQXLOD command processing has completed for an MVS system and processor operations is waiting for the IEA101A message on the operator console. If this condition is displayed in red, it means that something is wrong (such as processor operations having lost its communication link with the target system) rather than that processor operations has been waiting for the message too long.

WAITING FOR IEA347A MESSAGE
(Yellow or red) MVS only: processor operations has replied to the IEA101A message and is waiting for the IEA347A message on the operator console. If this condition is displayed in red, it means that something is wrong (such as processor operations lost its communication link with the target system) rather than that processor operations has been waiting for the message too long.

WAITING FOR VM START MESSAGE
(Yellow or red) VM only: processor operations is waiting for the first VM IPL message on the operator console. ISQXLOD, ISQXIPL, or ACTIVATE command processing has completed for a VM system and processor operations is waiting for the first VM IPL message on the operator console. If this condition is displayed in red, it means that something is wrong (such as processor operations having lost its communication link with the target system) rather than that processor operations has been waiting for the message too long.

If you issue ISQXDST with the name of a target system, the Target System Summary panel is displayed (Figure 154 on page 402).
This panel has the following PF keys:

- The PF7 key displays the Processor Operations Interested Operator List panel (the one accessed with the ISQXMON command).
- The PF9 key displays the Target Hardware Summary panel. This panel provides detailed status information about the target hardware that the target system is defined on.
- The PF11 key displays the connection Path Detail panel. This panel provides detailed status information about a specific connection path.

The Target System Summary panel shows the following specific information:

**Target System Name**
The name assigned in the customization dialog to this target system.

**Target System Description**
Short textual description of this target system, defined in the customization dialog.

**Status**
(Updated dynamically) The current value of the processor operations internal variable *tstat*.

You can change this status from INITIALIZED, LOADFAILED, or IPLFAILED to IPLCOMPLETE by issuing the ISQVARS command to change the internal variable *tstat*. (For more information about specifying *tstat* as a keyword on the ISQVARS command, see the description of the ISQVARS command in *IBM Tivoli System Automation for z/OS Operator’s Commands*). You would want to do this when you perform a cold start of processor operations while a target system is already running.

After the cold start, you issue the ISQXIII command to the target system that is already running so that its status becomes INITIALIZED, then you change the value of *tstat* to IPLCOMPLETE.
You may also want to change the status manually if the cause of the LOADFAILED status was corrected by using the pass-through facility of processor operations, or if it was corrected locally at the site of the target system.

You can also issue the ISQVARS command to change the internal variable tstat from IPLCOMPLETE to IPLFAILED. Processor operations sets a status of IPLCOMPLETE when it initiates a load of a target system and receives an operating system specific message that indicates that the operating system received a level where it is ready to work. However, neither of these operating systems provide a sufficient indication that the load process failed, preventing processor operations from changing the target system status appropriately.

Valid values for this status field are as follows:

**CLOSED**
- (Green) The target system has been closed.

**INITIALIZED**
- (Green) The target system has been successfully initialized.

**IPL COMPLETE**
- (Green, yellow, or red) For a TPF or VSE system, the Load or Load Clear process started. For an z/OS system, processor operations automation responded to the IEA347A message. For a VM system, processor operations automation responded to the VM start message. If this condition is displayed in red, this is done to get your attention to inform you that something is wrong (such as processor operations lost its communication link with the target system) rather than that the IPL process failed. This condition will revert to being displayed in green when the problem is corrected.

**IPL FAILED**
- (Red) z/OS only: Error messages were generated during the initial program load process.

**LOAD FAILED**
- (Red) The Load or Load Clear operation did not complete successfully.

**NOT ACTIVE**
- (Yellow) Activate response was received from the target and did not indicate that stage-1 activation was complete. If the target is running in an LPAR, the logical partition is found not activated.

**NOT OPERATING**
- (Red) Target hardware indicates a not operating condition for the image. No CP status is available for problem determination.

**STAGE-1 ACTIVATE COMPLETE**
- (Yellow) The Support Element power-on reset has occurred but the load is not yet complete.

**UNDECIDABLE**
- (Yellow) SNMP Path Only: The target hardware is not communicating with the support element or the support element is not communicating with the HMC.
UNKNOWN
(Green) No attempt has been made to initialize or activate this target system. This status can be set only by a cold start of processor operations.

WAITING FOR IEA101A START MESSAGE
(Yellow or red) z/OS only: ACTIVATE, ISQXIPL, or ISQXLOD command processing has completed for an z/OS system and processor operations is waiting for the IEA101A message on the operator console. If this condition is displayed in red, it is to get your attention to inform you that something is wrong (such as processor operations lost its communication link with the target system) rather than that processor operations has been waiting for the message too long.

WAITING FOR IEA347A MESSAGE
(Yellow or red) z/OS only: processor operations has replied to the IEA101A message and is waiting for the IEA347A message on the operator console. If this condition is displayed in red, this is done to get your attention to inform you that something is wrong (such as processor operations lost its communication link with the target system) rather than that processor operations has been waiting for the message too long.

WAITING FOR VM START MESSAGE
(Yellow or red) VM only: processor operations is waiting for the first VM IPL message on the Support Element CI. The ACTIVATE or LOAD command processing has completed for a VM system and processor operations is waiting for the first VM IPL message. If this condition is displayed in red, it is to get your attention to inform you that something is wrong (such as processor operations lost its communication link with the target system) rather than that processor operations has been waiting for the message too long.

Target Hardware
(Updated dynamically) The current value of the processor operations internal variable thwstat. Valid values are as follows:

blank (Not highlighted) No problem has been reported for the target hardware. You can also use the ISQVARS command to change thwstat to blanks after correcting an error condition reflected in this field.

DEGRADED
(Yellow) SNMP Path Only: Target hardware runs with degraded performance (specific hardware types only).

POWERED OFF
(Red) The CPC is physically powered off.

POWERSAVE
(Red) SNMP Path Only: Power utility for the target hardware failed. Target hardware put in power save state.

SERVICE
(Red) SNMP Path Only: Service status enabled for the target hardware.

SERVICE REQUIRED
(Yellow) SNMP Path Only: The next disruption in the target hardware will result in degraded capacity or will fail.
TARGET HARDWARE PROBLEM
(Red) The Support Element of the processor (CPC) reported a target hardware problem.

Attention
(Updated dynamically) The current value of the processor operations internal variable tattn. Valid values are as follows:

- blank (Not highlighted) The target system has not encountered any of the following conditions. You can also use the ISQVARS command to change tattn to blanks after correcting an error condition reflected in this field.

ALERT
(Updated dynamically) The current value of the processor operations internal variable scstat.

- blank (Not highlighted) Either the target system has not encountered any of the following conditions, or the condition has been resolved.

- PATH PROBLEM
(Red) The focal point system cannot communicate with the Support Element on the NetView connection. View the Path Detail panel for more information (see "Display Path Details" on page 415).

ACTIVE
(Green) The path is ready for communications.
SNMP SESSION BUSY
(Red) Processor operations cannot presently communicate with the target Support Element, the problem appears to be recoverable.

SNMP SESSION PROBLEM
(Red) Processor operations cannot communicate with the target Support Element.

SUSPENDED
(Red) The connection path to the target hardware has been suspended. No ISQXIII command that is using this connection path will work until it is resumed again. For suspend/resume operation you can use the command characters of ISQXDST panel ISQESUM or the ISQXCON command. Note that this status is propagated to all affected target systems.

Lock Holder
(Updated dynamically) The current NetView operator that owns the lock for the target system and the connection path. Locks can be set using the ISQXLOC and ISQXUNL commands. Locks grant the exclusive use of the connection path of a target hardware and its associated target systems.

Last Significant Message
(Updated dynamically) The text of the last command response or report received from the target system. The date and time of the message is also displayed. The text of the message is truncated to 79 characters, including the message identifier but not including the time stamp. The message is not removed from the panel when the condition that generated the message is resolved.

Displaying Processor Resource Information (SNMP Connections)
To view target hardware summary of a processor, select PF9 from the Target System Summary panel (Figure 154 on page 402) or from the Status Summary panel (Figure 153 on page 399) on a NetView console.
This panel has the following PF keys:

- **PF2** displays the zBX Base Operation Blade View panel ISQEZBX (see “ISQEZBX Panel Example” on page 413). This function is only available if the target hardware has a zEnterprise blade extension attached and if an ensemble definition exists, with this target hardware selected as a member.

- **PF7** displays the Processor Operations Interested Operator List panel (accessed with the ISQXMON command).

- **PF11** displays the Path Detail panel (“Display Path Details” on page 415). This panel provides detailed status information about a specific NetView path. First place the cursor on the line with the name of the path that you want more information about, and then press PF11.

The Target Hardware Summary panels show the following specific information:

**Ensemble Membership**
Ensemble entry name or n/a. Shows the name of the ensemble, this target hardware is assigned as a member.

**Target Hardware Name**
The name assigned to this definition of target hardware in the customization dialog.

**Target Hardware Type**
Indicates the machine type of a System z or 390-CMOS processor.

**Target Hardware Mode**
LPAR, ESA

**Target Hardware Description**
Short textual description of this definition of target hardware, defined in the customization dialog.

**Initialized Target Systems**
(updated dynamically) List of target systems currently initialized to this target hardware. For non-LPAR-mode target hardware, this field lists one
or none. For LPAR-mode target hardware, this field can list multiple target systems. This field is blank if no target systems are initialized to this target hardware.

**Communication Path Error Status**
(updated dynamically) Status information in case of a communication problem between the NetView focal point and the target hardware.

Valid values for the status of the NetView connection path are:

**SNMP SESSION BUSY**
(red) Processor operations cannot presently communicate with the target support element, the problem appears recoverable.

**SNMP SESSION PROBLEM**
(red) Processor operations cannot communicate with the target support element.

**Last Significant Message**
The last significant message or alert associated with this target hardware.

**THW Site Information**
THW location information for this target hardware as defined in the Policy Database.

**ZBX Extensions**
YES or NO. Indicates if the target hardware has a blade extension unit attached.

**ZBX Management**
YES, NO, n/a. If the target hardware has a zBX attached and is also defined as an ensemble member, the value is YES. If the target hardware has no zBX attached, the value is n/a. If a zBX is attached, but no ensemble membership is defined, the value is NO.

**How to Monitor and Control Ensemble connections and zBX resources**

If you have defined ensemble connections together with target hardware that has a zBX unit attached, you can use the ProcOps ISQXDST status system to monitor and control the following zEnterprise blade extension resources: blade centers, blades, the zBX unit, and the connection to the ensemble HMC, which is the communication end point for the ProcOps.

For a standard ProcOps configuration, with target systems and an ensemble connection to a target hardware with a zBX attachment, use the command form: ISQXDST <ensemble_name> to display the ensemble main selection panel ISQEENS.

For a ProcOps configuration with no target system definitions, the ISQXDST command entered without a parameter is sufficient to display the ensemble main selection panel ISQEENS.
The panel has the following PF keys:

- The PF1 key displays the ISQEENS panel help.
- The PF2 key (zBXCntl) invokes the zBX Base Operation Blade View panel to manage zBX resources of the selected THW Node. Place the cursor on one of the 8 ensemble THW Node fields and press PF2. Note, that this function works only for THW node names shown in the format: <thwname>:<netid>.<nau>. For empty fields (...) or node members without a configured HTTP protocol, the zBXcntl function has no effect.
- The PF3 key returns from the ISQEENS panel to the previous panel view.
- The PF4 key schedules the ISQXIII command to initialize the current ensemble connection for execution on the associated ensemble control task. When the ISQXIII command is complete, the Conn Status field will be updated with the latest status.
- The PF5 key schedules the ISQXCLS command to close the current ensemble connection for execution on the associated ensemble control task. When the ISQXCLS command is complete, the Conn Status field will be updated with the latest status.
- The PF6 key invokes the NetView ROLL function.
- The PF10 key selects the ensemble previous to the currently selected one and refreshes the ISQEENS panel with the new selected ensemble. If the ensemble selected is the first one, pressing PF10 selects the last ensemble.
- The PF11 key selects the ensemble next to the currently selected one and refreshes the ISQEENS panel with the new selected ensemble. If the ensemble selected is the last one, pressing PF11 selects the first ensemble.
- The PF12 key terminates the ISQXDST dialog and returns to the NCCF screen.

The Ensemble View panel shows the following specific information:

**Connection configuration and status fields**

**Description**
- Shows the text you entered in the SA customization dialog as a short description of this ensemble.
Real EnsName
Is the ensemble name defined on the HMC for this ensemble. This name can be different to the ensemble entry name used in ProcOps. SA ProcOps requires the entry names for its operation. Real ensemble names can be longer than 8 character and can contain mixed case characters or blanks. Note, that the real ensemble name is required when you define an ensemble with the SA customization dialog. The real ensemble name is also a supported target parameter of the ISQECMD ensemble command interface.

Conn Status
Current status of the ensemble connection. Valid status values are:

<table>
<thead>
<tr>
<th>Status Value</th>
<th>Color</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVE</td>
<td>green</td>
<td>Ensemble connection was successfully established and is currently active.</td>
</tr>
<tr>
<td>CLOSED</td>
<td>green</td>
<td>Ensemble connection is currently closed.</td>
</tr>
<tr>
<td>CONNECTING</td>
<td>yellow</td>
<td>Ensemble connection initialization failed. It will be retried after the specified polling frequency time period.</td>
</tr>
<tr>
<td>DORMANT</td>
<td>green</td>
<td>Ensemble connection not initialized since the last ProcOps start.</td>
</tr>
<tr>
<td>SUSPENDED</td>
<td>red</td>
<td>The ensemble connection has been suspended and cannot be used until it is resumed again, using the ISQXCON command. No polling is performed while an ensemble connection is suspended.</td>
</tr>
<tr>
<td>ENS SESSION PROBLEM</td>
<td>red</td>
<td>The ensemble connection experienced a PROBLEM problem after a successful initialization. If a polling frequency was defined for the connection, ProcOps will try to re-establish the connection at every polling time.</td>
</tr>
</tbody>
</table>

HMC Login User
This is the userid defined on the zEnterprise ensemble HMC, ProcOps uses for its HMC communication. Depending on the userid's task level and resource scope, which is set on the HMC when the userid is defined, zBX resources are accessible to be managed by ProcOps.

Poll Frequency, Poll Retries (for the ensemble HMC connection)
The frequency is defined in minutes and represents the time period between two ensemble connection pollings. A polling frequency value of 0 means that no polling is performed. The frequency value for an ensemble can be set using the SA customization dialog, or at ProcOps runtime with the ISQVARS
command. The polling retry value is used by the polling routine to determine if and how often a failing ensemble communication command should be retried before giving up the polling.

**Cmd retries, cmd wait (the ensemble HMC connection)**

These are general ensemble command control values to define how often a failing ensemble command should be retried before reporting it as failed and how long to wait for a command response from the HMC before reporting a timeout condition. The command retry and command wait values for ensemble commands can be set using the SA customization dialog, or at ProcOps runtime with the ISQVARS command.

**Connection Type**

Always set to HTTP. ProcOps exploits the System z HMC Web Services API which is a HTTP based IP Stack (alternate to the default stack). If you want to use a different IP stack on the ProcOps FP system for the HTTP communication with the zEnterprise ensemble HMC, you can define this using the SA customization dialog, or at ProcOps runtime with the ISQVARS command. Note, that after you have specified an alternate IP stack at ProcOps runtime, you must recycle the HMC connection to activate this change.

**Lock Holder**

With ProcOps you can use the ISQXLOC/ISQXUNL commands to lock an ensemble connection, so that only your operator tasks can use it. Other operators or autotasks cannot use the ensemble connection while it is locked. The Lock Holder field shows the name of the operator or autotask, currently holding the lock. Note, that normally locks are set only for a short time period, to gain exclusive control for important ensemble work.

**HMC Addresses**

In a zEnterprise ensemble environment a primary and an alternate HMC must be configured. The primary HMC’s IP address or hostname is required when you define an ensemble in the SA customization dialog. If an alternate HMC address is specified, the ProcOps ensemble connection polling will PING both HMC addresses and report bad results. Note, that in case of an HMC switch (primary becomes alternate) the IP address of the ‘old’ primary HMC will remain the IP address of the ‘new’ HMC.

**THW Node information**

In a zEnterprise ensemble, up to 8 nodes can be defined as members. A node is an IBM zEnterprise (TM) System, with or without a zEnterprise BladeCenter Extension (zBX) attached. In ProcOps, a node is a target hardware (processor) with a membership to an ensemble. The processor and the ensemble membership can be defined using the SA customization dialog.

Panel ISQECENS lists the THW ensemble member nodes in two columns each with four fields. Unused member fields are marked: ‘.......’. If a node is listed in the format <thw>:<netid>.<nau>, means that a zBX is expected to be attached and that management for the zBX resources is intended. This is configured by adding ‘HTTP’ to the connection protocol list for a processor with the SA configuration dialog. If only the <thw> target hardware name is displayed, the processor target hardware has an ensemble membership defined, however the zBX support protocol ‘HTTP’ is not configured.

**How to Monitor and Control zBX resources**

The ISQEZBX panel can be used to query information of the following zBX objects:
- The blade extension unit,
The associated ensemble,
The rack mounted blade centers,
The blades mounted in the blade centers

For the mounted blades, the ACTIVATE and DEACTIVATE operations commands are available to manage them. The ISQEZBX panel has no data entry fields. It uses the cursor position on the displayed zBX screen objects to select them for the PF-key assigned functions. Especially the panel operation with a 3270 screen emulator is supported, where you can position the cursor using a pointing device. If your 3270 emulator provides a customizable on-screen keypad function, you can even have the ISQEZBX panel function keys on the keypad, which allows you to control your zBX resources just with a pointing device.

Note: The layout of the view for racks, blade centers, and blades in panel ISQEZBX does not represent the exact physical layout of a zBX and its components, only a schematic view of a zBX is displayed. The blades are not shown in a view that represents their exact physical location in a blade center.

Cursor sensitive panel areas ( A ... F )

**A Ensemble**
Field shows the ensemble name. When cursor selected, an Ensemble report will be displayed in the report window when the PF2 function is performed.

**B Racks**
The line lists the configurable racks of the zBX in the range rack B to rack D. Available, accessible rack IDs are displayed highlighted. When this line is cursor selected, a PF2 Report function will generate a zBX Unit report and display the results in the report window area.

**C Blade Centers**
The line lists all blade centers detected, when the ISQEZBX panel was invoked. Unavailable blade centers are shown with a name of 'n/a'. Blade centers available are shown with their names. Only the first 6 characters of their names are displayed in this line. If the cursor is positioned on a blade center name field, a PF2 report function will display the report for this blade center in the report window area.

**D Blades**
In this screen area, the mounted blades of a zBX are listed in columns. For each possible rack of a zBX and its two possible blade centers, two blade columns are shown as the display location for the individual blades. If the cursor position is on a mounted blade on one of the columns, the blade center
of that column and the rack of the blade center are used to locate the correct zBX blade. The PF2 report function displays the result in the report window area.

### zBX Report Output Window
This screen section is used as the output area for the zBX reports. They start all with a AOFBxxxx report id. The remaining report data is shown in a condensed form to make best use of the limited area size. If you wish to see the original zBX report, just position the cursor in the window area and perform PF2. A WINDOW panel is displayed showing the original AOFBxxxx report. If you need help information for a AOFBxxxx report id, use PF1 in the report window area.

### Any other screen area
With the cursor on any other area than A to E, performing the PF2 report function has no effects. When you use the PF1 Help function, this panel help will be displayed. Independent from the cursor location you can always perform the PF9 Show/Refresh function.

#### ISQEZBX Panel Example

```
<table>
<thead>
<tr>
<th>ISQEZBX</th>
<th>SA z/OS Proc-Ops zBX Base Operation Blade View</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ensembl</td>
<td>/ TH Node Member: HYDRA / R93</td>
</tr>
<tr>
<td>--- Rack B ---</td>
<td>--- Rack C ---</td>
</tr>
<tr>
<td>B.1</td>
<td>B.2</td>
</tr>
<tr>
<td>0 0 0 0 0</td>
<td>0 0 0 0 0</td>
</tr>
<tr>
<td>=p= 0 0 0</td>
<td>=p= 0 0 0</td>
</tr>
<tr>
<td>=p= 0 0 0</td>
<td>=p= 0 0 0</td>
</tr>
<tr>
<td>=x= 0 0 0</td>
<td>=x= 0 0 0</td>
</tr>
<tr>
<td>=x= 0 0 0</td>
<td>=x= 0 0 0</td>
</tr>
<tr>
<td>=p= 0 0 0</td>
<td>=p= 0 0 0</td>
</tr>
<tr>
<td>=x= 0 0 0</td>
<td>=x= 0 0 0</td>
</tr>
<tr>
<td>=x= 0 0 0</td>
<td>=x= 0 0 0</td>
</tr>
<tr>
<td>=x= 0 0 0</td>
<td>=p= 0 0 0</td>
</tr>
</tbody>
</table>

AOFB0011 ZBXDATA HYDRA STATUS(SUCCESS) TSTIME(120303085035) HYDRA CPC(R93)
ZBX NAME(2458-001-000000033333) MTYPE(2458) MMODEL(001)
MSERIAL(000000033333) DESC(Represents one zBX) CDPX150E(6) CISAOPTE(3)
CPowerE(29) CXE(35) MAXDPX150E(112) MAXISAOPTE(112) MAXPOWERE(112) MAXXE(112)
PF1=Hlp PF2=Report PF3=Exit PF4=Act PF5=Deact PF6=Roll PF9=Show/Refr PF12=Quit
```

**Figure 158. zBX Base Operation Blade View**

In the above ISQEZBX panel snapshot the cursor was positioned on the Rack header line and PF2 key was pressed. As a result a zEnterprise Blade Extension unit report was returned in the ISQEZBX message window area.

Panel ISQEZBX provides the following PF key functions:

- The PF1 key invokes the AOFBxxxx report message specific help function, depending on the cursor position, either the general panel help, or if the cursor position is on the report window.
- The PF2 key executes the specific query command function for this object, if the cursor is positioned on a zBX object line or column. The result is displayed in a special report window at the lower part of the panel. The following ensemble commands are used: ZBXDATA, ENSDATA, BCDATA, and BLDATA. The reports are presented in a condensed form, however you can use the PF2
function again with the cursor positioned in the report window area to display a new panel with the original report messages. With BCDATA and BLDATA processing, the panel status of the selected blade or blade center object will automatically change to the new status shown in the report if there was a status change. Note, that when you place the cursor on an empty (...) blade column position, an empty report window is shown.

- The PF3 key returns from this panel to the next higher level in the ISQXDST panel hierarchy.
- The PF4 key ACTIVATEs a blade, before the function starts, you must confirm it (PF10) to proceed or cancel (PF11) the request. This function activates the selected blade unit which is located in a Blade Center Chassis of a zBX rack. After a successful activation, the blade status of a BLDATA report (PF2) shows 'operating'. Note, that the zBX VH (virtual host) related to this blade will be automatically activated. The VS (virtual server) objects related to the VH must be extra configured for automatic activation. VSs not configured for automatic activation have a status of 'not-operating' after a successful blade activation. Activating a blade while the blade is still active and work is still running on the virtual servers of that blade, will interrupt this work because the blade is deactivated before it is activated again. The PF4 blade activation calls the ISQROUTE ProcOps function to schedule an ISQECMD with the blade activate ensemble command. This allows the user to perform other ISQEZXBX functions immediately after the blade activation has been scheduled. You must use either PF9 or PF2 with the cursor on the blade position to get any blade status change information displayed on the panel.
- The PF5 key DEACTIVATEs a blade, before the function starts, you must confirm it (PF10) to proceed or cancel (PF11) the request. This function deactivates the selected blade unit which is located in a Blade Center Chassis of a zBX rack. After a successful deactivation, the blade status of a BLDATA report (PF2) shows 'no-power'. Note, that the zBX VH (virtual host) related to this blade will be automatically deactivated. The VS (virtual server) objects related to the VH will automatically be deactivated and any work running is disrupted. After successful deactivation of a blade, the related VS objects have a status of 'not-operating'. The PF5 blade deactivation calls the ISQROUTE ProcOps function to schedule an ISQECMD with the blade deactivate ensemble command. This allows the user to perform other ISQEZXBX functions immediately after the blade activation has been scheduled. You must use either PF9 or PF2 with the cursor on the blade position to get any blade status change information displayed on the panel.
- The PF6 key invokes the NetView ROLL function
- The PF9 key invokes the zBX Show/Refresh function. When panel ISQEZXBX is initially displayed in a panel session, the list of the mounted blades on the zBX is empty. The initial panel display shows the available racks highlighted with their available blade center names displayed. The PF9 function must be performed first to fill the blades columns with the available mounted blade information. The PF9 function can also be used to refresh all blade columns.
- The PF12 key (Quit) returns from ISQEZXBX panel and exit ISQXDST. This returns you to the NCCF screen.

Acceptable status properties for blades

With the exception of racks and workloads, the possible status values of other zBX objects can be configured to be an 'ACCEPTED' status. zBX objects with this status attribute set can be considered as not requiring special attention, regardless of their actual status. In fact, the HMC user interface marks all zBX objects in an acceptable status with a green status indicator. You can use the HMC Ensemble Management
task and set/unset the accepted status marker for a status value using the zBX object's Details view. Press the apply button to activate the changes.

The ISQEZBX panel displays all blades with the accepted status property set in green. The accepted status properties for blades are also shown in the AOFB0016 BLDATA reports. Acceptable status properties for other zBX objects than blades are currently ignored by the ISQEZBX panel display, but can be determined in the reports.

**Blade type and status indicators**

The following blade status indicators are shown in the rows/columns reserved for the zBX blades:

- Indicates an empty place, no blade is mounted:
  
  ... (color blue)

- Indicators for a mounted blade of type 'system-x':
  
  =x= (color green, status is accepted)
  -x- (color yellow reverse, status is transient)
  _X_ (color red reverse, status is unaccepted)

- Indicators for a mounted blade of type 'power':
  
  =p= (color green, status is accepted)
  -p- (color yellow reverse, status is transient)
  _P_ (color red reverse, status is unaccepted)

- Indicators for a mounted optimizer hardware component of type 'isaopt' or 'dpxi50z':
  
  =o= (color green, status is accepted)
  -o- (color yellow reverse, status is transient)
  _O_ (color red reverse, status is unaccepted)

**How to Monitor Communications Path Status**

The communications paths from the processor operations focal point system to target systems are of two types: NVC and SNMP. You can display details of the connection for these types of path.

**Display Path Details**

Press PF11 on the Target System Summary panel or Target Hardware Summary panel to display the Path Detail Panel, as shown in Figure 159 on page 416.
The Path Detail panel shows the following information:

**Target Hardware Name**
The name assigned in the customization dialog for the processor that this connection path details belong to.

**Connection Type**
Type of path assigned in the customization dialog to be used for communication between the processor operations focal point and the target hardware. Connection types can be the following:

**SNMP**
IP-based communications protocol that requires that the address information for either SE or HMC are defined. If both addresses are customized in the customization dialog, the SE address has priority.

**NVC**
SNA-based communications protocol that requires the CPC SNA address.

**LU Name**
The NAU part of the CPC SNA address, which is also the logical unit name assigned to the NetView connection path.

**Network ID**
The NETID part of the CPC SNA address, which is also the name of the SNA network ID for the NetView connection.

**Support Element IP Address or Hostname**
The name or IP address assigned in the customization dialog to the SE of this target hardware.

**Hardware Management Console IP Address or Hostname**
The name or IP address assigned in the customization dialog to the HMC of this target hardware.

**Connection Authorization**
In case of an NVC connection, the authorization token is displayed. For
SNMP connections the community value is shown. The authorization values are defined in the customization dialog for a selected target hardware.

**Lock Holder**
(Updated dynamically) The current NetView operator owning the lock for the connection path. Locks can be set and unset using the ISQXLOC and ISQXUNL command. Locks grant the exclusive use of a connection path.

**SNMPv3 Security**
SNMPv3 security status.
- **Y** The connection is SNMPv3 and authenticated by SNMPv3 User and Password.
- **N** The connection is SNMPv2 and authenticated by the community name (Connection Authorization).

**SNMPv3 User**
SNMPv3 User specified for the connection.

**Path Status**
(Updated dynamically) The status of the NetView path. Valid values are:
- **UNKNOWN** (Green) No command was ever issued to this path name.
- **ACTIVE** (Green) The path is ready for communications.
- **PATH PROBLEM** (Red) The focal point system cannot communicate with the Support Element on the NetView connection. This status is accompanied by a Last Significant Message.

**Poll Frequency**
The time (in minutes) that processor operations should poll the Support Element for status.

**Poll Retries**
(Updated dynamically) The number of attempts to retry that should occur before processor operations notifies the operator that the connection failed.

**Note:** Updates for 'Path Poll Frequency' (pollfreq) or 'Path Poll Retries' (pollrtry) have no impact on the polling task; to effectively change these values, the update must be performed in the SA z/OS customization dialog Processor Information panel (AOFGCPD0); afterwards, rebuild and reload the Automation Control File (ACF) and restart processor operations.

**Last Significant Message**
(Updated dynamically) The text of the last command response or report received from the target system. The text of the message is truncated to 79 characters, including the message identifier but not including the time stamp. The message is not removed from the panel when the condition that generated the message is resolved.
ISQXIII

Purpose

The ISQXIII command initializes (or re-initializes) a target system, or a target system connection. When initializing a target system, ISQXIII establishes a connection to the target processor and enables its console connections.

When initializing a target hardware, ISQXIII establishes a connection to the target processor and enables the console connection for all target systems defined that run on the specified target hardware.

In addition, this command can be used to establish a session to an Ensemble HMC, which is required to manage zBX Blade resources with ProcOps.

Syntax

```
ISQXIII target_system_ProcOps_name
     OC
     SC
     ensemble_ProcOps_name
     target_hardware_ProcOps_name
```

Requirements

The requirements for this command to complete successfully are:

- Processor operations must be active.
- The addressed target system must have a status of UNKNOWN or CLOSED.
- The addressed target system must be initialized if you use the `target_system_ProcOps_name console_designator` syntax.
- No associated console or ensemble connection can be locked by another task.

Parameters

- `target_system_ProcOps_name` Is the name of the target system. This is a required parameter, and must be a valid target system name.
  - OC Specifies the active operator console (not valid for the coupling facility).
  - SC Specifies the system console connection.

- `ensemble_ProcOps_name` Is the ProcOps name of a zBX Ensemble.

- `target_hardware_ProcOps_name` Is the name of the target hardware. This is a required parameter, and must be a valid target hardware name.

Defaults

None.
Usage

Issue the ISQXIII command to a target system to prepare it for processor operations monitoring and control.

Issue the ISQXIII command to a zBX ensemble HMC to prepare it for processor operations monitoring and control.

Issue the ISQXIII command to a target hardware to prepare it and all defined target systems for processor operations monitoring and control.

Restrictions and Limitations

The following restrictions or limitations apply:

- To re-initialize a target system, you must first close it with the ISQXCLS command (see “ISQXCLS” on page 392).
- To initialize a console connection, the target system must have been initialized previously.
- To re-initialize an ensemble session, you must first close it with the ISQXCLS command.

Return Codes

0  The ISQXIII command processed successfully and the processor operations environment for the specified target system or ensemble has been initialized or re-initialized.
4  During initialization of a target system or ensemble one or more of the following problems occurred:
   - No system consoles could be contacted.
   - No operator consoles could be contacted.
   - No ensemble HMC could be contacted.
8  You used syntax that is not valid, specified a target system that is not initialized, or specified an invalid ensemble name.
12  An internal error occurred or connection path status is SUSPENDED to either the processor hardware (SE/HMC) or ensemble HMC.
16  A severe error occurred.
32  The processor operations environment does not exist; the ISQXIII command was not processed.

Initializing a Target System

Initializing a target system starts communication with the target hardware and enables the console connections. It also updates the interested operator lists for consoles at the target system. The interested operator lists are updated because an operator can join an interested operator list by issuing the ISQXMON command after processor operations has been started but before the target system has been initialized.

Initializing an Ensemble HMC Connection

Initializing this connection starts by logging in at the addressed Ensemble HMC. It then starts to discover the zBX base resources such as blades, available for monitoring and control, using this ensemble connection. The defined userid for this ensemble remains connected to the HMC, until an ISQXCLS is performed or processor operations is eliminated.
Operators can use the ISQXMON command to join/leave the interested operator group of an ensemble to get notifications and reports from the ensemble and its resources.

**Initializing a Target Hardware**

Using this command form allows you to initialize target systems of a target hardware defined to ProcOps in one step. The individual ISQXIII commands for target systems are scheduled to run on the ProcOps control task of the target hardware. When a target system is already INITIALIZED while ISQXIII for a target hardware is running, it is ignored.

---

**ISQXLOC**

**Purpose**

The ISQXLOC command locks access to a ProcOps entity to serialize operations to that console with other operator tasks. Entities can be either target system names together with a console designator or ensemble names.

**Syntax**

```
-- ISQXLOC target_system_ProcOps_name OC --
--                  target_hardware_name S  --
--                                 ensemble_ProcOps_name --
```

**Requirements**

The requirements for this command to complete successfully are:
- Processor operations must be active.
- The entity must not be locked by another task.
- The `target_system_ProcOps_name` syntax requires that the addressed target system be initialized.

**Parameters**

- `target_system_ProcOps_name`
  - Is the name of the target system.
  - **OC** Specifies the active operator console (not valid for the coupling facility).
  - **SC** Specifies the active system console (not valid for the OCF-based processors).

- `target_hardware_name`
  - Is the name of the target hardware.
  - **S** Specifies the active system console.

- `ensemble_ProcOps_name`
  - Is the ProcOps name of a zBX ensemble.

**Defaults**

None.
Usage

All processor operations commands that require serialization automatically lock the entity when they start and unlock the entity when they complete.

The lock prevents processor operations from sending requests to an entity—they are rejected unless issued by the same task that issued the ISQXLOC command—until the lock is removed by the ISQXUNL command (see “ISQXUNL” on page 428).

If you are going to issue several commands to an entity, you can issue the ISQXLOC command to lock access to the entity to ensure the commands are processed without interruption by commands from other tasks. When you are finished with this sequence of commands, issue the ISQXUNL command to unlock access to the entity.

If the ISQXLOC command is issued within an automation procedure, or within a set of called automation procedures, the locked entity should be unlocked (using the ISQXUNL command) after all processing of the automation procedure is completed. It is not normally necessary to lock an entity for the processing of an automation procedure because of the serialization of the ISQEXEC command.

Any change in lock status is reflected in the status facility.

Restrictions and Limitations

The following restrictions or limitations apply to this command:

- The parameters are not checked for validity.
- Once obtained by a task, another task cannot obtain the lock until it is released by the owning task.
- An ISQXLOC command issued from within a locked sequence of commands is rejected.

Return Codes

0  The ISQXLOC command completed successfully.
4  The entity was already locked by the current task.
8  You used syntax that is not valid, specified an entity that is not valid, or in the case that the entity is a target system, it is not initialized.
12 The entity is locked by another task.
32 The processor operations environment does not exist; the ISQXLOC command was not processed.

Any return codes greater than 8 that are not listed above indicate that an error occurred in the NetView CNMLOCK service.

ISQXMON

Purpose

The ISQXMON command adds you to or deletes you from the interested operator list for a target system console or an ensemble. When added, you can monitor a console and receive all message traffic from the target system or ensemble.

The following messages are sent to the interested operator's NetView screen:

- Operating system messages from target systems (target systems only)
• Processor hardware messages (target hardware only)
• Hardware status or Ensemble object status changes
• Command completion messages from the processor hardware or ensemble
• Query command responses from the processor hardware or ensemble

For target system and target hardware entities, messages for interested operators are all prefixed with the message ID ISQ901I and are copies of the original ISQ900I messages that ProcOps provides for its internal processing and automation.

For ensembles, messages for interested operators are prefixed with the message ID ISQ801I and are copies of the original ISQ800I messages that ProcOps provide for its internal processing and automation.

Syntax

```
/SM590000/SM590000
  ISQXMON  target_system_ProcOps_name  ON/OFF
  SC/OC
  /SM630000
```

Requirements

Processor operations must be active.

Parameters

```
target_system_ProcOps_name
  Is the name of the target system you want to monitor. The name must be a valid target system name.

  If the specified target system is a logical partition, you will receive messages from other LPARs running on the same target hardware if they are defined in your active ProcOps control file. The target system must be operational and its ProcOps status must not be UNKNOWN or CLOSED.

  ON  Indicates that you are to be added to the list of interested operators.

  OFF Indicates that you are to be deleted from the list of interested operators.

  SC  Specifies the hardware connection that includes the operating system console and the hardware system console functions.

  OC  Specific operator console connection (no longer supported, message ISQ042I will be issued when specified).
```

```
ensemble_ProcOps_name
  Is the name of the ensemble you want to monitor. The name must be a valid ensemble name.

  ON  Indicates that you are to be added to the list of interested operators.

  OFF Indicates that you are to be deleted from the list of interested operators.
```
Defaults

ON.

Usage

The ISQXMON command may be issued at any time after processor operations has been started.

For target systems: Interested operator lists are maintained per console connection. There is one list for each target system. System console and operator console functions are supported. The maximum number of interested operators is 12 per connection (but only 10 can be specified through the configuration dialogs).

As an alternative to ISQXMON, clist script ISQGMSRT is available as a sample application. Member INGEI002 in your SINGSAMP library contains the REXX source code, together with the instructions to install and use the application. ISQGMSRT allows you to maintain interested operator lists for single target systems and to colorize the ISQ901I messages.

For ensembles: Interested operator lists for ensembles cannot be pre-defined using the configuration dialogs. If you are interested in messages from ensemble objects to be displayed on your screen, you must first use ISQXMON to receive them, after ProcOps is started.

Restrictions and Limitations

The following restrictions or limitations apply:

- Issuing ISQXMON ON for a target system running on a target hardware in LPAR mode has the effect that you will receive messages from other target systems that are running on the same target hardware.
- Issuing ISQXMON OFF for a target system running on a target hardware in LPAR mode has the effect that, as long as other target systems on that hardware still have set ISQXMON ON, you will receive messages from all target systems that share the connection.
- There is a maximum of 12 interested operators per console.
- A NetView group name cannot be added or removed from a list with the ISQXMON command.
- There can be no more than 256 unique operator names in all interested operator lists.
- ISQ901I messages sent to the interested operators are suppressed in the NetView log by ProcOps.
- ISQ801I messages sent to the interested operators are suppressed in the NetView log by ProcOps.

Return Codes

0  The ISQXMON command processed successfully. The operator who issued the ISQXMON command has been added to or deleted from the interested operators list of the specified console, as requested.
4  You specified incorrect command syntax.
6  The interested operator list is full.
8  You specified a NetView group name for the list to which you want to be added as an interested operator.
16  An internal error occurred.
The processor operations environment does not exist; the ISQXMON command was not processed.

ISQXOPT

Purpose

The ISQXOPT command displays target system information. Some information can be changed at this panel.

Syntax

```
ISQXOPT target_system_ProcOps_name
```

Requirements

Processor operations must be active.

Parameters

- `target_system_ProcOps_name`
  - Is the name of the target system. It must be a valid target system name of 8 characters or less.

Defaults

None.

Usage

Issue `ISQXOPT target_system_ProcOps_name` to display a full-screen panel that shows the values of the variables that define the operation of a specific target system, such as initial program load parameters and the responses to IPL messages. The following fields cannot be modified:

- Hardware mode
- Partition
- Target mode
- Operating system
- Hardware type
- Automatic response selection

Examples of this panel are shown in the following:

- [Figure 160 on page 425](#) in basic-mode using an OCF-based processor
- [Figure 161 on page 425](#) in LPAR mode using an OCF-based processor
Issuing this command for a non-existent target system results in a panel for a Basic-mode system with a message that the specified target system does not exist. You can recover from this condition by simply typing in the name of the desired target system in the first field of the panel and pressing the Enter key. You can also type over the name of a valid target system on a valid panel and press the Enter key to change the display to that for a different target system.

Some variables can be changed by typing a new value over the old value and pressing Enter. When the variable is updated, the ISQXOPT panel is displayed again and another variable can be updated.
The information contained in the panel is retrieved, and the NetView View command presents the panel. The information you enter is checked and error messages are generated if the information is not valid. These error messages, which identify the field with the problem, are displayed on a separate panel that shows all error messages for all fields.

Restrictions and Limitations
- You cannot issue the ISQXOPT command from within a program.
- You must issue the ISQXOPT command from an NCCF command line.

Return Codes
Return codes are not applicable because you cannot issue this command from within a program.

Purpose
The ISQXPSM command starts and stops the ProcOps Service Machine (PSM) or returns its VM CP status from the system where it is running. In addition, PSM commands can be forwarded to the PSM for execution with this command.

Syntax
```
ISQXPSM target_hardware_name START
    STOP
    STATUS
    CMD( PSM command )
```

PSM command:
```
- ISQACT MH
- ISQCLEAR M
- ISQUERY Q
- ISQSTATUS
- ISQTRACE MSERV ON
    CSERV OFF
    MH
    LOGGER
    CNSERVER
    TCPIP
```

Requirements
The requirements for this command to complete successfully are:
- Processor operations must be active.
- To issue a START, STOP, or STATUS command:
  1. The hosting VM system and the PSM name must be defined in ProcOps.
  2. The VM host system of the PSM, associated with the specified `target_hardware_name`, must have been initialized with the ProcOps ISQXIII command.
3. A VM user, having the authority to issue XAUTOLOG, FORCE, and QUERY commands, must be logged on to the console via SE/HMC.

- To issue a CMD command, the PSM must be active.

**Parameters**

*target_hardware_name*

The name of a processor of type PSM.

**START**

Starts the PSM virtual machine that is specified in `target_hardware_name`.

**STOP**

Stops the PSM virtual machine that is specified in `target_hardware_name`.

**STATUS**

Returns the VM CP status for the PSM virtual machine that is specified in `target_hardware_name`.

**CMD**

Forwards a PSM command to the ProcOps Service Machine of the specified target hardware for execution and returns the result.

**PSM command**

One of the PSM diagnosis and service commands that is listed in the syntax fragment, see Chapter 8, “PSM Commands - Special Requests,” on page 533 for more details.

**Defaults**

None.

**Usage**

See Chapter 8, “PSM Commands - Special Requests,” on page 533 for information about the PSM commands that are available.

**Restrictions and Limitations**

Only a subset of the available PSM commands can be used with the CMD option of ISQXPSM. To use the full set of commands, a VM user must be logged on as PSM and enter the commands at the PSM console.

The following commands can only be entered from the PSM console directly:

- ISQMSG
- ISQPSM
- STOPALL

**Return Codes**

0   The ISQXPSM command completed successfully.
4   One of the following occurred:
  - No VM host system has been defined for this PSM.
  - The PSM's VM host target system has not initialized.
  - The PSM has not been defined for the specified target hardware.
  - The specified target hardware type is not PSM.
  - There is an error in the parameter list.
6   Authorization error. Either the user ID that was checked to access the specified target hardware has not been authorized, or the allowed access level is not sufficient.
One of the following occurred:
• ISQSEND returned an error while processing a START, STOP, or STATUS request.
• An error occurred while waiting for the response of a START, STOP, or STATUS request.

The ProcOps environment does not exist; the ISQPSM command was not processed.

### ISQXUNL

**Purpose**

The ISQXUNL command cancels a ProcOps entity lock instituted by the ISQXLOC command. Entities can be either target system names together with a console designator or ensemble names.

**Syntax**

```
ISQXUNL

-target_system_ProcOps_name

-target_hardware_name S

-ensemble_ProcOps_name
```

**Requirements**

The requirements for this command to complete successfully are:
• Processor operations must be active.
• The addressed entity must not be locked by another task.
• The `target_system_ProcOps_name` entity requires that the addressed target system be initialized.

**Parameters**

- **target_system_ProcOps_name**
  - Is the name of the target system for the specified console.
  - **OC** Specifies the active operator console (not valid for the coupling facility).
  - **SC** Specifies the active system console (not valid for OCF-based processors).

- **target_hardware_name**
  - Is the name of the target hardware.
  - **S** Specifies the active system console.

- **ensemble_ProcOps_name**
  - Is the name of the ProcOps name of a zBX Ensemble.

**Defaults**

None.

**Usage**

The ISQXUNL command removes a entity lock established by the ISQXLOC command (see “ISQXLOC” on page 420). Any change in lock status is reflected in the status facility.
Restrictions and Limitations

The ISQXUNL command is effective only when issued from the same task that
issued ISQXLOC to establish the entity lock.

Return Codes

0  The ISQXUNL command completed successfully.
4  The specified entity was not locked.
8  One of the following was specified:
   • Invalid syntax
   • Invalid entity name
   • Target system entity that is not initialized
12  The console is locked by another task.
32  The processor operations environment does not exist; the ISQXUNL command
    was not processed.

For return codes greater than 12 that are not listed, a NetView CNMLOCK service
problem occurred.
Chapter 6. Common Commands

Common commands are preceded by the ISQCCMD command (described in detail on page "ISQCCMD" on page 355). Use common commands in APIs whenever possible because they provide a single product image across various hardware and software implementations. Regardless of the processor type or the operating system running at your target system, the common command is the same. This can potentially minimize the need for future modifications to your automation routines should you modify or upgrade your processor hardware or operating system type.

ACTIVATE

Purpose

The ACTIVATE command causes the target system to perform a complete initialization. This initialization extends from a power-on reset to performing the initial program load process.

Syntax

```
ISQCCMD target_system_ProcOps_name ACTIVATE
  target_hardware_name.LPAR_name
  group_name
  subgroup_name
  target_hardware_name
```

```
FORCE(NO)
```

```
FORCE(NO)
```

```
FORCE(NO)
```

```
FORCE(NO)
```

```
CNAME(profile_name)
```

Requirements

The requirements for the ACTIVATE command to complete successfully are:

- Processor operations must be active.
- The addressed target system must have a status of UNKNOWN, CLOSED, or INITIALIZED for the default setting of FORCE(NO) to work. See Examples for a list of status settings.

Parameters

For a definition of target_system_ProcOps_name, target_hardware_name.LPAR_name, group_name, subgroup_name, and target_hardware_name see "ISQCCMD" on page 355.

FORCE

Requests conditional processing of commands that are disruptive to the operating system control program and application work that is in progress. Use of this operand is based on the operating state of the target processor.

NO Specifies that processing of the command is to continue only if the target
ACTIVATE

CPU control program is not in the operating state. Specify FORCE(NO) in a situation where you would not want to disrupt the control program work-in-progress.

YES
Specifies that processing of the command is to continue even if the target CPU control program is in the operating state. Specify FORCE(YES) in a situation where disruption of the control program work-in-progress is not important, such as in recovery situations.

FORCE(NO) requires that the associated command fails when useful work is taking place. For ACTIVATE, fail states occur if an operating mode is detected and FORCE(NO) is specified, or when FORCE defaults to NO. The command fails whenever the processor is in those states.

CNAME
Specifies the name of an image or reset profile to be sent to the processor's support element.

For image activation, it is recommended to let processor operations select the profile name automatically rather than overriding processor operations' selection with the profile_name parameter. If specified, the operand must match the target_hardware_name.LPAR_name.

For CPC activation, as indicated by target_hardware_name, SA z/OS uses the reset profile in the ACTIVATE operations management command that is sent to the support element. If this keyword is not used, SA z/OS selects the activation reset profile that is specified in the configuration dialog for target_hardware_name. If an activation reset profile was not specified in the configuration dialog, SA z/OS selects the default activation reset profile that is specified with the CCNTL common command or on the support element directly.

profile_name
Indicates the name of the image or reset profile. It is recommended that you let processor operations select the profile name automatically rather than overriding processor operations' selection with the profile_name parameter.

Usage

Caution: The ACTIVATE command can be very disruptive when you specify FORCE(YES), so use this command carefully. The SA z/OS processor operations status facility only knows about status changes that are performed from within SA z/OS. If you load and start a target system's operating system outside of SA z/OS (such as from the SA z/OS processor operations passthru facility or from the console at the remote target system) rather than by using the SA z/OS commands, SA z/OS does not know that the status of the target system has changed.

The effect of the ACTIVATE FORCE keyword applies only to target systems that are defined to SA z/OS. Because SA z/OS does not know that the target system was activated from outside of SA z/OS, it considers the target system status to be UNKNOWN. The functions performed by the ACTIVATE common command will disrupt the operating system even if you specify the FORCE(0) option, because the SA z/OS status facility does not indicate that the target system is already initialized and running.

If you do not specify FORCE(YES):
ACTIVATE

• The ACTIVATE command checks whether the same logical partition name is in use by another target system. If so, the ACTIVATE command is rejected; if not, processing continues.

• The ACTIVATE command also checks target hardware that is not logically partitioned to see if the target hardware is in use by another target system. If so, the ACTIVATE command is rejected; if not, processing continues.

ACTIVATE processing can also produce the result that more target systems are affected than the one explicitly specified in the command or those included in a target group or subgroup specified in the command. The additional target systems are CPC images on a processor in LPAR mode. These additional target systems are activated when a Reset or Image profile specifies additional profile names in the AUTOACT operand. To avoid unintended activations, operators should be kept aware of the linkages between profiles.

Using either scheduled requests or auto-activated logical partitions causes SA z/OS to receive "unsolicited" responses from one or more target systems. SA z/OS uses these responses to update its status information and to attempt to initialize the corresponding target systems.

When you specify FORCE(YES):

• The ACTIVATE command overrules the requirement that a target system must have a status of UNKNOWN, CLOSED, or INITIALIZED.

• Processing of the ACTIVATE command is performed even if the target system has a status condition of IPL COMPLETE, IPL FAILED, LOAD FAILED, or one of the WAITING FOR...MESSAGE conditions.

The ACTIVATE command performs only the processing that is required. For example, it performs a power-on reset only when required, and it does not initialize a target system that is already initialized correctly.

Remember: A real THW can only do one ACTIVATE at any time. On a VM system, no such restriction applies and no serialization is done.

Restrictions and Limitations

The ACTIVATE command does not perform resource requirement checks.

The default form of the ACTIVATE command is rejected if the addressed target system has a status other than UNKNOWN, CLOSED, or INITIALIZED.

Return Codes

The ISQCCMD command (see “ISQCCMD” on page 355) generates the return codes for common command processing.

ACTIVATE of VM Guest Systems

Purpose

If a target system running on a guest machine is part of an ACTIVATE request, this results in an XAUTOLOG command being issued to the VM system. This starts the guest machine using the profile defined for it in VM. This profile usually causes a guest operating system (MVS, LINUX, VSE, VM) to be loaded, but it is possible that no IPL command is executed. This corresponds to the use of a hardware
profile where no automatic IPL is done. For an MVS target system the RUN state of the VM guest machine is set to ON.

Parameters

FORCE(YES)
A CP FORCE command is issued to the VM system to stop the guest, if the guest is currently active. Then CP XAUTOLOG is issued to start the guest again.

FORCE(NO)
Guest status is checked. If the guest is known to be NOT OPERATING then a CP FORCE command is issued to stop the guest.

CP XAUTOLOG is issued. This means if the guest is already started (logged on) and known to be processing normally, the ACTIVATE command is rejected.

Restrictions and Limitations

The CNAME parameter is not supported.

CBOU

Purpose

The CBOU command causes the target hardware processor to perform the specified capacity backup (CBOU) function if available.

With the CBU command you can temporarily increase the central processor (CP) capacity of your central processor complex (CPC). In case of an outage, which may have reduced your overall processor capacity, additional workload can be processed on that CPC.

Syntax

```
/SM590000/SM590000
ISQCCMD target_system_ProcOps_name CBOU
   target_hardware_name.LPAR_name
   target_hardware_name
   CBOU
/SM590000/SM590000
```

```
   CMD(STATUS)
   CMD(UNDO)
   CMD(ACTIVATE)
   PW(password)
   CMD(TESTACT)
```

Requirements

The requirements for the CBU command to complete successfully are:

- Processor operations must be active.
- The addressed target system must be initialized, or, if the target hardware is addressed, at least one target system on that hardware must be initialized.
- For UNDO, ACTIVATE and TESTACT, the SE of the target hardware must have Capacity backup set to 'Enabled' in either the Customize Console Services window (Console Workplace 2.9 and later versions) or the Enable Support
Element Console Services window (Console Workplace 2.8 and previous versions). This applies to ACTIVATE and TESTACT regardless of whether the CBU activation password is provided or not.

With Console Workplace 2.10 and later versions, for ACTIVATE and TESTACT, the CBU records must be loaded on the SE of the target hardware. Multiple CBU records are supported.

- With Console Workplace 2.10 and later versions, for ACTIVATE, TESTACT and UNDO, the flag ‘Allow capacity change API requests’ must be set in ‘Customize API Settings’ on the SE.

**Parameters**

For a definition of `target_system_ProcOps_name, target_hardware_name.LPAR_name, and target_hardware_name`, see [“ISQCCMD” on page 355](#). 

**CBU**

Indicates the CBU command function to be executed.

**CMD**

The command to be executed

**STATUS**

Returns the current status of the CBU facility of the addressed CPC. The possible status values returned are:

**AVAILABLE**

CBU is installed on the CPC.

**ACTIVATED**

CBU is started through ACTIVATE command.

**NOT_ACTIVATED**

CBU is installed but not activated.

**NOT_ENABLED**

CBU is installed but not enabled.

**NOT_INSTALLED**

CBU is not installed on the CPC.

**UNAVAILABLE**

CBU is installed but no longer available.

In addition, if the CBU Status is ACTIVATED, the activation and expiration dates are returned. Note that for System z9® and older hardware, the expiration date corresponds to the activation period expiration date. For System z10™ hardware and later, the expiration date corresponds to the capacity record expiration date.

The ISQCCMD response message ISQ464I contains the status report.

**ACTIVATE**

Activates the previously installed and enabled CBU.

For **Console Workplace 2.10 and later versions**: Multiple CBU records can exist on the SE. In this case, the decision logic is as follows:

- If one of the CBU records is marked as Default, CBU Activate works with this record.
- If none of the CBU records is marked as Default, CBU Activate works with the ‘oldest’ CBU record.
This Logic applies for TESTACT as well. Note that with the CBU common command, it is NOT possible to select a specific CBU record.

**TESTACT**
Activates the previously installed and enabled CBU in test mode.

**UNDO**
Deactivates a previously activated CBU to the regular configuration. Note that this function is not available on all CPCs supporting CBU. If the UNDO command fails, the regular CPC configuration must be restored by performing a POWERON-RESET of the CPC.

**For Console Workplace 2.10 and later versions:** Multiple CBU records can exist on the SE. In this case, the decision logic for the UNDO command is as follows:

- If one of the CBU records is marked as Default, CBU Undo works with this record.
- If none of the CBU records is marked as Default, CBU Undo works with the 'oldest' CBU record.

**PW**
The CBU activation password for ACTIVATE or TESTACT. If provided, this password is used for CBU activation instead of calling the Remote Service Facility (RSF).

**For Console Workplace 2.10 and later versions:** The CBU activation password is no longer supported. If provided, it is ignored.

**Defaults**
The default of the CBU command is CMD(STATUS).

**Usage**
Issue the CBU command to temporarily increase the central processor (CP) capacity of your central processor complex (CPC). The increased capacity may require to update your CPC’s activation profiles, so that the additional processor resources can be used in your processor configuration.

Note that the term *Console Workplace* refers to the title line, that is displayed in the Main Application Window of each CPC Support Element. The following lists the SE Workplace Version numbers and their related IBM System names:

- 2.11: IBM System z196
- 2.10: IBM System z10
- 2.9: IBM System z9
- 2.8: IBM eServer™ zSeries 990

For more information about CBU, see *System z Capacity on Demand User’s Guide* or *Capacity Backup User’s Guide*, available for the particular System z on IBM Resource Link®.

**Restrictions and Limitations**
The CBU common command is only supported for SNMP connections. When issuing a CBU request for another ProcOps connection type, it will be rejected.
Return Codes

The ISQCCMD command (see "ISQCCMD" on page 355) generates the return codes for common command processing.

CCNTL

Purpose

The CCNTL common command allows you to:

- Assign a RESET activation profile to be used as the default for the addressed CPC.
- List the current default assignment of the CPC.

Syntax

```
ISQCCMD
   target_system_ProcOps_name
   target_hardware_name.LPAR_name
   target_hardware_name

APNAME (aproc_name)
```

Requirements

The requirements for the CCNTL command to complete successfully are:

- Processor operations must be active.
- The addressed target system must be initialized, or, if the target hardware is addressed, at least one target system on that hardware must have the ProcOps status INITIALIZED.

Parameters

For a definition of `target_system_ProcOps_name`, `target_hardware_name.LPAR_name`, and `target_hardware_name`, see "ISQCCMD" on page 355.

**APNAME**

If specified, requests a change of the current default activation profile name.

`aproc_name`

Activation profile name with a length of 1-16 characters. Specifying a null value is tolerated, but the current default activation profile name is not changed.

Defaults

Lists the currently assigned default activation profile name of the CPC.

Usage

Use this command to control the default activation profile assignments of a CPC. You can prepare CPC configuration changes, such as changes in the IOCDS, in a separate RESET activation profile, then use this profile for the next CPC activation, and if successful use CCNTL to set it as the new default activation profile.
Note that in the SA z/OS processor configuration dialog, you can set the name of a RESET activation profile as the default for a target hardware activation. This is a ProcOps-specific default that is used with the ISQCCMD common command interface.

The RESET activation profile name that can be changed using CCNTL is the default name that is provided by the processor hardware. This default applies for activation from the Support Element or the HMC operator interfaces.

The CCNTL response is returned in an AOFA0006 multi-line report. For more information about this report format see the appendix "Response Messages, Error Strings, Condition Codes" in IBM Tivoli System Automation for z/OS Messages and Codes.

The command response reports from the processor hardware are available as ProcOps ISQ900I messages with a console indicator of SC. Note that the message originator following the message identifier is either a ProcOps target system name, or a target hardware name together with an LPAR name. In order to get a copy of such hardware messages on your screen as ISQ901I messages, use the ISQXMON command.

This common command can also be used for processors or systems that you have configured with a BCP Internal Interface connection in your SA z/OS policy. If ProcOps is active, and your common command targets a processor or system that has the same name in your ProcOps and BCP Internal Interface configuration, ProcOps is always preferred. To avoid this, you must either stop ProcOps or use names that are uniquely assigned to either ProcOps or the BCP Internal Interface.

**Additional Programming Information**

On command completion, the immediate HW response report messages are also available in a PIPE KEEP with the name 'ISQ.SNMP'. You can directly access this HW data when using this KEEP as your first Pipe stage. See "Using Immediate ISQCCMD Common Command Responses" on page 357.

**Restrictions and Limitations**

The CCNTL common command is only supported for SNMP connections and BCP internal interface connections configured for LPAR management. When issuing a CCNTL request for another ProcOps connection type, it will be rejected.

Only the length of the specified profile name is verified. No other activation profile validations are performed.

**Return Codes**

The ISQCCMD command (see "ISQCCMD" on page 355) generates the return codes for common command processing.

**CONDATA**

**Purpose**

The CONDATA command collects basic information about the console of the addressed target hardware and presents this information in a multi-line report.
CONDATA

The term console designates either a Support Element (SE) or Hardware Management Console (HMC), whatever console type is in use for the connection.

Syntax

```
ISQCMD target_system_ProcOps_name CONDATA
```

```
target_hardware_name.LPAR_name
```

```
target_hardware_name
```

Requirements

The requirements for the CONDATA command to complete successfully are:

**Processor Operations environment**

- Processor operations must be active.
- The addressed target system must be initialized, or, if the target hardware is addressed, at least one target system on that hardware must be initialized.

**LPAR Management environment**

- SA z/OS must be fully initialized.
- The processor hardware of the addressed target system or target hardware LPAR must be configured for connection type INTERNAL in the SA z/OS customization dialogs.

Parameters

For a definition of target_system_ProcOps_name, target_hardware_name.LPAR_name, and target_hardware_name, see "ISQCMD" on page 355.

Defaults

None.

Usage

Use this command to get basic information about a console, such as console name and type, console application version number, or the configured IP addresses of this console. In addition, the HW EC Level of the console is shown and if an alternate console will take over automatically in case the primary fails.

The data is returned in an AOFA0022 multi-line report. For more information about this report format see the appendix "Response Messages, Error Strings, Condition Codes" in *IBM Tivoli System Automation for z/OS Messages and Codes*.

The command response reports from the processor hardware are available as ProcOps ISQ900I messages with a console indicator of SC. Note that the message originator following the message identifier is either a ProcOps target system name, or a target hardware name together with an LPAR name. In order to get a copy of such hardware messages on your screen as ISQ901I messages, use the ISQXMON command.

This common command can also be used for processors or systems that you have configured with a BCP Internal Interface connection in your SA z/OS policy. If ProcOps is active, and your common command targets a processor or system that...
CONDATA

has the same name in your ProcOps and BCP Internal Interface configuration, ProcOps is always preferred. To avoid this, you must either stop ProcOps or use names that are uniquely assigned to either ProcOps or the BCP Internal Interface.

Additional Programming Information

On command completion, the immediate HW response report messages are also available in a PIPE KEEP with the name 'ISQ.SNMP'. You can directly access this HW data when using this KEEP as your first Pipe stage. See “Using Immediate ISQCCMD Common Command Responses” on page 357.

Restrictions and Limitations

The CONDATA common command is only supported for SNMP connections and BCP Internal Interface connections configured for LPAR Management. When issuing a CONDATA request for another ProcOps connection type, it will be rejected.

Depending on the console version, not all basic console information is available such as EC Level, console IP address or autoswitch setting. The autoswitch setting applies only to Support Element consoles.

Return Codes

The ISQCCMD command (see “ISQCCMD” on page 355) generates the return codes for common command processing.

CPCDATA

Purpose

The CPCDATA command collects configuration and status information about the addressed target hardware and presents this information in a multi-line report.

The report includes CPC and IMAGE (logical partition) information if the processor is running in LPAR mode.

Syntax

```
ISQCCMD [target_system_ProcOps_name] [target_hardware_name.LPAR_name] [target_hardware_name]
```

Requirements

The requirements for the CPCDATA command to complete successfully are:

Processor Operations environment

- Processor operations must be active.
- The addressed target system or target hardware.lpar must have been initialized using the ISQXIII command.

LPAR Management environment

- SA z/OS must be fully initialized.
CPCDATA

- The processor hardware of the addressed target system or target hardware LPAR must be configured for connection type INTERNAL in the SA z/OS customization dialogs.

Parameters

For a definition of target_system_ProcOps_name, target_hardware_name.LPAR_name, and target_hardware_name, see “ISQCCMD” on page 355.

Defaults

None.

Usage

Use this command to get a snapshot of the current CPC and LPAR configuration. The data is returned in a AOFA0016 multiline report. For more information about this report format see the appendix "Response Messages, Error Strings, Condition Codes" in IBM Tivoli System Automation for z/OS Messages and Codes.

The command response reports from the processor hardware are available as ProcOps ISQ900I messages with a console indicator of SC. Note that the message originator following the message identifier is either a ProcOps target system name, or a target hardware name together with an LPAR name. In order to get a copy of such hardware messages on your screen as ISQ901I messages, use the ISQXMON command.

This common command can also be used for processors or systems that you have configured with a BCP Internal Interface connection in your SA z/OS policy. If ProcOps is active, and your common command targets a processor or system that has the same name in your ProcOps and BCP Internal Interface configuration, ProcOps is always preferred. To avoid this, you must either stop ProcOps or use names that are uniquely assigned to either ProcOps or the BCP Internal Interface.

Additional Programming Information

On command completion, the immediate HW response report messages are also available in a PIPE KEEP with the name 'ISQ.SNMP'. You can directly access this HW data when using this KEEP as your first Pipe stage. See “Using Immediate ISQCCMD Common Command Responses” on page 357.

Restrictions and Limitations

The CPCDATA common command is only supported for SNMP connections and BCP Internal Interface connections configured for LPAR Management. When issuing a CPCDATA request for another ProcOps connection type, it will be rejected.

Return Codes

The ISQCCMD command (see “ISQCCMD” on page 355) generates the return codes for common command processing.
The CTRLCONS common command allows you to:
- RESTART the console application of the addressed HMC or SE (for an SE, either the primary or the alternate application can be restarted)
- RESTART or SHUTDOWN the addressed HMC or SE console (for an SE, either the primary or the alternate console can be restarted or shut down)

Syntax

```plaintext
ISQCCMD__target_system_ProcOps_name__target_hardware_name.LPAR_name__target_hardware_name
CTRLCONS__/SM590000

CMD(SHUTDOWN) ITEM(CONS)
CMD(RESTART) ITEM(CONS) TYPE(PRIMARY)
ITEM(APPL) TYPE(ALTERNATE)
```

Requirements

The requirement for the CTRLCONS command to complete successfully is:
- Processor operations must be active.

Parameters

For a definition of `target_system_ProcOps_name`, `target_hardware_name.LPAR_name`, and `target_hardware_name`, see “ISQCCMD” on page 355.

**CMD**
The command to be executed. Can be either RESTART or SHUTDOWN.

**Note**: SHUTDOWN is only allowed with ITEM(CONS). If it used with ITEM(APPL), the CTRLCONS command will be rejected.

**ITEM**
The target of the command. Specify either:
- CONS for the SE or HMC console
- APPL for the SE or HMC console application

**TYPE**
The type of console or console application.

**PRIMARY**
Refers to the primary SE or HMC console or console application.

**ALTERNATE**
Refers to the secondary SE console or console application.

**Note**: ALTERNATE can only be used for SE connections; if it is used for an HMC connection, the CTRLCONS command will fail.
CTRLCONS

Defaults

The CTRLCONS command defaults to TYPE(PRIMARY).

Usage

Use this command to shut down or restart the HMC or SE Console or to restart the HMC or SE console application. On command execution, all ProcOps connections to that console or console application that are still active will be closed before executing the Shutdown or Restart itself in order to avoid uncontrolled connection loss as a consequence of the Shutdown or Restart command.

Note: If a console SHUTDOWN is performed, the console must be restarted manually.

Restrictions and Limitations

The CTRLCONS common command is only supported for SNMP connections. When issuing a CTRLCONS request for another ProcOps connection type, it will be rejected.

All required command parameters are checked if provided with the CTRLCONS command. All command parameters are checked for valid content.

Due to technical limitations on the SE console side, an SE Application Restart will actually be executed as a SE Console Restart.

Condition Codes

The following condition code is returned in the form 00B000xx, where xx is the condition code:

82 Invalid Session Type of ASYNC encountered. It must be SYNC.

Return Codes

The ISQCCMD command (see “ISQCCMD” on page 355) generates the return codes for common command processing.

DEACTIVATE

Purpose

The DEACTIVATE command causes the target system to end normal operation. It also closes the system console and operator console ports.

Syntax

```
>> ISQCCMD -target_system_ProcOps_name- DEACTIVATE
   -target_hardware_name.LPAR_name-
   -target_hardware_name-
   -group_name-
   -subgroup_name-
```
DEACTIVATE

Requirements
Processor operations must be active for this command to complete successfully.
The addressed target system must be initialized.

Parameters
For a definition of target_system_ProcOps_name, target_hardware_name, target_hardware_name.LPAR_name, group_name, and subgroup_name, see “ISQCCMD” on page 355.

FORCE
Requests conditional processing of commands that are disruptive to the operating system control program and application work that is in progress. Use of this operand is based on the operating state of the target processor.

NO Specifies that processing of the command is to continue only if the target CPU control program is not in the operating state. Specify FORCE(NO) in a situation where you would not want to upset the control program work-in-progress.

YES Specifies that processing of the command is to continue even if the target CPU control program is in the operating state. Specify FORCE(YES) in a situation where disruption of the control program work-in-progress is not important, such as in recovery situations.

The default, if FORCE is not explicitly stated, is FORCE(NO).

FORCE(NO) requires that the associated command fail when useful work is taking place. The DEACTIVATE command’s formal fail criteria when FORCE(NO) is specified or when FORCE defaults to NO is operating.

Defaults
The DEACTIVATE command defaults to FORCE(NO).

Usage
Use this common command to close a target system. Closing a target system does the following:
• Ends any active sessions with the target system
• Disables the target system
• Performs any required end-of-day activities
• Quiesces the target system

If this command is directed to a target hardware name rather than to a target system ProcOps name, the entire processor complex is brought to a power-off state.

If you do not specify FORCE(YES), the DEACTIVATE command checks to make sure it will not be disruptive by performing the following processing:
DEACTIVATE

- If the target system runs in a logical partition and the same logical partition name is initialized to another target system in the same target hardware, the DEACTIVATE command is rejected and processing of the command halts.
- On a non-partitioned target system, if the target hardware has another target system initialized to it, the DEACTIVATE command is rejected and processing of the command halts.

Restrictions and Limitations
- When closing a target system, the DEACTIVATE common command issues an MVS QUIESCE (or equivalent) command and hardware system reset at the system console port if appropriate. This processor command may be processed in the NNT-driven part of shutdown procedures, if necessary.
- The DEACTIVATE common command also performs a system reset of the target system, and disables and releases the focal-point ports.
- To reinitialize a target system that was shut down with the DEACTIVATE common command, you must issue the ACTIVATE common command (see “ACTIVATE” on page 431).

Return Codes
The ISQCCMD command (see “ISQCCMD” on page 355) generates the return codes for common command processing.

DEACTIVATE of VM Guest Systems

Purpose
If a target system running in a guest machine is part of a DEACTIVATE request, a CP FORCE command is issued by the PSM for that guest machine. The VM host system is not affected.

Parameters

FORCE(NO)
- If FORCE(NO) is specified for a guest machine, the guest status is checked. If it is NOT OPERATING then a CP FORCE command is issued to stop the guest. If the status is any other value, the CP FORCE command is not issued and the DEACTIVATE request is rejected.

FORCE(YES)
- If FORCE(YES) is specified for a guest machine, a CP FORCE command is issued for the guest. If there is an active operating system running in the guest machine, it is first stopped using the appropriate QUIESCE processing for the operating system.
  
  For LINUX systems, the use of CP FORCE command means that a shutdown signal is sent to the operating system. This, in turn, allows the LINUX system to initiate its own controlled shutdown process.

Restrictions and Limitations
DEACTIVATE is not supported for target_hardware_names. (The PSM cannot be powered off.)
EXTINT

Purpose

The EXTINT command causes the target system to perform the external interrupt function on a central processor that exists on a target hardware processor.

Syntax

```
ISQCCMD
  target_system_ProcOps_name
  target_hardware_name.LPAR_name
  group_name
  subgroup_name
  EXTINT
```

Requirements

Processor operations must be active for this command to complete successfully.

The addressed target system must be initialized for this command to complete successfully.

Parameters

For a definition of `target_system_ProcOps_name`, `target_hardware_name.LPAR_name`, `group_name`, and `subgroup_name`, see "ISQCCMD" on page 355.

Defaults

None.

Usage

Issue the EXTINT command to generate an interrupt request to the processor which will be processed at the end of the current unit of work (unless other interrupt requests are pending), causing the program status word (PSW) to be swapped with a new PSW, and initiating processing from an address in the new PSW.

Restrictions and Limitations

- This command is not supported for coupling facility target systems.

Return Codes

The ISQCCMD command (see [ISQCCMD Return Codes](#)) generates the return codes for common command processing.

EXTINT for a VM Guest System

Purpose

This causes a CP EXTERNAL KEY command to be issued for the guest machine.
GETCLUSTER

Purpose

The command returns a list of the CPC addresses that are in the scope of control in the current CPC environment. If the ProcOps SNMP connection is to a Support Element, the list represents the defined CPCs of the so called Master HMC, which has the LIC Change function enabled. If the ProcOps SNMP is to a HMC, the GETCLUSTER command returns the CPC addresses of the Defined CPCs Group of this HMC.

Use the GETCLUSTER command to get basic information about all CPCs in the neighborhood of your CPC, regardless if the processors are defined to System Automation, or not.

Syntax

```
ISQCCMD target_system_ProcOps_name target_hardware_name.LPAR_name GETCLUSTER
```

Requirements

The requirements for the GETCLUSTER command to complete successfully are:

Processor Operations environment

- Processor operations must be active.
- The addressed target system or target hardware.Lpar must have been initialized using the ISQXIII command.

LPAR Management environment

- SA z/OS must be fully initialized.
- The processor hardware of the addressed target system or target hardware LPAR must be configured for connection type INTERNAL in the SA z/OS customization dialogs.

Parameters

For a definition of `target_system_ProcOps_name`, `target_hardware_name.LPAR_name`, and `target_hardware_name`, see “ISQCCMD” on page 355.

Defaults

None.

Usage

This common command can also be used for processors or systems that you have configured with a BCP Internal Interface connection in your SA z/OS policy. If ProcOps is active, and your common command targets a processor or system that has the same name in your ProcOps and BCP Internal Interface configuration, ProcOps is always preferred. To avoid this, you must either stop ProcOps or use names that are uniquely assigned to either ProcOps or the BCP Internal Interface.
GETCLUSTER

Additional Programming Information

On command completion, the immediate HW response report messages are also available in a Pipe KEEP with the name 'ISQ.SNMP'. When using this KEEP as your first Pipe stage, you can directly access this HW data.

Restrictions and Limitations

The GETCLUSTER common command is only supported for SNMP connections and BCP Internal Interface connections configured for LPAR Management. When issuing this request for another ProcOps connection type, it will be rejected.

Return Codes

The ISQCCMD command (see ISQCCMD Return Codes) generates the return codes for common command processing.

GETIINFO

Purpose

The GETIINFO command returns a subset of the CPCDATA command response. It allows you to retrieve the status and mode of the specified target system LPAR together with the defined LPAR number. If available, OS-related information such as operating system name or version data is returned.

Use the GETIINFO command if you know the LPAR name or system name and you are interested only in HW information for this LPAR or system. For performance reasons you are not recommended to use CPCDATA in such cases.

Syntax

```
ISQCCMD target_system_ProcOps_name target_hardware_name.LPAR_name GETIINFO
```

Requirements

The requirements for the GETIINFO command to complete successfully are:

Processor Operations environment

- Processor operations must be active.
- The addressed target system or target hardware.lpar must have been initialized using the ISQXIII command.

LPAR Management environment

- SA z/OS must be fully initialized.
- The processor hardware of the addressed target system or target hardware LPAR must be configured for connection type INTERNAL in the SA z/OS customization dialogs.

Parameters

None.
GETINFO

Defaults

None.

Usage

This common command can also be used for processors or systems that you have configured with a BCP Internal Interface connection in your SA z/OS policy. If ProcOps is active, and your common command targets a processor or system that has the same name in your ProcOps and BCP Internal Interface configuration, ProcOps is always preferred. To avoid this, you must either stop ProcOps or use names that are uniquely assigned to either ProcOps or the BCP Internal Interface.

Additional Programming Information

On command completion, the immediate HW response report messages are also available in a Pipe KEEP with the name 'ISQ.SNMP'. When using this KEEP as your first Pipe stage, you can directly access this HW data.

Restrictions and Limitations

The GETINFO common command is only supported for SNMP connections and BCP Internal Interface connections configured for LPAR Management. When issuing this request for another ProcOps connection type, it will be rejected.

Return Codes

The ISQCCMD command (see ISQCCMD Return Codes) generates the return codes for common command processing.

GETISTAT

Purpose

The GETISTAT command returns a subset of the CPCDATA command response. It allows you to retrieve the current HW status of a target system LPAR.

Use the GETISTAT command if you know the LPAR name or system name and you are interested only in HW information for this LPAR or system. For performance reasons you are not recommended to use CPCDATA in such cases.

Syntax

/SM590000/SM590000
ISQCCMD
(target_system_ProcOps_name)
(target_hardware_name.LPAR_name)
GETISTAT

Requirements

The requirements for the GETISTAT command to complete successfully are:

Processor Operations environment

- Processor operations must be active.
- The addressed target system or target hardware.Lpar must have been initialized using the ISQXIII command.
GETISTAT

LPAR Management environment

- SA z/OS must be fully initialized.
- The processor hardware of the addressed target system or target hardware LPAR must be configured for connection type INTERNAL in the SA z/OS customization dialogs.

Parameters

None.

Defaults

None.

Usage

This common command can also be used for processors or systems that you have configured with a BCP Internal Interface connection in your SA z/OS policy. If ProcOps is active, and your common command targets a processor or system that has the same name in your ProcOps and BCP Internal Interface configuration, ProcOps is always preferred. To avoid this, you must either stop ProcOps or use names that are uniquely assigned to either ProcOps or the BCP Internal Interface.

Additional Programming Information

On command completion, the immediate HW response report messages are also available in a Pipe KEEP with the name 'ISQSNMP'. When using this KEEP as your first Pipe stage, you can directly access this HW data.

Restrictions and Limitations

The GETISTAT common command is only supported for SNMP connections and BCP Internal Interface connections configured for LPAR Management. When issuing this request for another ProcOps connection type, it will be rejected.

Return Codes

The ISQCCMD command (see ISQCCMD Return Codes) generates the return codes for common command processing.

GETSDGR

Purpose

The command returns the reason details available from target hardware processors that operate in DEGRADED state.

Use the GETSDGR command if you need information about the reason of the DEGRADED status.
GETSDGR

Syntax

```
ISQCCMD
  target_system_ProcOps_name
  GETSDGR
  target_hardware_name.LPAR_name
  target_hardware_name
```

Requirements

The requirements for the GETSDGR command to complete successfully are:

**Processor Operations environment**
- Processor operations must be active.
- The addressed target system or target hardware LPAR must have been initialized using the ISQXIII command.

**LPAR Management environment**
- SA z/OS must be fully initialized.
- The processor hardware of the addressed target system or target hardware LPAR must be configured for connection type INTERNAL in the SA z/OS customization dialogs.

Parameters

For a definition of `target_system_ProcOps_name`, `target_hardware_name.LPAR_name`, and `target_hardware_name`, see [“ISQCCMD” on page 355](#).

Defaults

None.

**ISQVARS Keywords**

- **NOT_DEGRADED**
  System is not in DEGRADED mode
- **MEM_REduced**
  Loss of memory
- **MEM_BUS_FAILURE**
  Loss of channels due to CPC hardware failure
- **NODE_NOT_RUNNING**
  One or more books are no longer functioning
- **RING_OPEN**
  The ring connecting the books is open
- **CBU_EXPIRATION**
  Capacity backup resources have expired
- **MRU_FAILURE**
  Modular Refrigeration Unit (cooling) problem
- **TEMPERATURE_PROBLEM**
  Processor frequency reduced due to temperature problem
- **IML_WAS_IN_DEGRADED_MODE**
  CPC was IMLed during temperature problem
GETSDGR

Usage

This common command can also be used for processors or systems that you have configured with a BCP Internal Interface connection in your SA z/OS policy. If ProcOps is active, and your common command targets a processor or system that has the same name in your ProcOps and BCP Internal Interface configuration, ProcOps is always preferred. To avoid this, you must either stop ProcOps or use names that are uniquely assigned to either ProcOps or the BCP Internal Interface.

Additional Programming Information

On command completion, the immediate HW response report messages are also available in a Pipe KEEP with the name ‘ISQ.SNMP’. When using this KEEP as your first Pipe stage, you can directly access this HW data.

Restrictions and Limitations

The GETSDGR common command is only supported for SNMP connections and BCP internal interface connections configured for LPAR management. When issuing this request for another ProcOps connection type, it will be rejected.

Not all processor types provide DEGRADED support. If you issue GETSDGR for such target hardware, the GETSDGR command will fail with a condition code of 0B100001, object not defined, which means that the HW does not provide this function.

Return Codes

The ISQCCMD command (see [ISQCCMD Return Codes]) generates the return codes for common command processing.

GETSINFO

Purpose

The GETSINFO command returns a subset of the CPCDATA command response. It allows you to retrieve the status and mode of the specified target hardware together with information about the machine type model or serial number.

Use the GETSINFO command if you know the target hardware or a target system name running on that hardware. For performance reasons you are not recommended to use CPCDATA in such cases.

Syntax

```
ISQCCMD target_system_ProcOps_name GETSINFO target_hardware_name.LPAR_name
```

Requirements

The requirements for the GETSINFO command to complete successfully are:

Processor Operations environment

- Processor operations must be active.
The addressed target system or target hardware LPAR must have been initialized using the ISQXIII command.

**LPAR Management environment**
- SA z/OS must be fully initialized.
- The processor hardware of the addressed target system or target hardware LPAR must be configured for connection type INTERNAL in the SA z/OS customization dialogs.

**Parameters**
For a definition of `target_system_ProcOps_name`, `target_hardware_name.LPAR_name`, and `target_hardware_name`, see [“ISQCCMD” on page 355](#).

**Defaults**
None.

**Usage**
This common command can also be used for processors or systems that you have configured with a BCP Internal Interface connection in your SA z/OS policy. If ProcOps is active, and your common command targets a processor or system that has the same name in your ProcOps and BCP Internal Interface configuration, ProcOps is always preferred. To avoid this, you must either stop ProcOps or use names that are uniquely assigned to either ProcOps or the BCP Internal Interface.

**Additional Programming Information**
On command completion, the immediate HW response report messages are also available in a Pipe KEEP with the name 'ISQ.SNMP'. When using this KEEP as your first Pipe stage, you can directly access this HW data.

**Restrictions and Limitations**
The GETSINFO common command is only supported for SNMP connections and BCP Internal Interface connections configured for LPAR Management. When issuing this request for another ProcOps connection type, it will be rejected.

**Return Codes**
The ISQCCMD command (see [ISQCCMD Return Codes](#)) generates the return codes for common command processing.

---

**GETSSTAT**

**Purpose**
The GETSSTAT command returns a subset of the CPCDATA command response. It allows you to retrieve the current HW status of target hardware.

Use the GETSSTAT command if you know the target hardware or a target system name running on that hardware. For performance reasons you are not recommended to use CPCDATA in such cases.
GETSSTAT

Syntax

```
>> ISQCCMD target_system_ProcOps_name GETSSTAT
  -target_hardware_name.LPAR_name
  -target_hardware_name
```

Requirements

The requirements for the GETSSTAT command to complete successfully are:

**Processor Operations environment**
- Processor operations must be active.
- The addressed target system or target hardware.lpar must have been initialized using the ISQXIII command.

**LPAR Management environment**
- SA z/OS must be fully initialized.
- The processor hardware of the addressed target system or target hardware LPAR must be configured for connection type INTERNAL in the SA z/OS customization dialogs.

Parameters

For a definition of `target_system_ProcOps_name`, `target_hardware_name.LPAR_name`, and `target_hardware_name`, see “ISQCCMD” on page 355.

Defaults

None.

Additional Programming Information

On command completion, the immediate HW response report messages are also available in a Pipe KEEP with the name 'ISQ.SNMP'. When using this KEEP as your first Pipe stage, you can directly access this HW data.

Restrictions and Limitations

The GETSSTAT common command is only supported for SNMP connections and BCP Internal Interface connections configured for LPAR Management. When issuing this request for another ProcOps connection type, it will be rejected.

Return Codes

The ISQCCMD command (see [ISQCCMD Return Codes](#)) generates the return codes for common command processing.

ICNTL

Purpose

The ICNTL common command allows you to query and change LPAR-specific values and settings. The PR/SM™ hardware component or WLM use this information to manage partition performance and to distribute shared processor resources among the images of a CPC. Changing the image control values and
settings with ICNTL allows you to influence LPAR performance and LPAR resource capacities at run time.

**Syntax**

```
ISQCCMD target_system_ProcOps_name ICNTL target_hardware_name.LPAR_name
```

```
CMD(LIST)  PT(GPP)  PT(proctp)
```

```
CMD(UPDATE) VAR(vnm) VAL(vll)
CMD(READ) VAR(vnm) PT(GPP) PT(proctp)
```

**vnm:**

```
- CLUSTER
- DEFCAP
- GRPCAP
- GRPPRF
- PWG
- PWCC
- PWI
- PWIC
- PWNN
- PWXX
- WLME
```

**proctp:**

```
- GPP
- AAP
- IFL
- ICF
- IIP
```

**Requirements**

The requirements for the ICNTL command to complete successfully are:

**Processor Operations environment**

- Processor operations must be active.
- The addressed target system or target hardware.LPAR must have been initialized using the ISQXIII command.

**LPAR Management environment**

- SA z/OS must be fully initialized.
- The processor hardware of the addressed target system or target hardware LPAR must be configured for connection type INTERNAL in the SA z/OS customization dialogs.

**Parameters**

For a definition of `target_system_ProcOps_name` and `target_hardware_name.LPAR_name`, see "ISQCCMD" on page 355.
ICNTL

CMD
The image control command that is to be executed.

LIST
Returns a list of the current image control variable settings in a AOFA0007 response report from the target hardware image.

Only variables supported by the target processor type are shown in the response report. Other read errors are marked as 'ReadErr:xxx' in the variable value field.

Refer to the appendix section "Data Exchange Services "0B100xxx" in IBM Tivoli System Automation for z/OS Messages and Codes, if you receive 'ReadErr:xxx' responses for additional information.

UPDATE
Changes the value of the specified image control variable and returns an AOFA0007 response report.

READ
Returns the current setting of the specified image control variable in an AOFA0007 response report.

VAR
Image control variable name vnm.

CLUSTR
Name of the CPC LPAR cluster that the LPAR belongs to.
This is a READONLY variable.

Value information: Cluster of one or more LPARs, running on one zSeries 900 or later CPC, that are in the same sysplex and are running z/OS 1.1 or later in z/Architecture® mode. The scope of an LPAR Cluster is currently the same as the scope of an I/O Cluster, however the term LPAR Cluster is also used in relation to WLM LPAR CPU Management and Channel Subsystem I/O Priority Queuing. LPAR clusters are used for z/OS Intelligent Resource Director (IRD) capabilities. Cluster names can be defined using HCD.

DEFCAP
Defined capacity.

Value information:
0 No defined capacity for the LPAR.
1–2147483647 Amount of defined LPAR capacity.

Note that the maximum value represents the highest value accepted by the SE/HMC, which may not represent the correct capacity limit in your environment.

The defined capacity expressed in terms of Millions of Service Units (MSUs) per hour. MSUs is a measure of processor resource consumption. The amount of MSUs that a logical partition consumes is dependent on the model, the number of logical processors that are available to the partition, and the amount of time that the logical partition is dispatched. The defined capacity value specifies how much of the logical partition capacity is to be managed by Workload Manager (WLM) for the purpose of software pricing.
GRPCAP

Defined group capacity.

**Value information:**

- **0** No defined group capacity for the LPAR.
- **1-2147483647**
  
  Amount of defined group capacity the LPAR belongs to.

  Note that the maximum value represents the highest value accepted by the SE/HMC, which may not represent the correct capacity limit in your environment.

The defined capacity expressed in terms of Millions of Service Units (MSUs) per hour. MSUs is a measure of processor resource consumption.

GRPPRF

Capacity Group Profile name where the addressed target system or logical partition is a member. Members of capacity groups share a common MSU value. This is an alternative to the DEFCAP specification, where the MSU value is partition-specific. You can change the MSU CAPACITY value of a group profile using the PROFILE common command.

**Value information:**

- **nnnnnnnn**

  The name of the group profile. The name must be alphanumeric and a maximum of 8 characters. If you specify a name that does not exist as an activation profile of type GROUP, the ICNTL command fails with a condition code of 0B100005.

Note that GRPPRF is available on System z processors or later. Reading GRPPRF on older processors will return GRPPRF without a value. Updating GRPPRF on older processors fails with a condition code of 0B100001.

PWC

Current processing weight.

This is a READONLY variable.

**Value information:**

- **0** The CPC image does not represent a logical partition, or the LPAR has only dedicated processors assigned to it.
- **1–999**

  Represents the relative amount of shared processor resources that is currently allocated to the LPAR.

PWCC

Current processing weight capped indicator.

This is a READONLY variable.

**Value information:**

- **YES**
  
  The current processing weight for the LPAR is capped. It represents the maximum share of processor resources regardless of the availability of excess processor resources.

- **NO**

  The current processing weight for the LPAR is not capped. It represents the share of resources that is guaranteed to the LPAR when all CP resources are in use. When excess processor resources are available, the LPAR can use them if necessary.
PWI
Initial processing weight.

Value information:
0  The CPC image does not represent a logical partition, or the LPAR has only dedicated processors assigned to it. This value is not valid for UPDATE.
1–999  Represents the relative amount of shared processor resources that is initially allocated to the LPAR.

Note: If the WLME flag has a value of YES, the PWI value must be in the range from PWMN to PWMX.

PWIC
Initial processing weight capped indicator.

Value information:
YES  The initial processing weight for the LPAR is capped. It represents the maximum share of processor resources regardless of the availability of excess processor resources.
NO  The initial processing weight for the LPAR is not capped. It represents the share of resources that is guaranteed to the LPAR when all CP resources are in use. When excess processor resources are available, the LPAR can use them if necessary.

PWMN
Minimum processing weight.
The guaranteed minimum share of non-dedicated processor weight, when all processing resources are in use.

Value information:
0  The CPC image does not represent a logical partition, or the LPAR has only dedicated processors assigned to it. This value is not valid for UPDATE.
1–999  A value that is less than or equal to the initial processing weight that is used to define the minimum relative amount of shared processor resources that is allocated to the LPAR.

Note: This value is only applied if the WLME flag has a value of YES.

PWMX
Maximum processing weight.
Maximum relative amount of shared processor resources for this LPAR.

Value information:
0  The CPC image does not represent a logical partition, or the LPAR has only dedicated processors assigned to it. This value is not valid for UPDATE.
1–999  A value that is greater than or equal to the initial processing weight that is used to define the maximum relative amount of shared processor resources that is allocated to the LPAR.

Note: This value is only applied if the WLME flag has a value of YES.
WLME

Workload Manager Enabled flag.

**Value information:**

**YES**

z/OS Workload Manager is used to manage shared processor resources for a logical partition, according to the settings of the minimum and maximum weight specifications.

**NO**

PR/SM is managing the processing weight. The PWMN and PWMX values do not apply.

VAL

Value specification for variable UPDATE operation.

*val*

See the value information for the [VAR parameter](#).

PT

The Processor Type parameter allows you to specify which processor type the corresponding Image Control information should be read or updated for.

proctp

The processor type:

- **GPP** General Purpose Processor
- **AAP** Application Assist Processor
- **IFL** Integrated Facility for LINUX processor
- **ICF** Internal Coupling Facility processor
- **IIP** z9 Integrated Information Processor

**Notes:**

1. The IFA processor type (Integrated Facility for Application processor) has been replaced by AAP.

2. You cannot specify special purpose processors (AAP, IFL, ICF, IIP) for the CLUSTR, DEFCAP, GRPCAP, GRPPRF or WLME command parameters.

**Defaults**

If no ICNTL parameters are specified, a CMD(LIST) is executed.

If no processor type is specified with the PT parameter, the default value of GPP (general purpose processor) is used.

**Usage**

The ICNTL common command can be used to set LPAR performance related parameters, such as an LPAR's initial processing weight, PWI, or the Defined Capacity, DEFCAP.

For example, DEFCAP can be used to automate the lowering and raising of the MSU value (defined capacity, soft-cap). This LPAR value is important when Sub-Capacity License charging for software is in place. An optimized setting of the DEFCAP value can help to control software cost.

The equivalent to the ICNTL common command can be performed on the HMC using the CPC Operational Customization tasks: Change LPAR Controls and Change LPAR Group Controls.

The command response reports from the processor hardware are available as ProcOps ISQ900I messages with a console indicator of SC. Note that the message originator following the message identifier is either a ProcOps target system name,
or a target hardware name together with an LPAR name. In order to get a copy of
such hardware messages on your screen as ISQ901I messages, use the ISQXMON
command.

This common command can also be used for processors or systems that you have
configured with a BCP Internal Interface connection in your SA z/OS policy. If
ProcOps is active, and your common command targets a processor or system that
has the same name in your ProcOps and BCP Internal Interface configuration,
ProcOps is always preferred. To avoid this, you must either stop ProcOps or use
names that are uniquely assigned to either ProcOps or the BCP Internal Interface.

Additional Programming Information

On command completion, the immediate HW response report messages are also
available in a PIPE KEEP with the name 'ISQ.SNMP'. You can directly access this
HW data when using this KEEP as your first Pipe stage. See "Using Immediate
ISQCCMD Common Command Responses" on page 357.

Restrictions and Limitations

The ICNTL common command is only supported for SNMP connections and BCP
Internal Interface connections configured for LPAR Management. If you issue an
ICNTL request for another ProcOps connection type, it will be rejected.

The ICNTL common command works only for processors running in LPAR mode.
Depending on the processor machine type, the number of ICNTL variables may
vary. Use ICNTL CMD(LIST) to determine the set of the available ICNTL variables
for a processor.

You cannot specify special purpose processors (AAP, IFL, ICF, IIP) for the CLUSTR,
DEFCAP, or WLME command parameters.

For Special Purpose Processor types (AAP, IFL, ICF, IIP), variables ‘CLUSTR’,
‘DEFCAP’ and ‘WLME’ are not supported.

For an ICF-type (Internal Coupling Facility processor) special purpose processor,
ensure that LPAR that you specify corresponds to a Coupling Facility LPAR in
order to read or update meaningful values.

Condition Codes

The following are condition codes that ICNTL might encounter. They are returned
in the form 00B000xx, where xx is the condition code.
62 No (valid) ICNTL command (LIST, UPDATE, READ) is specified in the request
or the command is incomplete. The request is rejected.
63 The variable name that is specified in the request is not supported. The request
is rejected.
65 The value operand of the request was not recognized by ICNTL. The request is
rejected. Contact IBM Support.
66 The processor type value of the request is invalid. The request is rejected.

Return Codes

The ISQCCMD command (see "ISQCCMD" on page 355) generates the return
codes for common command processing.
**Purpose**

The LOAD command performs the initial program load process for a target system processor. This command is not supported for coupling facility target systems.

**Syntax**

```
ISQCCMD(target_system_ProcOps_name, target_hardware_name.LPAR_name, group_name, subgroup_name) LOAD CLEAR(NO) CLEAR(YES)
```

**Requirements**

The requirements for this command to complete successfully are:

- Processor operations must be active.
- The addressed target system must be initialized.
- The addressed target system must have a status of at least INITIALIZED.
- The addressed target system cannot have a status of UNKNOWN, CLOSED, or IPLCOMPLETE for the default setting of FORCE(NO) to work.

**Parameters**

For a definition of `target_system_ProcOps_name, target_hardware_name.LPAR_name, group_name, and subgroup_name`, see "ISQCCMD" on page 355.

**CLEAR(NO)**

Indicates the function of clearing storage is not to be performed; the LOAD common command issues a normal LOAD command. No space is allowed between the keyword CLEAR and the left parenthesis. This is the default.

**CLEAR(YES)**

Indicates the function of clearing storage is to be performed; the LOAD common command issues a LOADCLEAR command. No space is allowed between the keyword CLEAR and the left parenthesis.

**FORCE(NO)**

Requests conditional processing of commands that are disruptive to the operating system control program and application work that is in progress. Use of this operand is based on the operating state of the target processor.

**NO**

Specifies that processing of the command is to continue only if the target CPU control program is not in the operating state. Specify FORCE(NO) in a situation where you would not want to upset the control program work-in-progress.
LOAD

YES
  Specifies that processing of the command is to continue even if the target
  CPU control program is in the operating state. Specify FORCE(YES) in a
  situation where disruption of the control program work-in-progress is not
  important, such as in recovery situations.

  The default, if FORCE is not explicitly stated, is FORCE(NO).

  FORCE(NO) requires that the associated command fails when useful work is
  taking place. For this command the formal fail criteria when FORCE(NO) is
  specified or when FORCE defaults to NO is the operating state.

IPLADDR
  Indicates that the variable following it in parenthesis is the IPL address
  parameter.
  
  (ipl_address)
  Specifies the IPL address as four hexadecimal digits. No space is allowed
  between the keyword IPLADDR and the left parenthesis.

IPLPARM
  Indicates that the variable following it in parenthesis is an IPL parameter.
  
  (ipl_parameter)
  Specifies an IPL parameter to be issued with the Load command. This
  parameter can be 1 through 8 characters long. Lowercase letters are
  translated to uppercase letters. No space is allowed between the keyword
  IPLPARM and the left parenthesis.

CNAME
  Specifies the name of a Load profile to be sent to the processor's support
  element. If this keyword is not used, the LOAD operations management
  command that SA z/OS sends to the support element has a CNAME operand
  with the Load profile name that was specified in the configuration dialogs.

  If this keyword is not used and no Load profile name was specified in the
  configuration dialogs, SA z/OS omits the CNAME operand from the LOAD
  command sent to the support element. In this latter case, the support element
  uses the load parameters that were saved from the last LOAD command for
  the corresponding target image.
  
  profile_name
  Indicates the name of the Load profile. IBM recommends that you let
  SA z/OS select the profile name automatically rather than overriding
  SA z/OS's selection with the profile_name parameter.

Defaults

The LOAD command defaults to CLEAR(NO) and to FORCE(NO).

The ipl_address and ipl_parameter parameter default to the values defined in the
configuration dialogs, unless they have been changed with the ISQVARS or
ISQXOPT command, see ["ISQXOPT" on page 424], in which case the values
defined by the ISQVARS or ISQXOPT command are used.

For OCF-based processors, the ipl_address and ipl_parameter parameters default to
the values specified in the Load profile.
**Usage**

This common command assumes that the processor operations environment has been established and that the system console function has been assigned. For a target system operating on an LPAR-mode processor, it also assumes that the logical partition is active and that any required channels are online to the partition. If these assumptions are not correct, other initialization commands should be run first. These functions of the LOAD common command are all performed by the ACTIVATE common command.

If you do not specify `FORCE(YES)`, the LOAD common command checks to make sure it will not be disruptive by performing the following processing:

- If the target system runs in a logical partition and the same logical partition name is initialized to another target system in the same target hardware, the LOAD command is rejected and processing of the command halts.
- On a non-partitioned target system, if the target hardware has another target system initialized to it, the LOAD command is rejected and processing of the command halts.

**Restrictions and Limitations**

This command is not supported for coupling facility target systems.

If you do not specify FORCE(YES), the LOAD common command is rejected if the addressed target system has a status of UNKNOWN, CLOSED, or IPL COMPLETE. To perform the load process for a target system that is in a status condition of UNKNOWN or CLOSED, issue the ACTIVATE common command (see "ACTIVATE" on page 497). If the addressed target system has a status of IPLCOMPLETE, you can specify FORCE(YES) to perform the IPL process without deactivating the target system first.

For SNMP connections, if a CNAME keyword is specified with the LOAD command, any other load keyword operands are ignored. Note that this does not apply to the `FORCE(YES)` or `FORCE(NO)` parameter.

**Return Codes**

The ISQCCMD command (see "ISQCCMD" on page 355) generates the return codes for common command processing.

**LOAD for a VM Guest System**

**Purpose**

The CP IPL command is issued for the guest machine. (This assumes that the guest machine is already active.)

**Parameters**

**IPLADDR**
- This parameter must be present.

**IPLPARM**
- This parameter is translated into a LOADPARM operand.
Restrictions and Limitations

The CNAME parameter is not supported.

OOCOD

Purpose

The OOCOD common command allows you to:
- Query the status of the On/Off Capacity on Demand feature for a specific CPC
- Activate the On/Off Capacity on Demand feature
- Undo a previously performed Activate

Syntax

```
ISQCCMD target_system ProcOps_name -OOCOD

CMD ( STATUS )
CMD ( UNDO )
CMD ( ACTIVATE ) CTR ( contrno )
```

Requirements

The requirements for the OOCOD command to complete successfully are:
- Processor operations must be active.
- The addressed target system must be initialized, or, if the target hardware is addressed, at least one target system on that hardware must be initialized.
- For the STATUS command: No further requirements.
- For the ACTIVATE command: The On/Off Capacity on Demand feature must be enabled and installed and be in a status of not activated.
- For UNDO: This is only applicable after a successful ACTIVATE.
- With Console Workplace 2.10 and later versions, for ACTIVATE and UNDO, the flag 'Allow capacity change API requests' must be set in 'Customize API Settings' on the SE.

Parameters

For a definition of `target_system ProcOps_name, target_hardware_name.LPAR_name,` and `target_hardware_name`, see “ISQCCMD” on page 355.

CMD

The OOCOD command that is to be executed. This can be:

- **STATUS**
  Query the status of the On/Off Capacity on Demand feature and issue a report. If activated, the activation date is also provided.

  **Note:** The INSTALLED status is set when an On/Off Capacity on Demand contract is activated. If no contract is active, the status is NOT INSTALLED.
OOCOD

**ACTIVATE**
Put the On/Off Capacity on Demand feature in an active state. This command requires a valid contract number for successful execution.

**UNDO**
Put the On/Off Capacity on Demand feature in an inactive state.

**CTR**
Only required for the ACTIVATE command. The order number (contrno) must correspond to a valid contract number for the hardware that has the On/Off Capacity on Demand feature implemented.

**Note:** For SE Console Workplace 2.10 and later versions, contrno corresponds to the record ID of one of the On/Off Capacity on Demand records that have already been loaded on the processor hardware.

**Defaults**
The OOCOD command defaults to STATUS.

**Usage**
Use the OOCOD command to activate and deactivate the On/Off Capacity on Demand feature. You can retrieve the current On/Off Capacity on Demand feature settings with the STATUS command.

Note that the term Console Workplace refers to the title line that is displayed in the Main Application Window of each CPC Support Element. The following lists the SE Workplace Version numbers and their related IBM System names:

- 2.11: IBM System z196
- 2.10: IBM System z10
- 2.9: IBM System z9
- 2.8: IBM eServer zSeries 990

For more information about OOCOD, see System z Capacity on Demand User’s Guide, available for the particular System z on IBM Resource Link.

**Restrictions and Limitations**
The OOCOD command is only supported for SNMP connections. If you issue an OOCOD request for another ProcOps connection type, it will be rejected.

The OOCOD command applies only to hardware that supports the On/Off Capacity on Demand feature. For other hardware, the OOCOD command will fail.

**Condition Codes**
The following lists the condition codes that are returned in the form 00B000xx, where xx is the condition code:

- **83** Invalid OOCOD command provided (it must be STATUS, ACTIVATE or UNDO).
- **84** No Order Number (CTR) provided for the OOCOD ACTIVATE command.
- **85** The current ooCoD status of the target does not allow the action (for example, an UNDO without a previous ACTIVATE).
Return Codes

The ISQCCMD command (see "ISQCCMD" on page 355) generates the return codes for common command processing.

POWERMOD

Purpose

The POWERMOD common command allows you to:

- Query the power mode status of a specific CPC
- Change the power mode of a specific CPC (Static power save mode)

Static power save mode is designed to reduce power consumption on z196 when full performance is not required. It can be switched on and off during runtime with no disruption to currently running workloads, aside from the change in performance.

Syntax

```
ISQCCMD target_system_ProcOps_name target_hardware_name.LPAR_name target_hardware_name
POWERMOD
CMD(STATUS)
CMD(UPDATE) VAL(mode-value)
```

Requirements

The requirements for the POWERMOD command to complete successfully are:

- Processor operations must be active.
- The addressed target system must be initialized or, if the target hardware is addressed, at least one target system on that hardware must be initialized.
- Hardware Console Workplace must be version 2.11 or higher.
- For the STATUS command: No further requirements.
- For the UPDATE command: The CPC must be allowed or entitled to perform a power mode change.

Parameters

For a definition of `target_system_ProcOps_name`, `target_hardware_name.LPAR_name`, and `target_hardware_name`, see "ISQCCMD" on page 355.

CMD

The POWERMOD command that is to be executed. This can be:

- STATUS
  Query the power mode status of the CPC. The report issued by the command contains the following information:
  - Current power mode setting
  - Supported power mode values
  - Power mode change allowed (Y/N)
UPDATE

Change the CPC’s power mode, for example, from 100 (High Performance) to one of the supported Power Save values.

VAL

Only required for the UPDATE command. Contains the power mode the CPC should be changed to.

Defaults

The POWERMOD command defaults to STATUS.

Usage

Use the POWERMOD command to query and change the CPC’s power mode.

Note that the term 'Console Workplace' refers to the title line, displayed on the Main Application Window of each CPC Support Element (SE) or on the Hardware Management Console (HMC).

The following lists the SE/HMC Workplace Versions numbers and their related IBM System names:

- 2.11 - IBM System z196
- 2.10 - IBM System z10
- 2.9 - IBM System z9
- 2.8 - IBM eServer zSeries 990

Restrictions and Limitations

The POWERMOD command applies only to hardware that supports the Power Management feature. For other hardware, the POWERMOD command will fail.

The POWERMOD command is only supported for SNMP connections. If you issue a POWERMOD request for another Processor operations connection type, it will be rejected.

On air-cooled models, Static power save mode can be entered only once in a calendar day.

Condition Codes

The following lists the condition codes that are returned in the form 00B00xxx, where xx is the condition code:

- 097
  Unable to retrieve CPC’s current power mode status.
- 098
  Unable to retrieve CPC’s supported power modes list.
- 099
  Unable to retrieve power mode change allowed information.
- 100
  Unsupported POWERMOD command.
- 101
  Invalid POWERMOD ‘UPDATE’ parameter.
### Profile

**Purpose**

The PROFILE common command allows you to access the activation profiles of a CPC. Activation profiles contain configuration information about the CPC itself and its images (LPARs), as well as load information for the operating systems to be initialized on the CPC or its LPARs.

With PROFILE, the names of the activation profiles can be listed and the content of a profile can be queried. The profile content can be changed.

The activation profiles are stored in the Support Element of a CPC. They are used for CPC or image activation and for processor load operations that are executed using load profile information.

**Syntax**

```
ISQCCMD target_system_ProcOps_name PROFILE

TARGET(target_hardware_name.LPAR_name) target_hardware_name

CMD(OPEN)

CMD(CLOSE)

CMD(LIST) TYPE(ptype)

CMD(UPDATE) update parms

CMD(READ) read parms PT(proctype)
```

**ptype:**

- **RESET**
- **IMAGE**
- **LOAD**
- **GROUP**

**update parms:**

```
TYPE(ptype) NAME(pname) VAR(pvar) VAL(pval)
```

**read parms:**

```
TYPE(ptype) NAME(pname) VAR(pvar)
```

**pvar:**

- **POWERMOD**
Requirements

The requirements for the PROFILE command to complete successfully are:

Processor Operations environment

- Processor operations must be active.
- The addressed target system or target hardware LPAR must have been initialized using the ISQXIII command.

LPAR Management environment

- SA z/OS must be fully initialized.
- The processor hardware of the addressed target system or target hardware LPAR must be configured for connection type INTERNAL in the SA z/OS customization dialogs.

Parameters

For a definition of target_system_ProcOps_name, target_hardware_name.LPAR_name, and target_hardware_name, see "ISQCCMD" on page 355.

CMD

The profile command that is to be executed. This can be:
PROFILE

OPEN
This command must be executed as the first profile access command. It builds an internal list of the available profiles names for faster profile access. On successful completion an AOFA0020 report is returned, with the number of the available activation profiles for each of the profile types.

CLOSE
This command should be executed when your profile access is complete. It clears the internal profile name list. If the CLOSE is omitted, you can re-access the profiles without performing an OPEN command, however the internal profile list may no longer be current. On completion, an AOFA0020 report is returned.

LIST
Returns a list of the profile names for the specified profile TYPE in an AOFA0020 response report.

UPDATE
The content of the specified profile variable is changed to the value defined with the VAL parameter. The update response is returned in an AOFA0020 report. The update fails or is rejected if the hardware detects an invalid value.

READ
Returns the contents of the specified profile NAME and TYPE in an AOFA0020 response report. If a VAR parameter is added, only the content of the specified profile variable is returned.

Note that if the profile content is retrieved, variable names without a value may be listed. If this is the case, either the CPC does not support this profile variable, or a supported variable is not initialized.

You can use the HMC's CPC Operational Customization Task: Customize/Delete Activation Profiles to determine which profile variables are supported by a CPC's machine type.

TYPE
The activation profile type, \textit{ptype}, which can be one of the following:

RESET
This profile type is used to activate a CPC. The following profile variables are supported: IOCDS, ENDTSL, PRT, PRTT.

IMAGE
This profile type is used to activate images (LPARs). The following profile variables are supported: AUTOIPL, BRLBA, BPS, CS, CSRES, CPD, CPDRES, DEFCAP, ES, ESRES, GRPPRF, IPLADR, IPLPRM, IPLTPY, LUN, OSSLP, PWI, PWIC, PWMN, PWMX, WLME, WWPN.

LOAD
This profile type is used to load a CPC or an image. The following profile variables are supported: BRLBA, BPS, IPLADR, IPLPRM, IPLTPY, LUN, OSSLP, WWPN.

GROUP
This profile type is used to define capacity limits for a group of logical partitions. The CAPACITY profile variable is supported.

NAME
The activation profile name.

\textit{pname}
Name, 1-16 alphanumeric characters, without blanks. The named profile must be available on the Support Element.
VAR

The activation profile variable name `pvar`. The following is a list of variable names and their associated values.

AUTOIPL

This flag indicates if an automatic LOAD should be performed as part of the LPAR activation. The variable is valid for IMAGE profile only.

Value information:

YES
Perform an automatic LOAD after LPAR activation is complete, using the IPLTYP, IPLADR, and IPLPRM information that is stored in the IMAGE profile.

NO
Do not perform a LOAD after LPAR activation.

BRLBA

The boot record logical block address that is used for IPLs from a SCSI device. Valid for profile types IMAGE and LOAD.

Value information:

`pval`

The eight-byte (16 hexadecimal digit) boot record logical block address specifies the logical block address (LBA) of the boot record. (A boot record is used to locate an OS loader on an IPL disk. Normally, this boot record is located at LBA 0. The SCSI IPL function allows the boot record to be located at a different LBA.)

* When an asterisk (*) is specified as the UPDATE value, the actual data value is copied from the NetView Task Global variable BRLBA.

BPS

The boot partition specification that is used for IPLs from a SCSI device. Valid for profile types IMAGE and LOAD.

Value information:

`pval`

The boot program selector, a decimal value between 0 and 30, is used to select the section of the IPL disk on which the desired OS resides. (The SCSI IPL function allows up to 31 different operating systems to reside on one IPL disk.)

* When an asterisk (*) is specified as the UPDATE value, the actual data value is copied from the NetView Task Global variable BPS.

CAPACITY

The defined capacity limit not only for a single LPAR (see `The DEFCAP` parameter), but for a group of LPARs running on the same CPC, known as a capacity group. It represents the number of workload units (WLUs) a capacity group has assigned as its limit. Valid for profile type GROUP only.

Value information:

0
No group capacity limit defined.

1–2,147,483,647

Note that the maximum value represents the highest value accepted by the SE/HMC, which may not represent the correct capacity limit in your environment.

For more information about Group Capacity settings and how workload management and workload license charges are related, see `z/OS MVS Planning: Workload Management`, SA22-7602.
CPD
The number of logical central processors assigned to the partition for its dedicated use. The assignment occurs at partition activation time.

Value information:
- \( n \) The number of logical processors. For processor type GPP, 1 is the allowed minimum. For other processor types 0 is allowed. The maximum number of logical processors possible depends on your physical processor configuration.

CPDRES
The number of logical central processors assigned to the partition for its dedicated use. The logical processors are reserved to be used after partition activation, when needed.

Value information:
- \( n \) The number of logical processors. For processor type GPP, 1 is the allowed minimum. For other processor types 0 is allowed. The maximum number of logical processors possible depends on your physical processor configuration.

CPS
The number of logical central processors assigned to the partition, but shared with other partitions. The assignment occurs at partition activation time.

Value information:
- \( n \) The number of logical processors. For processor type GPP, 1 is the allowed minimum. For other processor types 0 is allowed. The maximum number of logical processors possible depends on your physical processor configuration.

CPSRES
The number of logical central processors assigned to the partition, but shared with other partitions. The logical processors are reserved to be used after partition activation, if needed.

Value information:
- \( n \) The number of logical processors. For processor type GPP, 1 is the allowed minimum. For other processor types 0 is allowed. The maximum number of logical processors possible depends on your physical processor configuration.

CS
The amount of Central Storage, in megabytes, allocated at partition activation time for its exclusive use.

Value information:
- \( n \) Enter a value that is compatible with the storage granularity supported by your CPC.

For storage granularity information, see the Processor Resource/Systems Manager Planning Guide, available for your CPC on IBM Resource Link.

CSRES
The amount of Central Storage, in megabytes, that can be allocated dynamically to the partition after activation.
Reserved storage provides the logical partition with an additional amount of storage that can be used only if it is not already being used by another active logical partition.

There is no minimum for reserved storage. Zero megabytes (0 MB) is a valid amount of reserved storage.

Value information:

0–n
Enter a value that is compatible with the storage granularity supported by your CPC.

For storage granularity information, see the Processor Resource/Systems Manager Planning Guide, available for your CPC on IBM Resource Link.

DEFCAP
The logical partition's defined capacity. Valid for profile type IMAGE.

Value information:

0
No defined capacity for the LPAR.
1–2147483647
Amount of defined LPAR capacity.

Note that the maximum value represents the highest value accepted by the SE/HMC, which may not represent the correct capacity limit in your environment.

The defined capacity expressed in terms of Millions of Service Units (MSUs) per hour. MSUs is a measure of processor resource consumption. The amount of MSUs that a logical partition consumes is dependent on the model, the number of logical processors that are available to the partition, and the amount of time that the logical partition is allocated. The defined capacity value specifies how much of the logical partition capacity is to be managed by Workload Manager (WLM) for the purpose of software pricing.

ENDTSL
The end time slice if the CPC image enters a wait state. Valid for the RESET profile type.

Value information:

YES
Indicates that a CPC Image should lose its share of running time when it enters a wait state.

NO
Indicates that a CPC Image should not lose its share of running time when it enters a wait state.

Note: This value can only be set if the processor running time type is set to 1, that is, set to a constant value (see The PRTT Parameter).

ES
The amount of Central Storage, in megabytes, allocated at partition activation time for use as expanded storage.

There is no minimum for expanded storage. Zero megabytes (0 MB) is a valid amount of expanded storage.

Value information:

0–n
Enter a value that is compatible with the storage granularity supported by your CPC.
**PROFILE**

For storage granularity information, see the *Processor Resource/Systems Manager Planning Guide*, available for your CPC on IBM Resource Link.

**ESRES**
The amount of Expanded Storage, in megabytes, that can be allocated dynamically to the partition after activation. This variable applies only if the partition mode is ESA/390 or ESA/390 TPF.

Reserved storage provides the logical partition with an additional amount of storage that can be used only if it is not already being used by another active logical partition.

There is no minimum for reserved storage. Zero megabytes (0 MB) is a valid amount of reserved storage.

**Value information:**

\(0-n\)

Enter a value that is compatible with the storage granularity supported by your CPC.

For storage granularity information, see the *Processor Resource/Systems Manager Planning Guide*, available for your CPC on IBM Resource Link.

**GRPPRF**
The name of the LPAR capacity group that the LPAR is a member of. This variable is valid for IMAGE profiles only.

**Value information:**

\(pval\)

An alphanumeric name with a maximum length of 8 characters.

* If you specify an asterisk as the UPDATE value, the actual data value is copied from the NetView task global variable GRPPRF.

**IPLADR**
The IPL address (load address) for IPLs from a channel attached device. (IPLTYP NORMAL) Valid for profile types IMAGE and LOAD.

**Value information:**

\(pval\)

Hexadecimal device number of a channel attached load device.

**NEXT**
Indicates that the IPLPRM value that is supplied by HCD through its hardware interface is to be used.

* When an asterisk (*) is specified as the UPDATE value, the actual data value is copied from the NetView Task Global variable IPLADR.

**Note:** Depending on the CPC hardware type, a device number of '0000' is accepted as IPLADR by the hardware. Do not use this value, either with SA z/OS ProcOps, or when using the HMC to specify a LOAD address. Unpredictable results may occur.

**IPLPRM**
The IPL parameter (load parm) for IPLs from a channel attached device. (IPLTYP NORMAL) Valid for profile types IMAGE and LOAD.

**Value information:**

\(pval\)

OS-supported, 8-character load parameter field of CMOS S/390® and zSeries hardware. See the OS documentation for information about the IPL parameters that it supports.
Indicates that the IPLPRM value that is supplied by HCD through its hardware interface is to be used.

* When an asterisk (*) is specified as the UPDATE value, the actual data value is copied from the NetView Task Global variable IPLPRM.

**IPLTYP**

The type of IPL.

**Value information:**

**NORMAL**

Use a channel attached device address to load a basic control program (OS). When this IPL type is used, IPLADR must be specified. IPLPRM is optional.

**SCSILOAD**

Uses the address of a SAN-attached SCSI device to load a BCP (OS).

When this IPL type is used, the following profile variables must be specified: BRLBA, BPS, LUN, WWPN. OSSLP is optional.

**SCSIDUMP**

Uses the address of a SAN-attached SCSI device to load a Standalone Dump Utility. The utility uses a dump area on the same SCSI device to store the main storage content.

When this IPL type is used, the following profile variables must be specified: BRLBA, BPS, LUN, WWPN. OSSLP is optional.

* When an asterisk (*) is specified as the UPDATE value, the actual data value is copied from the NetView Task Global variable IPLTYP.

**IOCDS**

The input/output configuration data set identifier. Valid for profile type RESET.

**Value information:**

*pval*

Two-character identifier of the IOCDS that is to be used for CPC activation with this profile.

* When an asterisk (*) is specified as the UPDATE value, the actual data value is copied from the NetView Task Global variable IOCDS.

**LUN**

The logical unit number of the SCSI IPL boot device. Valid for profile types: IMAGE, LOAD.

**Value information:**

*pval*

Eight-byte identifier (16 hexadecimal digits) of the logical unit representing the SCSI IPL device.

* When an asterisk (*) is specified as the UPDATE value, the actual data value is copied from the NetView Task Global variable LUN.

**OSSLP**

The operating-system-specific load parameter for IPL from SCSI device. Valid for profile types: IMAGE, LOAD.

For a READ operation, the data that is returned in the AOFA0020 response report is split into 8 lines. If OSSLP is not set, a string of 256 blanks is returned from the hardware. If not all OSSLP bytes are used, the string returned is always padded with blanks.

**Value information:**
pval
The operating system specific load parameters to be used for the
SCSILOAD or SCSIDUMP. The character string can be up to 256 bytes
long.

The OS-specific load parameter is optionally used to pass a string of
characters to the program that is being loaded. Neither the system nor
the machine loader interprets or uses these parameters. For example,
the OS-specific load parameter can be used to identify additional I/O
devices and related storage addresses that are required by the OS being
loaded.

* When an asterisk (*) is specified as the UPDATE value, the actual data
value is copied from the NetView Task Global variable OSSLP.

Use this method if you want to copy a mixed case character string, or
if the string contains multiple blank separated words.

PRT
Processor Running Time. Valid for the RESET profile type.

Value information:

1–100
User-defined processor running time.

Defines the amount of continuous time allowed for logical processors
to perform jobs on shared processors for the Defined CPC object.

Note:
1. This value can only be set if the processor running time type is set
to 1, that is, set to a constant value (see [The PRTT Parameter]).
2. If the processor running time type is set to 0, the value for
processor running time displayed with the READ command will
always be 0.

PRTT
Processor Running Time Type. Valid for the RESET profile type.

Value information:

0  The processor running time is dynamically determined by the system.
1  The processor running time is set to a constant value.

Defines whether the processor running time is dynamically determined by
the system or set to a constant value for the Defined CPC object.

PWI
The initial processing weight. Valid for profile type IMAGE.

Value information:

0  The CPC image does not represent a logical partition, or the LPAR has
only dedicated processors assigned to it. This value is not valid for
UPDATE.
1–999
Represents the relative amount of shared processor resources that is
initially allocated to the LPAR.

* When an asterisk (*) is specified as the UPDATE value, the actual data
value is copied from the NetView Task Global variable PWI.

Note: If the WLME flag has a value of YES, the PWI value must be in the
range from PWMN to PWMX.
**PWIC**

The initial processing weight capped indicator. Valid for profile type IMAGE.

**Value information:**

**YES**

The initial processing weight for the LPAR is capped. It represents the maximum share of processor resources regardless of the availability of excess processor resources.

**NO**

The initial processing weight for the LPAR is not capped. It represents the share of resources that is guaranteed to the LPAR when all CP resources are in use. When excess processor resources are available, the LPAR can use them if necessary.

* When an asterisk (*) is specified as the UPDATE value, the actual data value is copied from the NetView Task Global variable PWIC.

**PWMN**

The minimum processing weight. Valid for profile type IMAGE.

The guaranteed minimum share of non-dedicated processor weight, when all processing resources are in use.

**Value information:**

**0**

The CPC image does not represent a logical partition, or the LPAR has only dedicated processors assigned to it. This value is not valid for UPDATE.

**1-999**

A value that is less than or equal to the initial processing weight that is used to define the minimum relative amount of shared processor resources that is allocated to the LPAR.

* When an asterisk (*) is specified as the UPDATE value, the actual data value is copied from the NetView Task Global variable PWMN.

**Note:** This value is only applied if the WLME flag has a value of YES.

**PWMX**

The maximum processing weight. Valid for profile type IMAGE.

Maximum relative amount of shared processor resources for this LPAR.

**Value information:**

**0**

The CPC image does not represent a logical partition, or the LPAR has only dedicated processors assigned to it. This value is not valid for UPDATE.

**1-999**

A value that is greater than or equal to the initial processing weight that is used to define the maximum relative amount of shared processor resources that is allocated to the LPAR.

* When an asterisk (*) is specified as the UPDATE value, the actual data value is copied from the NetView Task Global variable PWMX.

**Note:** This value is only applied if the WLME flag has a value of YES.

**WLME**

The Workload Manager enabled flag. Valid for profile type IMAGE.

**Value information:**
YES
z/OS Workload Manager is used to manage shared processor resources for a logical partition, according to the settings of the minimum and maximum weight specifications.

NO
PR/SM is managing the processing weight. The PWMN and PWMX values do not apply.
* When an asterisk (*) is specified as the UPDATE value, the actual data value is copied from the NetView Task Global variable WLME.

WWPN
The worldwide port name of the disk controller for the SCSI IPL. Valid for profile type IMAGE.

Value information:
\( pval \)
The worldwide port name is the eight-byte permanent name (16 hexadecimal digits) that is assigned to the FC (fibre channel) adapter port of the SCSI target device containing the logical unit serving as the IPL device. The FC fabric must be configured in such a way that the FCP channel that is used for the IPL operation has access to this port.
* When an asterisk (*) is specified as the UPDATE value, the actual data value is copied from the NetView Task Global variable WWPN.

VAL
The value specification for an UPDATE operation.

\( pval \)
See the value information for the VAR parameter.

PT
The Processor Type parameter allows you to specify which processor type the corresponding activation profile information should be read or updated for.

\( proctype \)
The processor type:
- GPP General Purpose Processor
- AAP Application Assist Processor
- IFL Integrated Facility for LINUX processor
- ICF Internal Coupling Facility processor
- IIP z9 Integrated Information Processor

Notes:
1. The IFA processor type (Integrated Facility for Application processor) has been replaced by AAP.
2. Processor types other than GPP are only applicable for the IMAGE profile type and the activation profile variables PWI, PWIC, PWMN or PWMX.

Defaults
For UPDATE and READ actions, if a processor type is not specified with the PT parameter, the default value of GPP (general purpose processor) is used.

Usage
The PROFILE common command for SNMP connected processors can be used to manage important CPC and LPAR activation and operating system load information.

The supported set of profile variables can be queried and modified using the ISQCCMD PROFILE common command with automation procedures.
Automation of IPLs from SCSI devices can be performed using LOAD or IMAGE activation profiles.

The command response reports from the processor hardware are available as ProcOps ISQ901I messages with a console indicator of SC. Note that the message originator following the message identifier is either a ProcOps target system name, or a target hardware name together with an LPAR name. In order to get a copy of such hardware messages on your screen as ISQ901I messages, use the ISQXMON command.

This common command can also be used for processors or systems that you have configured with a BCP Internal Interface connection in your SA z/OS policy. If ProcOps is active, and your common command targets a processor or system that has the same name in your ProcOps and BCP Internal Interface configuration, ProcOps is always preferred. To avoid this, you must either stop ProcOps or use names that are uniquely assigned to either ProcOps or the BCP Internal Interface.

**Additional Programming Information**

On command completion, the immediate HW response report messages are also available in a PIPE KEEP with the name 'ISQ.SNMP'. You can directly access this HW data when using this KEEP as your first Pipe stage. See "Using Immediate ISQCCMD Common Command Responses" on page 357.

**Restrictions and Limitations**

The PROFILE common command is only supported for SNMP connections and BCP Internal Interface connections configured for LPAR Management. If you issue a PROFILE request for another ProcOps connection type, it will be rejected.

The PROFILE common command cannot be used to create or delete activation profiles; use the HMC to perform these tasks. The PROFILE command supports a subset of the available activation profile variables.

Processor types other than GPP are only applicable for the IMAGE profile type and the activation profile variables PWI, PWIC, PWMN or PWMX.

**Condition Codes**

The PROFILE common command uses the internal APROF service to process the requests. The following lists the condition codes that are returned in the form 00B000xx, where xx is the condition code:

62 The variable name that is specified in the request is not supported. The request is rejected.

63 The variable name that is specified in the request is not supported. The request is rejected.

64 The request was not recognized by the APROF service. The request is rejected. Contact IBM Support.

65 The value operand of the request was not recognized by APROF. The request is rejected. Contact IBM Support.

66 OPEN error. APROF could not allocate the profile list storage. The request is rejected. Contact IBM Support.

67 OPEN error. Profile was already opened. The request is rejected. Issue a CLOSE request first.

68 Profile TYPE error. Allowed types are RESET, IMAGE, and LOAD. The request is rejected.
PROFILE

69  Profile is not open. Issue an OPEN request first. The request is rejected.
6A  Profile not found error. If the named profile was created since the last OPEN, issue a CLOSE and reopen the profiles to access the new list of profiles. The request is rejected.
6B  Profile UPDATE error. Error in APROF parameter list. The request is rejected. Contact IBM Support.
6C  Profile UPDATE error. Internal variable name list error. The request is rejected. Contact IBM Support.
6D  Profile UPDATE error. The specified variable name is not valid. Re-specify the variable name and repeat the request. The request is rejected.
6E  Profile UPDATE error. The specified update value was not accepted. Re-specify the value and repeat the request. The request is rejected.
6F  Profile UPDATE error for OSSLP. A NetView Cglobal could not be accessed. The request is rejected. Contact IBM Support.

Return Codes

The ISQCCMD command (see “ISQCCMD” on page 355) generates the return codes for common command processing.

RESERVE

Purpose

The RESERVE common command allows you to gain exclusive control over a CPC and lock the execution of disruptive commands. While a CPC is reserved, no application, including the one holding the reserve, can issue hardware commands to the CPC itself or one of its logical partitions until the lock is released again. The lock also applies for manual CPC and LPAR operation using the CPC's Support Element or an HMC.

Syntax

```
/SM590000/SM590000
ISQCCMD  target_system_ProcOps_name  RESERVE

RCMD ( LIST )
RCMD ( OFF )
RNAME ( rnm )
```

Requirements

The requirements for the RESERVE command to complete successfully are:

- Processor operations must be active.
- The addressed target system must be initialized for this command to complete successfully. If the target hardware is addressed, at least one target system on that hardware must have the ProcOps status INITIALIZED.

Parameters

For a definition of target_system_ProcOps_name and target_hardware_name, see “ISQCCMD” on page 355.
RESERVE

RCMD
Reserve command to be executed.

LIST
Returns the current reserve ID, if one is set in the field APPLNAME of the AOFA0001 response report from the target hardware. If the APPLNAME field that is returned is empty, no reserve condition exists for the CPC. The format 'majorname.minorname' is used if the APPLNAME field indicates a reserve condition. The majorname '0AUTOOPS' is used if a reserve is set by ProcOps or other applications using the z900 API. The minorname returned is the reserve ID that is specified by an application.

ON
Sets a CPC reserve using the name that is specified with the RNAME parameter keyword.

OFF
Releases a previously set CPC reserve condition using the name that is specified with the RNAME parameter keyword.

RNAME
Reserve identification minor name.

\( rnm \)
Reserve ID minor name. The maximum length that is allowed is eight characters.

Defaults
If no RESERVE parameters are specified, an RCMD(LIST) is executed.

Usage
Locking a CPC with the RESERVE command for disruptive command execution should only be performed in recovery situations, where access to the CPC or its LPARs has to be controlled. The operator or recovery application can issue the ISQCCMD RESERVE to isolate and release a CPC.

Note that the hardware commands issue a reserve internally to indicate a busy condition to the HMC operator or to other applications for the duration of a command. This internal reserve is object specific (CPC, image) and is different from the ISQCCMD RESERVE in ProcOps.

The RESERVE command can be used to determine whether a Support Element has exclusive control enabled. Usually hardware service sets this control to make sure no hardware command can disrupt SE service activity. After service, exclusive control is usually reset to disable.

Issue a RESERVE RCMD(LIST) common command to determine whether exclusive control has been set from the CPC’s Support Element. The AOFA0001 report that is returned shows an application name in the format APPLNAME(\textit{netid.nau}), where \textit{netid} is the CPC’s netid, and \textit{nau} is the CPC’s nau.

The command response reports from the processor hardware are available as ProcOps ISQ900I messages with a console indicator of SC. Note that the message originator following the message identifier is either a ProcOps target system name, or a target hardware name together with an LPAR name. In order to get a copy of such hardware messages on your screen as ISQ901I messages, use the ISQXMON command.
Restrictions and Limitations

The RESERVE common command is only supported for SNMP connections. When issuing a RESERVE for other ProcOps connection types, it will be rejected.

The RESERVE hardware command works only for ProcOps sessions to the Support Element of the addressed CPC. ProcOps sessions to an HMC are not supported and the RESERVE command will fail.

The RESERVE hardware command is available for machine types from 2064 onwards.

The RESERVE hardware command is not available for machine types 967x or older S/390 CMOS processor types. For these machine types the RESERVE command will fail.

Return Codes

The ISQCCMD command (see “ISQCCMD” on page 355) generates the return codes for common command processing.

RESTART

Purpose

The RESTART command causes the target system processor to perform the restart function. This function consists of loading the restart program status word (PSW).

Syntax

```
>>>ISQCCMD   target_system_ProcOps_name       RESTART

-target_hardware_name.LPAR_name
-group_name
-subgroup_name
```

Requirements

Processor operations must be active for this command to complete successfully.

The addressed target system must be initialized for this command to complete successfully.

Parameters

For a definition of target_system_ProcOps_name, target_hardware_name.LPAR_name, group_name, and subgroup_name, see “ISQCCMD” on page 355.

Defaults

None.

Usage

The RESTART command is normally used to return the processor to an active processing state from a QUIESCE or SHUTDOWN condition.
The RESTART command can also be used to recover from various system problems.

Restrictions and Limitations
None.

Return Codes
The ISQCCMD command (see “ISQCCMD” on page 355) generates the return codes for common command processing.

RESTART of a VM Guest System
Purpose
For a guest machine a CP SYSTEM RESTART command is sent.

SECLOG
Purpose
The SECLOG command provides routing for security records as they are written in the Support Element (SE) or Hardware Management Console (HMC) logs of Processor operations for automation. The routing can be turned ON or OFF and the current routing STATUS can be determined. Depending on the session endpoint, the Security Log of an SE or HMC can be monitored.

Syntax

```plaintext
ISQCCMD target_system_ProcOps_name target_hardware_name SECLOG CMD(STATUS) CMD(slopt)
```

Requirements
The requirements for the SECLOG command to complete successfully are:
• Processor operations must be active
• The addressed target system must be initialized, or, if the target hardware is addressed, at least one target system on that hardware must be initialized.

Parameters
For a definition of `target_system_ProcOps_name` and `target_hardware_name`, see “ISQCCMD” on page 355.

CMD
Prefix for sub-command/Options.

slopt
The following sub-commands/Options are supported:

Status
Returns the current SECLOG setting.

ON
Turns ON the routing of SECLOG messages to Processor operations.
SECLOG

OFF
    Turns OFF the routing of SECLOG messages to Processor operations.

Defaults
The default of SECLOG is Sub-command CMD(STATUS).

Usage
The monitoring of Security Log messages in Processor operations exists for console events, such as User Access, Software Updates, Data Mirroring, Configuration Changes and additional information about operational commands. Use the task 'View Security Logs' available on the console Graphical User Interface (GUI) to view additional information. A typical scenario for SECLOG is to start it, before issuing an Operations Management or Configuration command, such as TCM or CBU. For further issues, the SECLOG messages in the Netlog of the Processor operations focal point, may be useful. It is recommended to turn SECLOG OFF again, once the management or configuration command is complete.

Restrictions and Limitations
The SECLOG command is only supported with SNMP connections.

Return Codes
The ISQCCMD command generates the return codes for common command processing.

START

Purpose
The START command causes the target system processor to perform the start function. This function consists of returning the operating system to the operating state from the stopped state initiated by the STOP command.

Syntax
```
    ISQCCMD -target_system_ProcOps_name -target_hardware_name.LPAR_name -group_name -subgroup_name -START
```

Requirements
Processor operations must be active for this command to complete successfully.

The addressed target system must be initialized for this command to complete successfully.

Parameters
For a definition of target_system_ProcOps_name, target_hardware_name.LPAR_name, group_name, and subgroup_name, see "ISQCCMD" on page 355.
**Defaults**

None.

**Usage**

The START command is equivalent to pressing the START key at a target system.

Use the START command to cancel the effects of the STOP command and allow the operating system to run again.

**Restrictions and Limitations**

• This command is not supported for coupling facility target systems.

**Return Codes**

The ISQCCMD command (see "ISQCCMD" on page 355) generates the return codes for common command processing.

**START of a VM Guest System**

**Purpose**

For a guest machine the CP BEGIN command is entered.

**STOP**

**Purpose**

The STOP command causes the target system processor to perform the stop function. This function consists of placing the processor in the stopped state at the end of the current unit of operation.

**Syntax**

```
ISSQCCMD target_system_ProcOps_name STOP
   target_hardware_name.LPAR_name
   group_name
   subgroup_name
```

**Requirements**

Processor operations must be active for this command to complete successfully.

The addressed target system must be initialized for this command to complete successfully.

**Parameters**

For a definition of target_system_ProcOps_name, target_hardware_name.LPAR_name, group_name, and subgroup_name, see "ISQCCMD" on page 355.

**Defaults**

None.
STOP

Usage

Use the STOP command to temporarily halt normal operations. You can use this command to interrupt processing to perform various hardware maintenance functions (such as cleaning the heads of a tape drive) and then continue processing from the point at which it stopped.

Note: The STOP command triggers a hardware status change event from the SE or HMC which sets the target hardware status to 'TARGET HARDWARE PROBLEM'. If you execute the ISQXDST command to display the target status, the compound status is displayed in the main panel. For more details about the target hardware status, refer to ISQVARS variable 'thwstat'.

Restrictions and Limitations

This command is not supported for coupling facility target systems.

Return Codes

The ISQCCMD command (see “ISQCCMD” on page 355) generates the return codes for common command processing.

STOP of a VM Guest System

Purpose

For a guest machine the CP STOP command is entered. (This will place the guest machine in a CP READ state, and a message is issued to this effect.)

STP

Purpose

The STP command allows you to manipulate server time protocol (STP) settings in an STP network. You can:

• Join, leave, or change an STP-only Coordinated Timing Network (CTN)
• Swap a Current Time Server (CTS) from Preferred to Backup, or vice versa
• Set Preferred, Backup, Arbiter and CTS server roles

Syntax

```plaintext
ISQCCMD
  target_system_ProcOps_name
  target_hardware_name
  target_hardware_name.LPAR_name
  STP

FORCE(YES)
  CMD(SET) STPID(stpid)
  Config

FORCE(NO)
  CMD(JOIN)
  CMD(SWAP)
  CMD(CHANGE)
  CMD(LEAVE)
```
Requirements

The requirements for the STP command to complete successfully are:

- Processor operations must be active.
- The addressed target system must be initialized, or, if the target hardware is addressed, at least one target system on that hardware must be initialized.
- The target hardware must be an STP-capable server. IBM System z196, IBM System z10 ECT, z10 BC, System z9 EC, z9 BC, and zSeries 990 and 890 with the required LIC installed are STP-capable.
- The target hardware must be an STP-enabled and STP-configured server.

Parameters

For a definition of target_system_ProcOps_name, target_hardware_name.LPAR_name, and target_hardware_name, see “ISQCCMD” on page 355.

CMD

Indicates the type of STP command to be performed. It can be one of the following:

JOIN

This command allows a CPC to join an STP-only CTN.

If the CPC is already participating in an STP-only CTN, it is removed from that CTN and joins the specified one.

If the CPC has an external time reference (ETR) ID, it is removed.

The target system cannot be the CTS.

LEAVE

This command removes a central processor complex (CPC) from an STP-only CTN. The target system cannot be the CTS.

CHANGE

This command is sent to the Defined CPC with the role of CTS in an STP-only CTN and changes the STP ID portion of the CTN ID for the entire STP-only CTN. The target system must be the CTS.

SET

This command sets the configuration for an STP-only Coordinated Timing Network (CTN). The target system must be the system that will become the Current Time Server (CTS).

SWAP

In a configured STP-only CTN, one CPC has the role of CTS. If the CTN has both a Preferred Time Server and a Backup Time Server configured, either one can be the CTS. This command swaps the role of CTS from Preferred Time Server to Backup Time Server or vice versa.

The target system must be the system that will become the CTS.

STPID

An alphanumeric string that represents the current STP identifier for the Defined CPC object. The string must not exceed 8 characters. It is applicable for the CMD options JOIN, CHANGE, SWAP, and SET.

FORCE

This parameter is only applicable for CMD (SET). It bypasses a number of
validity checks on server connectivity, and it allows the configuration of
erservers that may not be in POR-complete state or do not yet have coupling
link connectivity to the selected CTS.

**NO** Specifies that various verifications will be made before allowing the
configuration to be set. For example, connections between the Preferred
Time Server (CPC), Backup Time Server (CPC), and Arbiter will be
verified. If the connections are not there, the configuration will fail.

**YES**

Specifies that the configuration will be applied without verifications.

**Config**

Specifies the type of server that a configuration is being set for. It is only
applicable for CMD(SET). It can be one of the following:

**PREF** Assigns the Preferred Time Server role (the server that is assigned to
be the Preferred Stratum 1 server) in an STP-only CTN. The PREF
parameter is mandatory for the CMD(SET) command.

This is usually also the Current Time Server (CTS).

**BACKUP**

Assigns the Backup Time Server role in an STP-only CTN. This is the
server that is assigned to take over as the CTS, either because of a
planned or unplanned reconfiguration. This is usually a Stratum 2
server that should be attached to the Preferred Time Server as well as
all the servers that are attached to the Preferred Time Server.

If you do not specify BACKUP for the SET command, the BTS role will
be unset (that is, not configured).

**ARBITER**

Assigns the Arbiter role in an STP-only CTN. This is the server that is
assigned to provide additional means for the Backup Time Server to
determine whether it should take over as the Current Time Server.

If you do not specify ARBITER for the SET command, the Arbiter role
will be unset (that is, not configured). You must configure the Backup
Time Server (that is, specify the BACKUP parameter) if you configure
an Arbiter.

**CTS** Assigns the Current Time Server role in an STP-only CTN. This is the
server that is currently the Stratum 1 server for an STP-only CTN. The
only values that are allowed are Preferred or Backup. The CTS
parameter is mandatory for the CMD(SET) command.

**nodename**

An alphanumeric string that represents the name of the CPC and must not
exceed 8 characters.

**Defaults**

The STP command defaults to FORCE(NO).

**Usage**

Use this command to modify the STP configuration in an STP-only CTN.
Restrictions and Limitations

The STP common command is only supported for SNMP connections. If an STP request is issued for another processor operations connection type, it will be rejected.

Note: To perform an STP CMD(SET) command, all nodes that are referenced in the command should be defined in main HMC console.

Return Codes

The ISQCCMD command (see "ISQCCMD" on page 355) generates the return codes for common command processing.

STPDATA

Purpose

The STPDATA command allows you to query the Server Time Protocol (STP) configuration for the defined CPC object.

Syntax

```
ISQCCMD -target_system_ProcOps_name STPDATA -target_hardware_name -target_hardware_name.LPAR_name
```

Requirements

The requirements for the STPDATA command to complete successfully are:
- The target hardware must be an STP-capable server. IBM System z196, IBM System z10 ECT, z10 BC, System z9 EC, z9 BC, and zSeries 990 and 890 with the required LIC installed are STP-capable.
- The target hardware must be an STP-enabled server.
- Processor Operations environment
  - Processor operations must be active.
  - The addressed target system must be initialized, or, if the target hardware is addressed, at least one target system on that hardware must be initialized.
- LPAR Management environment
  - SA z/OS must be fully initialized.
  - The processor hardware of the addressed target system or target hardware LPAR must be configured for connection type INTERNAL in the SA z/OS customization dialogs.

Parameters

For a definition of target_system_ProcOps_name, target_hardware_name.LPAR_name, and target_hardware_name, see "ISQCCMD" on page 355.
STPDAT

Defaults

None.

Usage

Use this command to query the Server Time Protocol (STP) configuration for a specific CPC object. The data is returned in an AOFA0024 multiline report. For more information about this report format see the appendix “Response Messages, Error Strings, Condition Codes” in IBM Tivoli System Automation for z/OS Messages and Codes.

The command response reports from the processor hardware are available as processor operations ISQ900I messages with a console indicator of SC. Note that the message originator following the message identifier is either a processor operations target system name or a target hardware name together with an LPAR name. To have a copy of these hardware messages on your screen as ISQ901I messages, use the ISQXMON command.

This common command can also be used for processors or systems that you have configured with a BCP Internal Interface connection in your SA z/OS policy. If processor operations is active, and your common command targets a processor or system that has the same name in both your processor operations and BCP Internal Interface configuration, processor operations is always preferred. To avoid this, you must either stop processor operations or use names that are uniquely assigned to either processor operations or the BCP Internal Interface.

Additional Programming Information

On command completion, the immediate HW response report messages are also available in a PIPE KEEP with the name ISQ.SNMP. You can access this HW data directly when using this KEEP as your first Pipe stage. See “Using Immediate ISQCCMD Common Command Responses” on page 357.

Restrictions and Limitations

The STPDAT common command is only supported for SNMP connections and BCP Internal Interface connections that are configured for LPAR Management. When issuing a STPDAT request for another processor operations connection type, it will be rejected.

Condition Codes

The following condition codes are returned in the form 00B000xx, where xx is the condition code:

96 No STP configuration was found.

Return Codes

The ISQCCMD command (see “ISQCCMD” on page 355) generates the return codes for common command processing.
SYSRESET

Purpose

The SYSRESET command causes the target system processor to perform the system reset function. This function consists of resetting and initializing the processor.

Syntax

```
ISQCCMD
  target_system_ProcOps_name
  target_hardware_name.LPAR_name
  group_name
  subgroup_name
  SYSRESET
  CLEAR(NO)
  CLEAR(YES)

FORCE(NO)
  FORCE(NO)
  FORCE(YES)
```

Requirements

Processor operations must be active for this command to complete successfully.

The addressed target system must be initialized for this command to complete successfully.

Parameters

For a definition of `target_system_ProcOps_name`, `target_hardware_name.LPAR_name`, `group_name`, and `subgroup_name`, see “ISQCCMD” on page 355.

CLEAR(NO)

Indicates that the option of clearing storage is not being specified. This is the default value. No space is allowed between the keyword CLEAR and the left parenthesis.

CLEAR(YES)

Indicates that the option of clearing storage is being specified. No space is allowed between the keyword CLEAR and the left parenthesis.

FORCE

Requests conditional processing of commands that are disruptive to the operating system control program and application work that is in progress. Use of this operand is based on the operating state of the target processor.

NO

Specifies that processing of the command is to continue only if the target CPU control program is not in the operating state. Specify FORCE(NO) in a situation where you would not want to upset the control program work-in-progress.

YES

Specifies that processing of the command is to continue even if the target CPU control program is in the operating state. Specify FORCE(YES) in a situation where disruption of the control program work-in-progress is not important, such as in recovery situations.

The default, if FORCE is not explicitly stated, is FORCE(NO).
SYSRESET

FORCE(NO) requires that the associated command fail when useful work is taking place. For SYSRESET the formal fail criteria is operating when FORCE(NO) is specified or when FORCE defaults to NO. The command will fail whenever the processor is in this state.

Defaults

The SYSRESET command defaults to FORCE(NO) and to CLEAR(NO).

Usage

Issue SYSRESET to recover a processor from a machine stop condition. This command places the processor into the operating state from the check-stop state.

If you do not specify FORCE(YES), the SYSRESET command checks to make sure it will not be disruptive by performing the following processing:

- If the target system runs in a logical partition and the same logical partition name is initialized to another target system in the same target hardware, the SYSRESET command is rejected and processing of the command halts.
- On a non-partitioned target system, if the target hardware has another target system initialized to it, the SYSRESET command is rejected and processing of the command halts.

Restrictions and Limitations

This command is not supported for coupling facility target systems.

For SNMP-connected target hardware, the CLEAR parameter is supported by the zSeries processor family.

Return Codes

The ISQCCMD command (see “ISQCCMD” on page 355) generates the return codes for common command processing.

SYSRESET of a VM Guest System

Purpose

For a guest machine the CP SYSTEM RESET command is entered.

Parameters

CLEAR

If CLEAR(YES) is specified for a guest machine, a CP SYSTEM CLEAR is issued.

FORCE(NO)

Guest status is checked. If NOT OPERATING then a CP SYSTEM RESET command is issued.

TCDATA

Purpose

The TCDATA common command allows you to query the status and settings for a specific temporary capacity record.
### Syntax

```
ISQCCMD(target_system_ProcOps_name) TCDATA RECID(recid) target_hardware_name.LPAR_name target_hardware_name
```

### Requirements

The requirements for the TCDATA command to complete successfully are:

- On/Off Capacity on Demand (ooCoD) or Capacity Backup Unit (CBU) records, or both, must be installed on the addressed target hardware.
- The required temporary capacity record must be available on the addressed target hardware.

**Processor Operations environment**

- Processor operations must be active.
- The addressed target system must be initialized, or, if the target hardware is addressed, at least one target system on that hardware must be initialized.

**LPAR Management environment**

- SA z/OS must be fully initialized.
- The processor hardware of the addressed target system or target hardware LPAR must be configured for connection type INTERNAL in the SA z/OS customization dialogs.

### Parameters

For a definition of `target_system_ProcOps_name`, `target_hardware_name.LPAR_name`, and `target_hardware_name`, see "ISQCCMD" on page 355.

**RECID**

Defines the ooCoD or CBU record that TCDATA query should be executed for. If you specify a record ID of ‘LIST’, the TCDATA command will display a list of the installed capacity record IDs.

### Defaults

None.

### Usage

Use this command to query the status and settings for a specific temporary capacity record.

The data is returned in an AOFA0023 multiline report. For more information about this report format, see the appendix, "Response Messages, Error Strings, Condition Codes" in *IBM Tivoli System Automation for z/OS Messages and Codes*.

The command response reports from the processor hardware are available as ProcOps ISQ900I messages with a console indicator of SC. Note that the message originator following the message identifier is either a ProcOps target system name, or a target hardware name together with an LPAR name. To get a copy of such hardware messages on your screen as ISQ901I messages, use the ISQXMON command.
TCDATA

This common command can also be used for processors or systems that you have configured with a BCP Internal Interface connection in your SA z/OS policy. If ProcOps is active, and your common command targets a processor or system that has the same name in your ProcOps and BCP Internal Interface configuration, ProcOps is always preferred. To avoid this, you must either stop ProcOps or use names that are uniquely assigned to either ProcOps or the BCP Internal Interface.

Additional Programming Information

On command completion, the immediate HW response report messages are also available in a PIPE KEEP with the name 'ISQ_SNMP'. You can directly access this HW data when using this KEEP as your first Pipe stage. See "Using Immediate ISQCCMD Common Command Responses" on page 357.

Restrictions and Limitations

The TCDATA common command applies only to Series z10 hardware and later. The command fails for older hardware.

The TCDATA common command is only supported for SNMP connections and BCP Internal Interface connections configured for LPAR Management. When issuing a TCDATA request for another ProcOps connection type, it is rejected.

Condition Codes

The following condition codes are returned in the form 00B000xx, where xx is the condition code:

90 No temporary capacity record list was found.
91 Object not found. In this case, the temporary capacity record object was not found.
92 No matching record found for provided record ID.
93 Error found while parsing temporary capacity record XML data.

Return Codes

The ISQCCMD command (see "ISQCCMD" on page 355) generates the return codes for common command processing.

TCM

Purpose

The TCM common command allows you to add or remove temporary capacity resources (processors) for specific target hardware.

Syntax

```plaintext
/SM590000/SM590000
ISQCCMD
TCDATA
```

494 System Automation for z/OS: Operator's Commands
Requirements

The requirements for the TCM command to complete successfully are:

- Processor operations must be active.
- The addressed target system must be initialized, or, if the target hardware is addressed, at least one target system on that hardware must be initialized.
- On/Off Capacity on Demand (ooCoD) or Capacity Backup Unit (CBU) records, or both, must be installed on the addressed target hardware.
- The required resources (processors) must be available on the addressed target hardware.
- With Console Workplace 2.10 and later versions, for ADD and REMOVE, the flag 'Allow capacity change API requests' must be set in 'Customize API Settings' on the SE.

Parameters

For a definition of target_system.ProcOps_name, target_hardware_name.LPAR_name, and target_hardware_name, see "ISQCCMD" on page 355.

CMD

Possible commands are:

ADD Add processor resources to the addressed target hardware.
REMOVE Remove processor resources from the addressed target hardware.

TEST

Only applicable for CMD(ADD). It is used to indicate whether the activation of additional processors should be done on a test basis. The default is NO, that is, real activation.

Note: TEST(YES) is only supported for CBU records.

PRIO

Only applicable for CMD(ADD). It is used to indicate whether the activation of additional processors should be handled with priority. The default is NO, that is, no priority activation.

RECID

Defines the ooCoD or CBU record that the add or remove command should be executed for. The value consists of 8 alphanumeric characters.

SWMDL

Defines the Software Model that the add or remove command should result in. It is used to add or remove General Purpose Processors. The value consists of three alphanumeric characters.
PROC
Used to define up to five Special Purpose Processors, using the following variables:

**type**
Defines the processor type. The following values are allowed:

- **AAP**
  Application Assist Processor
- **IFL**
  Integrated Facility for Linux processor
- **ICF**
  Internal Coupling Facility processor
- **IIP**
  Integrated Information Processor
- **SAP**
  System Assist Processor

**step**
Defines the number of processors that should be added or removed. The valid value range for *step* is 0–9999.

*Note:* The two values must be separated with the `/` character.

**Defaults**
The PRIO and TEST parameters for CMD(ADD) default to NO.

**Restrictions and Limitations**
The TCM common command applies only to Series z10 hardware and later. The command fails for older hardware. The number of PROC parameters must not exceed five.

The TCM common command is only supported for SNMP connections. If a TCM request is issued for another processor operations connection type, it is rejected.

**Condition Codes**
The following condition code is returned in the form 00B000*xx*, where *xx* is the condition code:

- **94** Error found while composing temporary capacity add or remove XML data.

**Return Codes**
The ISQCCMD command (see "ISQCCMD" on page 355) generates the return codes for common command processing.
Chapter 7. Ensemble Commands

This chapter describes the ensemble commands provided by SA z/OS processor operations. Ensemble commands provide a single operator interface and API to monitor and control the resources of a zEnterprise Blade Extension HW.

ACTIVATE

Purpose

The ACTIVATE command accepts a request to perform asynchronously a complete initialization of the identified resource (virtual server or blade).

Syntax

```
ISQECMD
  target_hardware_ProcOps_name
  ensemble_ProcOps_name
  real_ensemble_name
  ACTIVATE
  BL bl_filter
  VS vs_filter

bl_filter:
  NAME(blname)
  CPC(cpcname)
  TYPE(bltype)

vs_filter:
  NAME(vsname)
  CPC(cpcname)
  BL(blname)
  VH(vhname)
  TYPE(vstype)
```

Requirements

The requirements for the ACTIVATE command to complete successfully are:

- Processor operations must be active.
- Processor operations must be in session with the ensemble that has the targeted resource in its scope.
- ACTIVATE VS command, virtual servers must be discovered to the Processor operations data model for the session using DISCOVERY VS ensemble command.

Parameters

For a definition of `target_hardware_ProcOps_name`, `ensemble_ProcOps_name` and `real_ensemble_name` refer to “ISQECMD” on page 360.

`*_filter`

Is a set of parameters allowing unambiguous identification of a target resource. Only full specification of all `*_filter` parameters ensures selection of a single object. Depending on the complexity of your ensemble, you may not need all the parameters; in most cases it is enough to specify NAME and CPC.
ACTIVATE

parameters to identify a target resource. If more than one resource corresponds to the selected filter, the command is rejected with condition code "00B6001B". For more information about the condition codes refer to the appendix "Response Messages, Error Strings, Condition Codes" in IBM Tivoli System Automation for z/OS Messages and Codes.

NAME
The name of the resource to ACTIVATE. The name is case-sensitive.

CPC
Defines the name (NAU) of a single zEnterprise Central Processor Complex (CPC) controlling appropriate IBM zEnterprise BladeCenter Extension (zBX).

BL
Defines the Unified Resource Manager-assigned name of the blade.

VH
Defines the Unified Resource Manager-assigned name of the virtualization host. The name is equal to the blade name.

TYPE
Defines the type of the objects to be activated. Values are:

For ACTIVATE BL command:

- **power**
  - the System z Power blade

- **system-x**
  - the System x® blade

- **isaopt**
  - the IBM Smart Analytics Optimizer blade

- **dpxi50z**
  - the DataPower® XI50 blade

For ACTIVATE VS command:

- **power-vm**
  - a virtual server running on a Power blade

- **x-hyp**
  - a virtual server running on a System x blade

Defaults
None.

Restrictions and Limitations

Virtualization hosts and virtual servers of the types "zvm" and "prsm" supported by the IBM zEnterprise Unified Resource Manager are not supported by the ISQECMD command.

The ACTIVATE command can be very disruptive and FORCE(NO) is not yet supported for the command so use this command carefully.

Usage

Use ACTIVATE BL command to activate a blade. This command also activates a virtualization host, hosted on the blade as a consequence of activating the hosting
environment. If "auto-start-virtual-servers" is true, the virtualization host activation will also activate all virtual servers on the virtualization host whose auto-start property is true.

Use ACTIVATE VS command to activate a virtual server.

Once the activation request is accepted, the command returns with an AOFB0001 report showing activation job initiation status and adds the "activate vs" request to internal list of submitted asynchronous commands. You may use LIST JOB command to see all successfully submitted asynchronous commands.

Once the activation job has completed on the target ensemble HMC, a job-completion notification is sent with the activation completion report as an AOFB0300 report and removes the activation request from the list of submitted asynchronous commands. The session should have a subscription for type JOB for appropriated resource classes (BL or VS) in order to receive job completion notifications and correctly manage the list of submitted asynchronous commands. For more information about AOFB* reports refer to the appendix "Response Messages, Error Strings, Condition Codes" in IBM Tivoli System Automation for z/OS Messages and Codes.

The command response reports and notifications from the ensemble HMC are available as ProcOps ISQ800I messages with a console indicator of SC. In order to get a copy of such hardware messages on your screen as ISQ801I messages, use the ISQXMON command.

**Return Codes**

The ISQECMD command generates the return codes for common command processing.

---

**BCDATA**

**Purpose**

The BCDATA ensemble command collects configuration and status information about addressed blade center and presents this information in a multi-line report.

**Syntax**

```
BCDATA
```

**Requirements**

The requirements for the BCDATA command to complete successfully are:
- Processor operations must be active.
- Processor operations must be in session with the ensemble that has the targeted resource in its scope.
Parameters

For a definition of target_hardware_ProcOps_name, ensemble_ProcOps_name and real_ensemble_name refer to "ISQECMD" on page 360.

**CPC**

Defines the name (NAU) of a single zEnterprise Central Processor Complex (CPC) controlling appropriate IBM zEnterprise BladeCenter Extension (zBX).

**BC**

Defines the Unified Resource Manager-assigned name of the single zBX blade center.

Defaults

None.

Restrictions and Limitations

None.

Usage

Use this command to get a snapshot of a single BC object properties. The data is returned as an AOFB0012 multiline report. For more information about this report format refer to the appendix "Response Messages, Error Strings, Condition Codes" in IBM Tivoli System Automation for z/OS Messages and Codes.

If you have more than one blade center in your ensemble, use BC and CPC parameters to limit the query to the single BC object. You may use the LIST BC ensemble command to see names and managing CPCs of all blade centers objects discovered for the target ensemble. Only a full specification of command parameters ensures selection of a single object. If more than one resource corresponds to the selected parameters, the command is rejected with condition code '00B6001B'. For more information about the condition codes refer to the appendix "Response Messages, Error Strings, Condition Codes" in IBM Tivoli System Automation for z/OS Messages and Codes.

The command response reports from the ensemble HMC are available as ProcOps ISQ800I messages with a console indicator of SC. In order to get a copy of such hardware messages on your screen as ISQ801I messages, use the ISQXMON command.

Return Codes

The ISQECMD command generates the return codes for common command processing.

BLDATA

Purpose

The BLDATA ensemble command collects configuration and status information about addressed zBX blade and presents this information in a multi-line report.
Syntax

```plaintext
ISQECMD target_hardware_ProcOps_name BLDATA ensemble_ProcOps_name real_ensemble_name CPC(cpcname)
BL(blnme) TYPE(bltype)
```

Requirements

The requirements for the BLDATA command to complete successfully are:
- Processor operations must be active.
- Processor operations must be in session with the ensemble that has the targeted resource in its scope.

Parameters

For a definition of `target_hardware_ProcOps_name`, `ensemble_ProcOps_name`, and `real_ensemble_name` refer to "ISQECMD" on page 360.

CPC
Defines the name (NAU) of a single zEnterprise Central Processor Complex (CPC) controlling appropriate IBM zEnterprise BladeCenter Extension (zBX).

BL  Defines the Unified Resource Manager-assigned name of the blade.

TYPE
Defines the type of a blade. Values are:
- `power` the System z Power blade
- `system-x` the System x blade
- `isaopt` the IBM Smart Analytics Optimizer blade
- `dpxi50z` the DataPower XI50 blade

Defaults

None.

Restrictions and Limitations

None.

Usage

Use this command to get a snapshot of the blade properties. The data is returned as an AOFB0016 multiline report. For more information about this report format refer to the appendix "Response Messages, Error Strings, Condition Codes" in IBM Tivoli System Automation for z/OS Messages and Codes.
If you have more than one blade in your ensemble, use BL, TYPE and CPC parameters to limit the query to the single blade object. You may use the LIST BL ensemble command to see names, types and managing CPCs of all blade objects discovered for the target ensemble. Only a full specification of command parameters ensures selection of a single object. If more than one resource corresponds to the selected parameters, the command is rejected with condition code "00B6001B". For more information about the condition codes refer to the appendix "Response Messages, Error Strings, Condition Codes" in IBM Tivoli System Automation for z/OS Messages and Codes.

The command response reports from the ensemble HMC are available as ProcOps ISQ800I messages with a console indicator of SC. In order to get a copy of such hardware messages on your screen as ISQ801I messages, use the ISQXMON command.

Return Codes

The ISQECMD command generates the return codes for common command processing.

CONDATA

Purpose

The CONDATA ensemble command collects basic information about the zEnterprise Hardware Management Console (HMC) managing the target ensemble and presents this information in a multi-line report.

Syntax

```
-- ISQECMD target_hardware_ProcOps_name CONDATA
```

Requirements

The requirements for the CONDATA command to complete successfully are:

- Processor operations must be active.
- Processor operations must be in session with the ensemble that has the targeted resource in its scope.

Parameters

For a definition of target_hardware_ProcOps_name, ensemble_ProcOps_name and real_ensemble_name refer to "ISQECMD" on page 360.

Defaults

None.

Restrictions and Limitations

None.
Usage

Use this command to get basic information about a console. The data is returned in an AOFB0022 multiline report. For more information about this report format refer to the appendix "Response Messages, Error Strings, Condition Codes" in IBM Tivoli System Automation for z/OS Messages and Codes.

The command response reports from the ensemble HMC are available as ProcOps ISQ800I messages with a console indicator of SC. In order to get a copy of such hardware messages on your screen as ISQ801I messages, use the ISQXMON command.

Return Codes

The ISQECMD command generates the return codes for common command processing.

DEACTIVATE

Purpose

The DEACTIVATE command accepts a request to deactivate asynchronously the identified resource (virtual server or blade).

Syntax

```
ISQECMD target_hardware_ProcOps_name DEACTIVATE
  ensemble_ProcOps_name
  real_ensemble_name

  BL bl_filter
    NAME(blname)
    CPC(cpcname)
    TYPE(bltype)
  VS vs_filter
    NAME(vsname)
    CPC(cpcname)
    BL(blname)
    VH(vhname)
    TYPE(vstype)
```

Requirements

The requirements for the DEACTIVATE to complete successfully are:

- Processor operations must be active.
- Processor operations must be in session with the ensemble that has the targeted resource in its scope.
DEACTIVATE

- For DEACTIVATE VS command, virtual servers must be discovered to the Processor operations data model for the session using DISCOVERY VS ensemble command.

Parameters

For a definition of target_hardware_ProcOps_name, ensemble_ProcOps_name and real_ensemble_name refer to "ISQECMD" on page 360.

*_filter
Is a set of parameters allowing unambiguous identification of a target resource. Only full specification of all *_filter parameters ensures selection of a single object. Depending on the complexity of your ensemble, you may not need all the parameters; in most cases it is enough to specify NAME and CPC parameters to identify a target resource. If more than one resource corresponds to the selected filter, the command is rejected with condition code "00B6001B". For more information about the condition codes refer to the appendix "Response Messages, Error Strings, Condition Codes" in IBM Tivoli System Automation for z/OS Messages and Codes.

NAME
The name of the resource to DEACTIVATE. The name is case-sensitive.

CPC
Defines the name (NAU) of a single zEnterprise Central Processor Complex (CPC) controlling appropriate IBM zEnterprise BladeCenter Extension (zBX).

BL
Defines the Unified Resource Manager-assigned name of the blade.

VH
Defines the Unified Resource Manager-assigned name of the virtualization host. The name is equal to the blade name.

TYPE
Defines the type of the objects to be activated. Values are:

For DEACTIVATE BL command:

   power
   the System z Power blade

   system-x
   the System x blade

   isaopt
   the IBM Smart Analytics Optimizer blade

   dpxi50z
   the DataPower XI50 blade

For DEACTIVATE VS command:

   power-vm
   a virtual server running on a Power blade

   x-hyp
   a virtual server running on a System x blade

FORCE
Requests conditional processing of commands that are disruptive to the operating system and application work that is in progress. Use of this operand is based on the operating state of the target resource.

NO
Specifies that processing of the command is to continue only if the
DEACTIVATE

target resource (blade or virtual server) is not in the operating state. Specify FORCE(NO) in a situation where you would not want to disrupt the operating environment.

**YES**  Specifies that processing of the command is to continue even if the target resource (blade or virtual server) is in the operating state. Specify FORCE(YES) in a situation where disruption of the operating environment is not important, such as in recovery situations.

**Defaults**

The DEACTIVATE command defaults to FORCE(NO).

**Restrictions and Limitations**

Virtualization hosts and virtual servers of the types "zvm" and "prsm" supported by the IBM zEnterprise Unified Resource Manager are not supported by the ISQECMD command.

**Usage**

Use DEACTIVATE BL command to deactivate a blade. This command also deactivates a virtualization host, hosted on the blade as a side effect of deactivating the hosting environment. The virtualization host deactivation will also deactivate all virtual servers on the virtualization host.

Use DEACTIVATE VS command to deactivate a virtual server.

If you do not specify FORCE(YES), the DEACTIVATE command checks to make sure it will not be disruptive by querying and validating the resource status.

Once the deactivation request is accepted, the command returns with an AOFB0001 report showing deactivation job initiation status and adds the "de-activate bl/vs" request to internal list of submitted asynchronous commands. You may use LIST JOB command to see all successfully submitted asynchronous commands.

Once the activation job has completed on the target ensemble HMC, a job-completion notification is sent with the deactivation completion report as an AOFB0300 report and removes the deactivation request from the list of submitted asynchronous commands.

The session should have a subscription for type JOB for appropriated resource classes (BL or VS) in order to receive job completion notifications and correctly manage the list of submitted asynchronous commands. For more information about AOFB* reports refer to the appendix "Response Messages, Error Strings, Condition Codes" in *IBM Tivoli System Automation for z/OS Messages and Codes*.

The command response reports and notifications from the ensemble HMC are available as ProcOps ISQ800I messages with a console indicator of SC. In order to get a copy of such hardware messages on your screen as ISQ801I messages, use the ISQXMON command.

**Return Codes**

The ISQECMD command generates the return codes for common command processing.
Purpose

Use this command to get a snapshot of basic properties of the virtual servers managed by the ensemble or the workload resource groups within the target ensemble and add the discovered data to the ProcOps data model.

Syntax

```
ISQECMD target_hardware_ProcOps_name ensemble_ProcOps_name real_ensemble_name DISCOVERY
```

Requirements

The requirements for the DISCOVERY command to complete successfully are:

- Processor operations must be active.
- Processor operations must be in session with the ensemble that has the targeted resource in its scope.

Parameters

For a definition of `target_hardware_ProcOps_name`, `ensemble_ProcOps_name` and `real_ensemble_name` refer to "ISQECMD" on page 360.

`TYPES`  
A comma-separated list of the VS types to be discovered. Supported values are:

- `POWER-VM`  
  Virtual servers of type "power-vm"

- `X-HYP`  
  Virtual servers of type "x-hyp"

- `ALL`  
  All Virtual Servers of the types "power-vm"and "x-hyp" should be discovered.

Defaults

`TYPES(ALL)`

Restrictions and Limitations

Virtual servers of the types "zvm" and "prsm" supported by the IBM zEnterprise Unified Resource Manager are not supported by the ISQECMD command.

Usage

ProcOps session to an ensemble manages internal data model allowing identification of ensemble objects based on their names, types, logical parents and
managing CPCs. The data model can be extended for dynamic resources using DISCOVERY VS and DISCOVERY WL ensemble commands.

Only objects discovered and available in the data model can be managed using ensemble commands and monitored for notifications.

- Use DISCOVERY VS command to add the virtual servers managed by the target ensemble to the data model. Only virtual servers physically residing on the zBX blade centers managed by the CPCs selected for the ensemble in the customization dialogs of SA z/OS are added to the data model.
- Use DISCOVERY WL command to add the workloads defined within the target ensemble to the data model.

The objects discovered using the command can be removed later from the data model using DROP ensemble command. You may use LIST command to show objects available in the data model and identify criteria for DISCOVERY/DROP commands.

DISCOVERY command just adds new objects and replaces existing objects with new data. The command does not delete any objects from the data model. It is recommended to use DROP and DISCOVERY commands in a sequence in case of inventory changes related to the VS and WL.

The number of discovered objects is returned in an AOFB0030 report. For more information about this report format refer to the appendix “Response Messages, Error Strings, Condition Codes” in IBM Tivoli System Automation for z/OS Messages and Codes.

The command response reports from the ensemble HMC are available as ProcOps ISQ800I messages with a console indicator of SC. In order to get a copy of such hardware messages on your screen as ISQ801I messages, use the ISQXMON command.

Return Codes

The ISQECMD command generates the return codes for common command processing.

DROP

Purpose

Use this command to delete specified objects from the ProcOps data model.

Syntax

```
ISQECMD target_hardware_ProcOps_name DROP WL

VS TYPES(ALL)

real_ensemble_name

TYPES(vstypes)
```

Requirements

The requirements for the DROP command to complete successfully are:

- Processor operations must be active.
Processor operations must be in session with the ensemble that has the targeted resource in its scope.

**Parameters**

For a definition of `target_hardware_ProcOps_name`, `ensemble_ProcOps_name` and `real_ensemble_name` refer to “ISQECMD” on page 360.

**TYPES**

A comma-separated list of the VS types to be deleted from data model.

Supported values are:

- **POWER-VM**
  Virtual servers of type "power-vm".

- **X-HYP**
  Virtual servers of type "x-hyp".

- **ALL**
  All Virtual Servers of the types "power-vm" and "x-hyp" should be deleted.

**Defaults**

TYPES(ALL)

**Restrictions and Limitations**

Virtual servers of the types “zvm” and “prsm” supported by the IBM zEnterprise Unified Resource Manager are not supported by the ISQECMD command.

**Usage**

ProcOps session to an ensemble manages internal data model allowing identification of ensemble objects based on their names, types, logical parents and managing CPCs.

Only objects discovered and available in the data model can be managed using ensemble commands and monitored for notifications.

- Use the DROP VS command to delete the virtual servers from the data model. The servers deleted from the data model are not available for further ensemble commands and the session will not receive any further notification related to the deleted objects.
- Use the DROP WL command to delete the workload resource groups from the data model. The workloads deleted from the data model are not available for further ensemble commands and the session will not receive any further notification related to the deleted objects.

The objects deleted with the command can be discovered later again using the DISCOVERY command. You may use the LIST command to show objects available in the data model and identify criteria for DISCOVERY/DROP commands.

It is recommended to use the DROP and DISCOVERY commands in a sequence in case of inventory changes related to the VS and WL to rediscover the data and keep the data model in sync with the ensemble definitions on the HMC.
The number of deleted objects is returned in an AOFB0030 report. For more information about this report format refer to the appendix "Response Messages, Error Strings, Condition Codes" in IBM Tivoli System Automation for z/OS Messages and Codes.

The command response reports from the ensemble HMC are available as ProcOps ISQ800I messages with a console indicator of SC. In order to get a copy of such hardware messages on your screen as ISQ801I messages, use the ISQXMON command.

**Return Codes**

The ISQECMD command generates the return codes for common command processing.

---

### ENSDATA

**Purpose**

The ENSDATA ensemble command collects basic information about the target ensemble and presents this information in a multi-line report.

**Syntax**

```
ISQECMD -target_hardware_ProcOps_name -ensemble_ProcOps_name -real_ensemble_name
```

**Requirements**

The requirements for the ENSDATA command to complete successfully are:

- Processor operations must be active.
- Processor operations must be in session with the ensemble that has the targeted resource in its scope.

**Parameters**

For a definition of `target_hardware_ProcOps_name`, `ensemble_ProcOps_name` and `real_ensemble_name` refer to "ISQECMD" on page 360.

**Defaults**

None.

**Restrictions and Limitations**

None.

**Usage**

Use this command to get basic information about the ensemble. The data is returned in an AOFB0017 multiline report. For more information about this report format refer to the appendix "Response Messages, Error Strings, Condition Codes" in IBM Tivoli System Automation for z/OS Messages and Codes.
ENSDATA

The command response reports from the ensemble HMC are available as ProcOps ISQ800I messages with a console indicator of SC. In order to get a copy of such hardware messages on your screen as ISQ801I messages, use the ISQXMON command.

Return Codes

The ISQECMD command generates the return codes for common command processing.

GETBCSTAT

Purpose

The GETBCSTAT ensemble command retrieves the current status of target blade centers and presents this information in a multi-line report.

Syntax

```
ISQECMD -target_hardware_ProcOps_name GETBCSTAT -ensemble_ProcOps_name -real_ensemble_name CPC(cpcname)
```

Requirements

The requirements for the GETBCSTAT command to complete successfully are:

- Processor operations must be active.
- Processor operations must be in session with the ensemble that has the targeted resource in its scope.

Parameters

For a definition of target_hardware_ProcOps_name, ensemble_ProcOps_name and real_ensemble_name refer to "ISQECMD" on page 360.

CPC

Defines the name (NAU) of a single zEnterprise Central Processor Complex (CPC) controlling appropriate IBM zEnterprise BladeCenter Extension (zBX).

BC

Defines the Unified Resource Manager-assigned name of the single zBX blade center.

Defaults

None.

Restrictions and Limitations

None.
GETBLSTAT

Usage

Use this command to retrieve status of one or more blade centers. The data is returned in an AOFB0020 report. For more information about this report format refer to the appendix "Response Messages, Error Strings, Condition Codes" in IBM Tivoli System Automation for z/OS Messages and Codes.

If you have multiple blade centers in your ensemble, use BC and CPC parameters to limit the query to specific blade centers. You may use the LIST BC ensemble command to see names and managing CPCs of all blade center objects discovered for the target ensemble.

The command response reports from the ensemble HMC are available as ProcOps ISQ800I messages with a console indicator of SC. In order to get a copy of such hardware messages on your screen as ISQ801I messages, use the ISQXMON command.

Return Codes

The ISQECMD command generates the return codes for common command processing.

GETBLSTAT

Purpose

The GETBLSTAT ensemble command retrieves the current status of target blades and presents this information in a multi-line report.

Syntax

```
ISQECMD getblstat CMD(target_hardware_ProcOps_name, ensemble_ProcOps_name, real_ensemble_name, GETBLSTAT, CPC(cpcname), BL(blname), TYPE(bltype))
```

Requirements

The requirements for the GETBLSTAT command to complete successfully are:

- Processor operations must be active.
- Processor operations must be in session with the ensemble that has the targeted resource in its scope.

Parameters

For a definition of `target_hardware_ProcOps_name`, `ensemble_ProcOps_name`, `real_ensemble_name` refer to "ISQECMD" on page 360.

- CPC
  - Defines the name (NAU) of a single zEnterprise Central Processor Complex (CPC) controlling appropriate IBM zEnterprise BladeCenter Extension (zBX).

- BL
  - Defines the Unified Resource Manager-assigned name of the blade.
GETBLSTAT

**TYPE**

Defines the type of a blade. Values are:

- **power**
  the System z Power blade
- **system-x**
  the System x blade
- **isaopt**
  the IBM Smart Analytics Optimizer blade
- **dpxi50z**
  the DataPower XI50 blade

**Defaults**

None

**Restrictions and Limitations**

None

**Usage**

Use this command to retrieve the status of one or more blades. The data is returned in an AOFB0020 report. For more information about this report format refer to the appendix "Response Messages, Error Strings, Condition Codes" in *IBM Tivoli System Automation for z/OS Messages and Codes*.

If you have multiple blades in your ensemble, use BL, TYPE and CPC parameters to limit the query to specific blades. You may use the LIST BL ensemble command to see names, types and managing CPCs of all blades discovered for the target ensemble.

The command response reports from the ensemble HMC are available as ProcOps ISQ800I messages with a console indicator of SC. In order to get a copy of such hardware messages on your screen as ISQ801I messages, use the ISQXMON command.

**Return Codes**

The ISQECMD command generates the return codes for common command processing.

---

**GETESTAT**

**Purpose**

The GETESTAT ensemble command retrieves the current ensemble communication status and presents this information in a single-line report.

**Syntax**

```
-ISQECMD target_hardware_ProcOps_name GETESTAT ensemble_ProcOps_name real_ensemble_name
```
**GETESTAT**

**Requirements**

The requirements for the GETESTAT command to complete successfully are:

- Processor operations must be active.
- Processor operations must be in session with the ensemble that has the targeted resource in its scope.

**Parameters**

For a definition of `target_hardware_ProcOps_name`, `ensemble_ProcOps_name` and `real_ensemble_name` refer to “ISQECMD” on page 360.

**Defaults**

None.

**Restrictions and Limitations**

None.

**Usage**

Use this command to retrieve an ensemble communication status representing the current communication status between the primary and alternate HMC. The data is returned in an AOFB0020 report. For more information about this report format refer to the appendix "Response Messages, Error Strings, Condition Codes" in *IBM Tivoli System Automation for z/OS Messages and Codes*.

The command response reports from the ensemble HMC are available as ProcOps ISQ800I messages with a console indicator of SC. In order to get a copy of such hardware messages on your screen as ISQ801I messages, use the ISQXMON command.

**Return Codes**

The ISQECMD command generates the return codes for common command processing.

---

**GETVHSTAT**

**Purpose**

The GETVHSTAT ensemble command retrieves the current status of target virtualization hosts (hypervisors) and presents this information in a multi-line report.

**Syntax**

```plaintext
>>ISQECMD target_hardware_ProcOps_name ensemble_ProcOps_name GETVHSTAT CPC(cpcname)
real_ensemble_name
```
GETVHSTAT

Requirements

The requirements for the GETVHSTAT command to complete successfully are:

- Processor operations must be active.
- Processor operations must be in session with the ensemble that has the targeted resource in its scope.

Parameters

For a definition of target_hardware_ProcOps_name, ensemble_ProcOps_name and real_ensemble_name refer to "ISQECMD" on page 360.

Parameters

CPC
Defines the name (NAU) of a single zEnterprise Central Processor Complex (CPC) controlling appropriate IBM zEnterprise BladeCenter Extension (zBX).

BL
Defines the Unified Resource Manager-assigned name of the blade.

VH
Defines the Unified Resource Manager-assigned name of the virtualization host. The name is equal to the blade name.

TYPE
Defines the type of a hypervisor. Values are:

- **power-vm**
  a virtualization host running on a Power blade

- **x-hyp**
  a virtualization host running on a System x blade

Defaults

None.

Restrictions and Limitations

Virtualization Hosts of types "zvm" and "prsm" supported by the IBM zEnterprise Unified Resource Manager are not supported by the ISQECMD command.

Usage

Use this command to retrieve status of one or more virtualization hosts. The data is returned in an AOFB0020 report. For more information about this report format refer to the appendix "Response Messages, Error Strings, Condition Codes" in IBM Tivoli System Automation for z/OS Messages and Codes.

If you have multiple hypervisors in your ensemble, use VH, BL, TYPE and CPC parameters to limit the query to specific virtualization hosts. You may use the LIST VH ensemble command to see names, types and managing CPCs of all virtualization hosts discovered for the target ensemble.
GETVSSTAT

The command response reports from the ensemble HMC are available as ProcOps ISQ800I messages with a console indicator of SC. In order to get a copy of such hardware messages on your screen as ISQ801I messages, use the ISQXMON command.

Return Codes

The ISQECMD command generates the return codes for common command processing.

GETVHSTAT

The command response reports from the ensemble HMC are available as ProcOps ISQ800I messages with a console indicator of SC. In order to get a copy of such hardware messages on your screen as ISQ801I messages, use the ISQXMON command.

Return Codes

The ISQECMD command generates the return codes for common command processing.

GETVSSTAT

Purpose

The GETVSSTAT ensemble command retrieves the current status of target virtual servers and presents this information in a multi-line report.

Syntax

```
/SM590000/SM590000
  ISQECMD
    target_hardware_ProcOps_name
    ensemble_ProcOps_name
    real_ensemble_name
  GETVSSTAT
    CPC(cpcname)
  VS(vsname)
    BL(biname)
    VH(vhname)
    TYPE(vstype)
/SM590000/SM630000
```

Requirements

The requirements for the GETVSSTAT command to complete successfully are:

- Processor operations must be active.
- Processor operations must be in session with the ensemble that has the targeted resource in its scope.
- Virtual Servers must be discovered to the Processor operations data model for the ensemble session using DISCOVERY VS ensemble command.

Parameters

For a definition of `target_hardware_ProcOps_name`, `ensemble_ProcOps_name` and `real_ensemble_name` refer to "ISQECMD" on page 360.

CPC

Defines the name (NAU) of a single zEnterprise Central Processor Complex (CPC) controlling appropriate IBM zEnterprise BladeCenter Extension (zBX).

VS

Defines the name of the virtual server. The name is unique to other existing virtual servers on the virtualization host/blade. The name is case-sensitive.

BL

Defines the Unified Resource Manager-assigned name of the blade.

VH

Defines the Unified Resource Manager-assigned name of the virtualization host. The name is equal to the blade name.

TYPE

Defines the type of the server. Values are:

```
power-vm
  a virtual server running on a Power blade
```
GETVSSTAT

x-hyp
   a virtual server running on a System x blade

 Defaults

None.

 Restrictions and Limitations

Virtual servers of types "zvm" and "prsm" supported by the IBM zEnterprise
Unified Resource Manager are not supported by the ISQECMD command.

 Usage

Use this command to retrieve the status of one or more discovered virtual servers.
The data is returned in an AOFB0020 report. For more information about this
report format refer to the appendix "Response Messages, Error Strings, Condition
Codes" in IBM Tivoli System Automation for z/OS Messages and Codes.

If you have multiple hypervisors in your ensemble, use VS, VH/BL, TYPE and
CPC parameters to limit the query to specific virtual server objects. You may use
the LIST VS ensemble command to see names, types and managing CPCs of all
virtual servers discovered for the target ensemble. Only servers discovered using
the DISCOVERY VS ensemble command and available in the ProcOps data model
can be queried.

The command response reports from the ensemble HMC are available as ProcOps
ISQ800I messages with a console indicator of SC. In order to get a copy of such
hardware messages on your screen as ISQ801I messages, use the ISQXMON
command.

 Return Codes

The ISQECMD command generates the return codes for common command
processing.

HMCSWITCH

 Purpose

The HMCSWITCH ensemble command initiates a primary/alternative Hardware
Management Console (HMC) role switch. The command is directed at a HMC that
is currently operating in an alternative role and defined as the alternative HMC in
the automation policy. As this command causes both consoles to be rebooted, any
active sessions (GUI-based or ProcOps-based) are terminated. Processor operations
session recovery recovers the session after a successful HMC switch in
environments supporting IP address swapping for the HMCs.

 Syntax

   >>>ISQECMD target_hardware_ProcOps_name HMCSWITCH
            ensemble_ProcOps_name
            real_ensemble_name

   HMCSWITCH
**Requirements**

The requirements for the HMCSWITCH command to complete successfully are:

- Processor operations must be active.
- Processor operations must be in session with the ensemble that has the targeted resource in its scope.
- TCP/IP Address or Hostname of alternative HMC should be defined in the policy for the ensemble.
- Both primary and the alternative HMC should use the same userid and password as defined in the policy for the ensemble.
- TCP/IP Address of the alternative HMC should be defined in the AT-TLS policy and loaded by the Policy Agent (PAGENT) in your z/OS environment.

**Parameters**

For a definition of `target_hardware_ProcOps_name`, `ensemble_ProcOps_name`, `and real_ensemble_name`, refer to ISQECMD in IBM Tivoli System Automation for z/OS Operator’s Commands.

**Defaults**

None.

**Restrictions and Limitations**

HMCSWITCH command requires IP address swapping configured for the ensemble HMCs. In this case the current primary HMC IP addresses is moved when making the alternative HMC the new primary. Otherwise, ProcOps does not recover the current ensemble session. You may verify your configuration using the ISQECMD CONDATA ensemble command. AOFB0022 multi-line report provides you with a property IPSWP showing IP swapping support.

When configured as recommended by IBM, the process of recovering from the failure of the primary HMC by a takeover of the alternative HMC includes the movement of the IP address from the former primary HMC to the new primary HMC. When this occurs, explicit redirection of API requests to the newly HMC Web Services API designated primary HMC is not needed.

However, if the IP address swapping is not possible in your network configuration, do not use the HMCSWITCH command.

**Usage**

Use this command to initiate a primary/alternative HMC role switch for the ensemble.

**Return Codes**

The ISQECMD command generates the return codes for common command processing.
LIST

Purpose

The LIST ensemble command allows you to query the ProcOps internal data discovered for target ensemble session. The list of discovered objects is presented as a multi-line report. The command does not interact with the HMC.

Syntax

```
ISQECMD target_hardware_ProcOps_name
    ensemble_ProcOps_name
    real_ensemble_name
        LIST BC bc_filter
        BL bl_filter
        CPC cpc_filter
        JOB
        VH vh_filter
        VS vs_filter
        WL wl_filter
        ZBX zbx_filter
```

**bc_filter:**

```
NAME(bcname)
CPC(cpcname)
```

**bl_filter:**

```
NAME(blname)
CPC(cpcname)
TYPE(bltype)
```

**cpc_filter:**

```
NAME(cpcname)
```

**vh_filter:**

```
NAME(vhname)
CPC(cpcname)
BL(blname)
TYPE(vhtype)
```

**vs_filter:**

```
NAME(vsname)
CPC(cpcname)
BL(blname)
TYPE(vstype)
```

**wl_filter:**

```
NAME(wlname)
```
**LIST**

**zbx_filter:**

```
     NAME(zbxname)
```

**Requirements**

The requirements for the LIST command to complete successfully are:

- Processor operations must be active.
- Processor operations must be in session with the ensemble that has the targeted resource in its scope.
- For LIST VS and LIST WL commands, Virtual Servers and Workloads must be discovered to the Processor operations data model for the ensemble session using DISCOVERY VS and DISCOVERY WL ensemble commands.

**Parameters**

For a definition of `target_hardware_ProcOps_name, ensemble_ProcOps_name` and `real_ensemble_name` refer to “ISQECMD” on page 360.

* _filter
  Is a set of parameters limiting the command to a subset of objects of the required object type. Only full specification of all *_filter parameters ensures selection of a single object.

**NAME**

The name of the object(s) to be selected for LIST report. The name is case-sensitive.

**CPC**

Defines the name (NAU) of a single zEnterprise Central Processor Complex (CPC) controlling appropriate IBM zEnterprise BladeCenter Extension (zBX).

**BL**

Defines the Unified Resource Manager-assigned name of the blade.

**VH**

Defines the Unified Resource Manager-assigned name of the virtualization host. The name is equal to the blade name.

**TYPE**

Defines the type of the objects(s) to be selected for LIST report. Values are:

For LIST BL command:

- **power**
  the System z Power blade

- **system-x**
  the System x blade

- **isaopt**
  the IBM Smart Analytics Optimizer blade

- **dpxi50z**
  the DataPower XI50 blade

For LIST VH and LIST VS commands:

- **power-vm**
  a virtual server running on a Power blade
x-hyp

a virtual server running on a System x blade

Defaults

None.

Restrictions and Limitations

Virtualization hosts and virtual servers of the types "zvm" and "prsm" supported by the IBM zEnterprise Unified Resource Manager are not supported by the ISQECMD command.

Usage

ProcOps session to an ensemble manages internal data model allowing identification of ensemble objects based on their names, types, logical parents and managing CPCs. The data model is filled in during initialization of the session using ISQXIII command and extended using DISCOVERY VS and DISCOVERY WL ensemble commands.

The following objects are discovered during initialization of the session:

- Ensemble
- Ensemble nodes (CPCs). Only CPCs selected for the ensemble in the customization dialogs of SA z/OS are added to the data model.
- IBM zEnterprise BladeCenter Extension (zBX) managed by the selected CPCs.
- zBX blade centers (BC), blades (BL) and virtualization hosts (VH)

Only objects discovered and available in the data model can be managed using ensemble commands and monitored for notifications. The LIST commands allow you to query the data model and identify objects for any other ensemble commands. LIST does not require communication to the ensemble HMC and generates no external IP traffic.

- Use LIST BC command to see one or more discovered blade centers.
- Use LIST BL command to see one or more discovered blades.
- Use LIST CPC command to see one or more discovered CPCs.
- Use LIST JOB command to see all the active asynchronous commands executed for the ensemble session using ACTIVATE and DEACTIVATE ensemble commands which are not yet completed.
- Use LIST VH command to see one or more discovered virtualization hosts (hypervisors).
- Use LIST VS command to see one or more discovered virtual servers discovered using DISCOVERY VS commands.
- Use LIST WL command to see one or more discovered workload resource groups discovered using DISCOVERY WL commands.
- Use LIST ZBX command to see one or more discovered zBXs.

The data is returned in an AOFB0020 multiline report. If no objects are available in the data model for specified _filter, the commands return empty report. For more information about this report format refer to the appendix "Response Messages, Error Strings, Condition Codes" in IBM Tivoli System Automation for z/OS Messages and Codes.
The command response reports from the ensemble HMC are available as ProcOps ISQ800I messages with a console indicator of SC. In order to get a copy of such hardware messages on your screen as ISQ801I messages, use the ISQXMON command.

**Return Codes**

The ISQECMD command generates the return codes for common command processing.

---

### SUBSCRIBE

**Purpose**

The SUBSCRIBE/UNSUBSCRIBE ensemble commands provide a filtering mechanism for event notifications received from the ensemble HMC for a session. The SUBSCRIBE command permits registration of a session to an ensemble for specific notification types and object classes.

**Syntax**

```
ISQECMD target_hardware_ProcOps_name
ensemble_ProcOps_name
real_ensemble_name

SUBSCRIBE

/LIST
```

**Requirements**

The requirements for the SUBSCRIBE command to complete successfully are:

- Processor operations must be active.
- Processor operations must be in session with the ensemble that has the targeted resource in its scope.

**Parameters**

For a definition of `target_hardware_ProcOps_name`, `ensemble_ProcOps_name` and `real_ensemble_name` refer to “ISQECMD” on page 360.

**TYPES**

A comma-separated list of the notification types. Supported values are:

- **STATUS**
  the status change notifications

- **PROPERTY**
  the property change notifications

- **INVENTORY**
  the inventory change notifications

- **JOB**
  the asynchronous commands completion notifications
SUBSCRIBE

ALL
   all of the above

NONE
   leaves the list of notification lists unchanged

CLASSES
   A comma-separated list of the object classes requiring notifications. Supported values are:

ENS
   the notifications for the ensemble

CPC
   the notifications for the CPCs

ZBX
   the notifications for the zBXs

RACK
   the notifications for the racks

BC
   the notifications for the blade centers

BL
   the notifications for the blades

VH
   the notifications for the virtualization hosts (hypervisors)

VS
   the notifications for the virtual servers

WL
   the notifications for the workload resource groups

ALL
   all of the above

NONE
   leaves the list of object classes unchanged

LIST
   When specified, report AOFB0019, containing the current subscription information for the selected ensemble session is returned. The parameter cannot be combined with TYPES and CLASSES.

Defaults

TYPES(ALL)

CLASSES(ALL)

Restrictions and Limitations

The TYPES and CLASSES are treated independently, just adding/removing subscriptions to internal types and classes lists. For example:

SUBSCRIBE TYPESPROPERTY CLASSESBC
SUBSCRIBE TYPESSTATUS CLASSESBL

results in a subscription for PROPERTY and STATUS for both BC and BL managed objects.
SUBSCRIBE

Usage

The IBM zEnterprise Unified Resource Manager includes an asynchronous notification facility by which client applications may subscribe to and receive notification messages regarding a set of predefined management events. These events include:

- Addition and removal of managed objects to/from the inventory of resources that are managed by the HMC (INVENTORY change notification).
- Changes to specified properties of managed object instances (PROPERTY change notification).
- Changes to the operational status of managed objects (STATUS change notification).
- Completion of asynchronously processed jobs (JOB completion notification).

The SUBSCRIBE command allows the ProcOps ensemble session to subscribe to these notifications for specific managed objects. The command is session-wide and affects all interested operators registered using ISQXMON command for a target ensemble, so use this command carefully. A session initialization using ISQXIII command performs initial subscription for the STATUS and JOB notifications for ALL classes. You may use SUBSCRIBE command to subscribe to additional notifications required for your automation procedures.

You may use SUBSCRIBE LIST command to query your current subscription for an ensemble session.

The notifications are delivered in a set of AOFB* reports:

- AOFB0100 for STATUS change
- AOFB0200 for PROPERTY change
- AOFB0300 for COMPLETION
- AOFB0400 for INVENTORY

For more information about these reports refer to the appendix "Response Messages, Error Strings, Condition Codes" in IBM Tivoli System Automation for z/OS Messages and Codes.

The notification reports from the ensemble HMC are available as ProcOps ISQ800I messages with a console indicator of SC. In order to get a copy of such notifications on your screen as ISQ801I messages, use the ISQXMON command.

Return Codes

The ISQECMD command generates the return codes for common command processing.

UNSUBSCRIBE

Purpose

The SUBSCRIBE/UNSUBSCRIBE ensemble commands provide a filtering mechanism for event notifications received from the ensemble HMC for a session. The UNSUBSCRIBE command permits the de-registering of the session to an ensemble for specific notification types and object classes.
UNSUBSCRIBE

Syntax

```
ISQECMD target_hardware_ProcOps_name UNSUBSCRIBE CLASSES(ALL)
ensemble_ProcOps_name TYPES(ALL)
real_ensemble_name TYPES(types)
CLASSES(classes)
```

Requirements

The requirements for the UNSUBSCRIBE command to complete successfully are:

- Processor operations must be active.
- Processor operations must be in session with the ensemble that has the targeted resource in its scope.

Parameters

For a definition of `target_hardware_ProcOps_name`, `ensemble_ProcOps_name` and `real_ensemble_name` refer to "ISQECMD" on page 360.

**TYPES**

A comma-separated list of the notification types. Supported values are:

- **STATUS**
  the status change notifications

- **PROPERTY**
  the property change notifications

- **INVENTORY**
  the inventory change notifications

- **JOB**
  the asynchronous commands completion notifications

- **ALL**
  all of the above

- **NONE**
  leaves the list of notification lists unchanged

**CLASSES**

A comma-separated list of the object classes requiring notifications. Supported values are:

- **ENS**
  the notifications for the ensemble

- **CPC**
  the notifications for the CPCs

- **ZBX**
  the notifications for the zBXs

- **RACK**
  the notifications for the racks

- **BC**
  the notifications for the blade centers
UNSUBSCRIBE

BL  the notifications for the blades
VH  the notifications for the virtualization hosts (hypervisors)
VS  the notifications for the virtual servers
WL  the notifications for the workload resource groups
ALL  all of the above
NONE  leaves the list of object classes unchanged

Defaults

TYPES(ALL)
CLASSES(ALL)

Restrictions and Limitations

The TYPES and CLASSES are treated independently, just adding/removing subscriptions to internal types and classes lists. For example:

UNSUBSCRIBE TYPES(PROPERTY) CLASSES(BC)

removes subscription to PROPERTY change notifications for ALL managed objects and removes subscription to ALL notification types for the blade centers.

Usage

The IBM zEnterprise Unified Resource Manager includes an asynchronous notification facility by which client applications may subscribe to and receive notification messages regarding a set of predefined management events. These events include:

- Addition and removal of managed objects to/from the inventory of resources that are managed by the HMC (INVENTORY change notification).
- Changes to specified properties of managed object instances (PROPERTY change notification).
- Changes to the operational status of managed objects (STATUS change notification).
- Completion of asynchronously processed jobs (JOB completion notification).

The UNSUBSCRIBE command allows the ProcOps ensemble session to remove a subscription for particular notification types and managed objects. The command is session wide and affects all interested operators for a target ensemble, so use this command carefully.

Return Codes

The ISQECMD command generates the return codes for common command processing.
**Purpose**

The VHDATA ensemble command collects configuration and status information about addressed zBX virtualization host (hypervisor) and presents this information in a multi-line report.

**Syntax**

```
/SM590000/SM590000
ISQECMD target_hardware_ProcOps_name ensemble_ProcOps_name real_ensemble_name
VHDATA CPC(cpcname)
/SM590000/SM590000
/BL(biname) VH(vhname) TYPE(vhtype)
```

**Requirements**

The requirements for the VHDATA command to complete successfully are:

- Processor operations must be active.
- Processor operations must be in session with the ensemble that has the targeted resource in its scope.

**Parameters**

For a definition of `target_hardware_ProcOps_name`, `ensemble_ProcOps_name` and `real_ensemble_name` refer to “ISQECMD” on page 360.

- **CPC**
  - Defines the name (NAU) of a single zEnterprise Central Processor Complex (CPC) controlling appropriate IBM zEnterprise BladeCenter Extension (zBX).

- **BL**
  - Defines the Unified Resource Manager-assigned name of the blade.

- **VH**
  - Defines the Unified Resource Manager-assigned name of the virtualization host. The name is equal to the blade name.

- **TYPE**
  - Defines the type of a hypervisor. Values are:
    - `power-vm` a virtualization host running on a Power blade
    - `x-hyp` a virtualization host running on a System x blade

**Defaults**

None.

**Restrictions and Limitations**

Virtualization Hosts of types "zvm" and "prsm" supported by the IBM zEnterprise Unified Resource Manager are not supported by the ISQECMD command.
**VHDATA**

**Usage**

Use this command to get a snapshot of the virtualization host properties. The data is returned as an AOFB0013 multiline report. For more information about this report format refer to the appendix "Response Messages, Error Strings, Condition Codes" in *IBM Tivoli System Automation for z/OS Messages and Codes*. If you have more than one blade/virtualization host in your ensemble, use BL, VH, TYPE and CPC parameters to limit the query to the single hypervisor object. You may use the LIST VH ensemble command to see names, types and managing CPCs of all virtualization hosts discovered for the target ensemble. Only full specification of command parameters ensures selection of a single object. If more than one resource corresponds to the selected parameters, the command is rejected with condition code '00B6001B'. For more information about the condition codes refer to the appendix "Response Messages, Error Strings, Condition Codes" in *IBM Tivoli System Automation for z/OS Messages and Codes*.

The command response reports from the ensemble HMC are available as ProcOps ISQ8001 messages with a console indicator of SC. In order to get a copy of such hardware messages on your screen as ISQ801I messages, use the ISQXMON command.

**Return Codes**

The ISQECMD command generates the return codes for common command processing.

---

**VSDATA**

**Purpose**

The VSDATA ensemble command collects configuration and status information about addressed zBX virtual server and presents this information in a multi-line report.

**Syntax**

```
ISQECMD target_hardware_ProcOps_name ensemble_ProcOps_name real_ensemble_name VSDATA CPC(cpcname)
```

```
VS(vsname) BL(blname) VH(vhname) TYPE(vstype)
```

**Requirements**

The requirements for the VSDATA command to complete successfully are:

- Processor operations must be active.
- Processor operations must be in session with the ensemble that has the targeted resource in its scope.
- Virtual Servers must be discovered to the Processor operations data model for the ensemble session using the DISCOVERY VS ensemble command.
Parameters

For a definition of target_hardware_ProcOps_name, ensemble_ProcOps_name and real_ensemble_name refer to "ISQECMD" on page 360.

CPC
Defines the name (NAU) of a single zEnterprise Central Processor Complex (CPC) controlling appropriate IBM zEnterprise BladeCenter Extension (zBX).

VS
Defines the name of the virtual server. The name is unique to other existing virtual servers on the virtualization host/blade. The name is case-sensitive.

BL
Defines the Unified Resource Manager-assigned name of the blade.

VH
Defines the Unified Resource Manager-assigned name of the virtualization host. The name is equal to the blade name.

TYPE
Defines the type of the server. Values are:

- power-vm
  a virtual server running on a Power blade
- x-hyp
  a virtual server running on a System x blade

Defaults

None.

Restrictions and Limitations

Virtual servers of types "zvm" and "prsm" supported by the IBM zEnterprise Unified Resource Manager are not supported by the ISQECMD command.

Usage

Use this command to get a snapshot of the virtual server properties. The data is returned as an AOFB0014 multiline report. For more information about this report format refer to the appendix "Response Messages, Error Strings, Condition Codes" in IBM Tivoli System Automation for z/OS Messages and Codes.

If you have more than one virtual server in your ensemble, use VS, BL, VH, TYPE and CPC parameters to limit the query to the single server object. You may use the LIST VS ensemble command to see names, types, hosting blades and managing CPCs of all virtual servers discovered for the target ensemble. Only a full specification of command parameters ensures selection of a single object. If more than one resource corresponds to the selected parameters, the command is rejected with condition code "00B6001B". For more information about the condition codes refer to the appendix "Response Messages, Error Strings, Condition Codes" in IBM Tivoli System Automation for z/OS Messages and Codes.

Only servers discovered using DISCOVERY VS ensemble command and available in the ProcOps data model can be queried.

The command response reports from the ensemble HMC are available as ProcOps ISQ800I messages with a console indicator of SC. In order to get a copy of such hardware messages on your screen as ISQ801I messages, use the ISQXMON command.
Return Codes

The ISQECMD command generates the return codes for common command processing.

WLDATA

Purpose

The WLDATA ensemble command collects basic configuration about a workload resource group object and presents this information in a multi-line report.

Syntax

```
ISQECMD target_hardware_ProcOps_name ensemble_ProcOps_name real_ensemble_name WLDATA WL(wlname)
```

Requirements

The requirements for the WLDATA command to complete successfully are:

- Processor operations must be active.
- Processor operations must be in session with the ensemble that has the targeted resource in its scope.
- Workloads must be discovered to the Processor operations data model for the ensemble session using DISCOVERY WL ensemble command.

Parameters

For a definition of `target_hardware_ProcOps_name`, `ensemble_ProcOps_name` and `real_ensemble_name` refer to “ISQECMD” on page 360.

`WL` Defines the display name specified for the workload resource group. The name is case-sensitive.

Defaults

None.

Restrictions and Limitations

None.

Usage

Use this command to get basic information about workload resource group and list of virtual servers assigned to the workload. The data is returned as an AOFB0015 multiline report. For more information about this report format refer to the appendix “Response Messages, Error Strings, Condition Codes” in IBM Tivoli System Automation for z/OS Messages and Codes.

If you have more than one workload defined in your ensemble, use WL parameter to limit the query to the single workload object. You may use the LIST WL ensemble command to see names of all workloads discovered for the target ensemble.
WLDATA

Only workloads discovered using the DISCOVERY WL ensemble command and available in the ProcOps data model can be queried.

The command response reports from the ensemble HMC are available as ProcOps ISQ800I messages with a console indicator of SC. In order to get a copy of such hardware messages on your screen as ISQ801I messages, use the ISQXMON command.

Return Codes

The ISQECMD command generates the return codes for common command processing.

ZBXDATA

Purpose

The ZBXDATA ensemble command collects configuration information about addressed IBM zEnterprise BladeCenter Extension (zBX) and presents this information in a multi-line report.

Syntax

```
ISQECMD target_hardware_ProcOps_name ZBXDATA CPC(cpcname) ensemble_ProcOps_name real_ensemble_name
```

Requirements

The requirements for the ZBXDATA command to complete successfully are:

- Processor operations must be active.
- Processor operations must be in session with the ensemble that has the targeted resource in its scope.

Parameters

For a definition of target_hardware_ProcOps_name, ensemble_ProcOps_name and real_ensemble_name refer to [“ISQECMD” on page 360](#).

CPC

Defines the name (NAU) of a single zEnterprise Central Processor Complex (CPC) controlling appropriate IBM zEnterprise BladeCenter Extension (zBX).

Defaults

None.

Restrictions and Limitations

None.

Usage

Use this command to get a snapshot of a single zBX object. The data is returned as an AOFB0011 multiline report. For more information about this report format refer
to the appendix "Response Messages, Error Strings, Condition Codes" in IBM Tivoli System Automation for z/OS Messages and Codes.

If you have more than one zBX in your ensemble, use CPC parameter to limit the query to a single ZBX object. You may use the LIST zBX ensemble command to see all zBX objects with their managing CPCs for the target ensemble.

The command response reports from the ensemble HMC are available as ProcOps ISQ800I messages with a console indicator of SC. In order to get a copy of such hardware messages on your screen as ISQ801I messages, use the ISQXMON command.

**Return Codes**

The ISQECMD command generates the return codes for common command processing.
Chapter 8. PSM Commands - Special Requests

The following special requests can be executed by the ProcOps Service Machine (PSM) Command Server. To issue these requests you must be logged on to the PSM. Requests can either only be entered directly from the PSM console or be used with the ISQPSM command as well as being entered from the PSM console:

- PSM commands that operate with ISQPSM are:
  - "ISQACT"
  - "ISQCLEAR"
  - "ISQUERY" on page 534
  - "ISQSTATUS" on page 534
  - "ISQTRACE" on page 535

- PSM console commands that can only be entered from the console are:
  - "ISQMSG"
  - "ISQPSM" on page 534
  - "STOPALL" on page 535

ISQACT

```bash
ISQACT MH
```

ISQACT is used to activate a thread after it has stopped.

MH indicates that the Message Handler thread is to be reactivated. This is useful in cases where the MH thread fails or terminates itself.

ISQCLEAR

```bash
ISQCLEAR M
```

ISQCLEAR empties the Messages Queue or Commands Queue. All entries are emptied. The queue itself is not deleted.

ISQMSG

```bash
ISQMSG INDENT MSGID ON OFF
```

ISQMSG is used to change the way that messages are displayed to the console user that is logged on the PSM. If these messages are also routed to NetView it also alters the form of the messages.

If INDENT ON is specified this alters the indentation of each message. Messages that are issued by different threads are indented by different amounts. This makes it easier to see the flow of a thread because all its messages are indented by the
same amount. Trace messages (activated by an ISQTRACE request) are also
indented. Messages from the Main Thread are not indented. Messages that are
issued as responses to PSM requests are not indented. If an indented message
continues on further lines, the continuation lines are not indented.

INDENT OFF causes all non-trace messages to be displayed without indentation.
The initial setting for the PSM is INDENT OFF. This initial setting is set each time
the PSM Control Program is started.

If MSGID OFF is specified, this causes each non-trace message to be displayed
without its message ID. This may be useful when INDENT ON is in effect, because
it reduces the length of each message and reduces the continuation of message text
on subsequent lines.

MSGID ON causes all non-trace messages to be displayed with their message IDs.
The initial setting for the PSM is ON. Trace messages do not have message IDs.

**Restrictions and Limitations**
The ISQMSG command can only be entered directly from the console of the PSM.

---

**ISQPSM**

The ISQPSM command starts the PSM server. This command should only be
entered after the PSM server has been stopped successfully with a STOPALL
command and the PSM virtual machine is in CMS mode.

**Restrictions and Limitations**
The ISQPSM command can only be entered directly from the console of the PSM.

---

**ISQUERY**

ISQUERY displays:
Q  How many entries are in the Message Queue.
E  A list of current events and event monitors.

Listing the current events and event monitors is useful in problem determination.

---

**ISQSTATUS**

ISQSTATUS gives the status of important components of the PSM operation.
**ISQTRACE**

```
ISQTRACE component ON
```

ISQTRACE is used to start or stop PSM tracing of a component of the PSM. Trace messages are always written to the VM console.

The value of `component` can be one of:

- **MSERV** Message server thread.
- **CSERV** Command server thread.
- **MH** Message handler thread.
- **LOGGER** Logger thread.
- **CNSERVER** Console server thread.
- **TCPIP** Selected TCPIP requests within the Message Server and Command Server threads.

If any trace is set on, regular messages issued to the VM console as part of the VMCONIO stream are sent to the VM console and are no longer routed back to ProcOps NetView.

Trace settings are maintained when a PSM is stopped. If the PSM is then restarted (without a re-logon of the CMS machine) the previous trace settings are retained.

When a PSM virtual machine is started all traces are set off.

**STOPALL**

```
STOPALL
```

STOPALL stops all threads and terminates the PSM control program. The PSM virtual machine remains active.

**Restrictions and Limitations**

The STOPALL command can only be entered directly from the console of the PSM.
Part 5. Appendixes
Appendix A. Definitions for I/O Operations Commands

This appendix provides additional information for using I/O operations (I/O-Ops) commands.

ESCON/FICON Director

Figure 162 is an example of a diagram that is used in this document to illustrate an ESCON director (or switch, as it is usually called).

![ESCON Director Diagram]

Figure 162. Sample of an ESCON Director Diagram

Switch Identifiers

A switch device number is a hexadecimal number containing up to four digits with any leading 0’s (zeros) padded by I/O-Ops. The ADDRESS parameter of the IODEVICE macro in the I/O Definition File (IODF) specifies the switch device number.

In many I/O-Ops commands, a switch device number is required in the input to identify a switch that is allocated to the issuing I/O-Ops (primary host). The program also displays the switch device number on output.

Two related terms apply in this context:
- A logical switch number (LSN) is a 2-digit hexadecimal number that is assigned with the SWITCH parameter on the CHPID macro of the IOPC. I/O-Ops does not use an LSN as input, but it does display an LSN in the output of many of its Display and Query commands.
- Hardware Configuration Definition (HCD) uses the term switch identifier (Switch ID), which is identical with the switch’s LSN.

**Note:** In a cascaded switch environment, it is recommended that all switches are defined to the HCD including their device numbers. This allows I/O-Ops to show the LSN also for cascaded switches.
### Switch Unique Identifier

You can use the DISPLAY SWITCH command to display the 26-character switch unique identifier. The format is:

```
009032002IBM26000001234567
```

- 12-character sequence number
- 2-character plant of manufacturer
- 3-character manufacturer (IBM)
- 3-character model number (002 is model 2)
- 6-character machine type

### Logical Switch Number (LSN)

An LSN is a 2-digit hexadecimal number that is assigned with the SWITCH parameter on the CHPID macro of the IOCP. I/O-Ops does not use an LSN as input, but it does display an LSN in the output of many of its display and query commands. Note that the LSN will show “- -” when this value is not valid. This can happen only to cascaded switches that have no channel attachment -- which is normally the case -- to the channel subsystem that I/O-Ops is running on. For this reason, it is recommended that all switches are defined to the Hardware Configuration Definition (HCD) including their device numbers. This allows I/O-Ops to resolve the LSN also for the above-mentioned switches.

In many I/O-Ops commands, a switch device number is required in the input to identify a switch that is allocated to the issuing I/O-Ops (primary host). The program also displays the switch device number on output.

### Port Addresses

A port address is a two-digit hexadecimal number that identifies an addressable port on a switch.

### Switch Ports

- **CH** A CNC or CVC CHPID is defined for this port.
- **CU** A control unit is assigned to this port.
- **CHCH** More than one CNC or CVC CHPID is assigned to this port.
- **CHCU** Both a CHPID and a control unit are assigned to this port. CHCU is valid if the CHPID is a CTC channel. I/O-Ops does not report such a dual assignment as a configuration mismatch. However, CHCU is not a valid combination for any other type of CHPID. I/O-Ops does report such a dual assignment as a configuration mismatch.
- **CV** An ESCON Converter Model 2 (CV converter) is attached to this port. A CV converter (9035) converts ESCON protocol to a parallel CHPID operating in block or byte mode.
- **CVCH** A CV converter and an ESCON channel (CNC or CVC) are both assigned to this port. I/O-Ops reports such a dual assignment as a configuration mismatch.
- **CVCU** A CV converter and a control unit are assigned to this port. I/O-Ops reports such a dual assignment as a configuration mismatch.
- **ISL** An Inter-Switch-Link is connected to this port.
Allow or Prohibit Mask of a Port

Each port is either allowed to have, or prohibited from having, a dynamic connection with another port on the same switch.

Collectively, these pairs of connectivity attributes form an allow/prohibit mask.

A Designates the ports for which a dynamic connection is allowed

P Designates the ports for which a dynamic connection is prohibited

An allow/prohibit mask is displayed by I/O-Ops as part of a matrix of a switch configuration. The program displays switch matrixes in its sample I/O-Ops ISPF dialog. See examples in IBM Tivoli System Automation for z/OS User’s Guide.

Switch Port Hardware Status on a Display Command

On the output of DISPLAY commands, I/O-Ops displays the hardware status for a port, as shown in Figure 163.

Port Status

H S C P

P - Port is prohibited from being dynamically connected with at least one other port
xx - Address of the port with which this port is statically connected
b - Port is DCM ineligible
B - Port is blocked from all communications
+ - Port is DCM eligible and allowed for DCM activities
- - Port is DCM eligible and not allowed for DCM activities

b - Port is online
A - Attachment of port is not valid
Cx - Port is 1 of 3 ports in a chain (C is displayed to the left of another state)
D - Bridge port degraded
I - Bridge port offline (inactive)
L - Link failure detected
M - Port is in maintenance mode
N - Port is not installed
O - Port is offline
S - Service is required for this port

Figure 163. Switch Port Hardware Status Format

For further information, see the program information block (PIB), which is described in the output block listed in “QUERY SWITCH” in IBM Tivoli System Automation for z/OS Programmer’s Reference.

Logical Tokens

A logical token is an identifier created by hardware configuration description (HCD) for each I/O resource that is defined in an input output definition file (IODF). If two or more systems share an IODF, they will have the same logical token for the same I/O resource.
Physical Tokens

A physical token is a 32-byte field that describes an I/O resource. Information in the physical token comes from either a node element descriptor (NED) or a node descriptor (ND). A NED is a 32-byte field that describes a node element of an I/O resource. A NED is associated with a physical control unit or device. An ND is a 32-byte field that describes a node interface of an I/O resource. An ND identifies a physical CHPID. When a physical token is input, the first 4 bytes are not used for comparison.

Channel Types

Types of channels:

- **BLOCK**: A parallel channel operating in block mode
- **BYTE**: A parallel channel operating in byte mode
- **CFR**: Coupling facility receiver
- **CFS**: Coupling facility sender
- **CFP**: Coupling facility peer channel
- **CBR**: Integrated Cluster Bus receiver
- **CBS**: Integrated Cluster Bus sender
- **CBP**: Integrated Cluster Bus peer
- **CBY_P**: An ESCON to byte converter channel
- **CIB**: Coupling over Infiniband
- **CNC_P**: A point-to-point ESCON channel
- **CNC_S**: A switched ESCON channel
- **CNC_***: A switched or point-to-point ESCON channel
- **CTC_P**: A point-to-point channel-to-channel channel
- **CTC_S**: A switched channel-to-channel channel
- **CTC_***: A switched or point-to-point channel-to-channel channel
- **CVC_P**: An ESCON to block converter channel
- **DSD**: Direct system device
- **EIO**: An emulated I/O channel
- **FC**: A FICON point-to-point channel
- **FC_S**: A FICON switched channel
- **FC_***: A FICON incomplete channel
- **FCP**: A FCP channel
- **FCV**: A FICON-to-ESCON bridge
- **ICR**: Internal coupling receiver
- **ICS**: Internal coupling sender
- **ICP**: Internal coupling peer
- **IQD**: Internal queued direct communication
- **ISD**: Internal system device
- **NATIVE**: Native interface
- **OSA**: An open systems adapter channel
- **OSC**: OSA console
- **OSD**: An OSA direct express channel
OSE  An OSA express channel
OSN  An OSA NCP channel
UNDEF A channel that is not defined

Note: The last character of the DCM managed channels’ acronyms is 'M', for example, FCM or CNCSM.

I/O-Ops can display, or return data on, all the channels known to it. For ESCON channels, FICON channels, and for parallel channels operating in block mode, the program can be used to change the connectivity between these resources and others in their paths.

A Parallel Channel
I/O-Ops can display information about parallel channels that operate in either byte (BY) or block (BL) mode. I/O-Ops can also change connectivity to a parallel channel operating in block mode.

An ESCON (CNC) Channel
A CNC channel path is used to transfer data between a host system image and an ESCON control unit. I/O-Ops can display data about a CNC channel path and change its connectivity.

An ESCON Channel Operating in Converted (CVC) Mode
A CVC channel path transfers data in blocks and a CBY channel path transfers data by bytes. A CVC or CBY channel path is an ESCON channel that has been converted so that it can interface with an ESCON Converter Model 1 (9034), or equivalent, for communication with a parallel control unit.

A CVC or CBY path always resembles a point-to-point parallel path, regardless whether it passes through a switch or not. Thus, a switchable CVC or CBY path must pass through a switch as a static connection.

A FICON (FC) Channel
An FC channel path is used to transfer data between a host system image and an FICON control unit. I/O-Ops can display data about an FC channel path and change its connectivity.

Coupling Facility Channels
I/O-Ops can display and return data about a Coupling Facility Receiver (CFR) channel and a Coupling Facility Sender (CFS) channel. However, I/O-Ops does not provide connectivity control for these types of channels.

OSA Channels
The OSA channel provides an S/390 internal channel attachment to local area networks (LANs) that support Internet protocols such as fiber distributed data interface (FDDI), token ring, Ethernet, and high speed serial interface (HSSI).

Making Connectivity Changes
You can use I/O-Ops as the single point of control for I/O connectivity changes in the active I/O configurations under its management.
In addition to an I/O-Ops base program running on a host, the host must be able to play a role with a group of hosts.

To develop safe procedures to protect your system’s connectivity, you need to understand the functional characteristics of the dynamic switches that you control with I/O-Ops. For example, the local console of Switch Directors has a setting that allows you to choose how the switch is restarted. If your operating procedures are to restart:

- From the same configuration that was most recently active on the Switch Director, use the Switch Director’s ACTIVE=SAVED setting
- From a fixed IPL file, regardless of what was most recently active on the Switch Director, you can set the Switch Director to use that file

But note that your use of I/O-Ops varies accordingly. Changes that you make through I/O-Ops are only automatically restored after events like power failures, if the Switch Directors are set to ACTIVE=SAVED. If they are not, consider using I/O-Ops functions to update each Switch Director’s IPL file in accordance with your updates directly to the active configuration.

Switch resources are self-defining. The relevant data is contained in a node descriptor (ND) of a port or an ESCON/FICON channel and in a node element descriptor (NED) of a switch device. I/O-Ops uses this data to ensure system integrity in its commands. It can also use the serial number of an ESCON/FICON resource to identify it.

When only the channel or the control unit is an ESCON unit, the non-ESCON unit can impose some limitations on I/O-Ops’s ability to ensure coordination between the logical definition of the unit and its physical presence. (Unlike ESCON units, a non-ESCON unit is not self-defining.)

**NOForce|NOCheck|Force Option Set**

With your choice among NOForce, NOCheck, and Force, you determine the stringency with which I/O-Ops forces the completion of a switching connectivity command under two conditions:

- If any of I/O-Ops’s relevant vary path requests fail.
- If I/O-Ops checked and detected the presence of a protocol converter (CV) between the I/O interface of a parallel channel operating in block mode and the port on a dynamic switch. (NOForce is required for the REMOVE CHP and RESTORE CHP commands. NOCheck has an identical effect on these 2 commands.)

**NOForce:** (IBM-supplied default) instructs I/O-Ops to send a connectivity command to the specified switch(es) only if all of the following conditions have been met:

1. The results of consensus vary-path processing results are unanimously affirmative.
2. If a CV converter has been detected in the parallel channel path, completion of the command will not potentially interrupt dynamic data transfer through the specified switch(es).
3. If the command is disrupting a CVC channel, that CHPID has been configured offline.
If you specify NOForce, you must also specify Vary. NOForce requires the UPDATE level of authorization if I/O-Ops is secured by RACF® or an equivalent program.

**NOCheck:** instructs I/O-Ops to send a connectivity command to the specified switch(es) only if all of the following conditions listed below have been met:
1. The results of consensus vary-path processing results are unanimously affirmative.
2. If the command is disrupting a CVC channel, that CHPID has been configured offline.

If you specify NOCheck, you must also specify Vary. NOCheck requires the UPDATE level of authorization if I/O-Ops is secured by RACF or an equivalent program. Commands entered at the system console automatically receive the highest, or control, authorization level.

**Force:** instructs I/O-Ops to send a connectivity command to the specified switch(es).
- If you specify Force, you must specify NOBackout.
- You cannot specify Force on a Remove Chp or Restore Chp command.
- Force requires the control level of authorization if I/O-Ops is secured by RACF or equivalent.

### Backout|Nobackout Option Set

By choosing between BBackout and NOBackout, you can determine whether I/O-Ops will attempt to reverse, or to back out, its successful vary path offline requests if at least one relevant path could not be varied offline successfully.
- If the changes could be backed out, pathing status is returned to the state before the command was processed.
- If the changes are not backed out, pathing status may be mixed. (You can enter a DISPLAY RESULTS or DISPLAY VARY command to study the resulting status.)

**BBackout (IBM-supplied default):** instructs I/O-Ops to attempt to back out the vary path offline requests that were successful under a number of error conditions, such as: requests if at least one path affected by the command could not be varied offline successfully. Some of the conditions that would require I/O-Ops to attempt to back out vary path requests are:
- At least one relevant vary path offline request was not successful. For example, a voting I/O-Ops has returned a negative vote or the time limit for consensus processing was reached before all votes were returned.
- A switch affected by the connectivity command responded negatively.
- A channel could not be configured offline successfully with a Remove Chp command.

If you select BBackout, you cannot select Force.

**NOBackout:** instructs I/O-Ops to not reverse any vary path requests that have been completed successfully during command processing or that are still in progress. You must specify NOBackout if you specify either NOVary or Force.
When a Command Fails

If you receive a message that a command failed, you first make sure I/O-Ops is running. Then, you should try to determine why the command failed, so you can correct it or report the problem to IBM.

- **Make sure I/O-Ops is started.** In MVS/ESA, to check if I/O-Ops has been started on the primary host, enter "display jobs, I/O_operations_procedure_name". If the I/O-Ops procedure name appears in the list, I/O-Ops has been started.

- **Determine why the command failed.** If a command does not complete successfully, I/O-Ops issues an error message indicating a host did not perform the command or a switch failed to perform the command, or both. Return codes and reason codes are issued when commands fail when using the API.

- **Use the DISPLAY RESULTS command.** This command displays information to help you quickly determine if the problem is host-related, switch-related, or both. A command can fail for one or more reasons, including switch-related or I/O-Ops-related problems.

The error message you receive might not provide you with sufficient information to determine the reasons why your command failed. If you need more information to understand the problem, you can display information about your configuration status to determine the specific reasons.

- **Correct the command failure.** Once you have determined the cause of a command failure and the present configuration status, you can begin to correct the problem using backout, reset, force, or other options.

Switching Commands

The members of a group (domain) must be either:

- The issuing I/O-Ops, which is the base program (primary host) at which you enter, or to which you send, the command.

- An I/O-Ops base program that shares access to at least one dynamic switch in common with the issuing I/O-Ops.

If the switching command contains Vary (path) as an implicit or explicit option, the issuing I/O-Ops broadcasts the request to the other base programs. Each base program issues the appropriate vary path requests to its own system (host) image. The results must be known by the issuing I/O-Ops within a preset time interval. After the results are in or when the time interval expires, the issuing I/O-Ops determines whether it and the voters have reached the consensus that all the vary path requests completed successfully. If so, the issuing I/O-Ops proceeds with command processing.

**Notes:**

1. Because an issuing I/O-Ops is also called the primary host, or the primary I/O-Ops host, a voter is also called a secondary host, or a secondary I/O-Ops host.

2. A voter is **active** if it has not been excluded, or reset, from voting, by the Reset Host command.

3. A voter is **inactive** if it has been excluded by a Reset Host Off command.

4. A voter is **purged** if it has been excluded by a Reset Host Purge command. To I/O-Ops, a purged voter simply does not exist.

5. The preset time interval is either the IBM-supplied default of 60 seconds or a time interval set with the Reset Timeout command.
You can change connections of ports on a switch with these commands:

- ALLOW
- BLOCK
- CONNECT
- DISCONNECT
- PROHIBIT
- UNBLOCK

The following examples help to illustrate the hierarchy of port attributes.

**PROHIBIT**

**Example**

\[ P \ (C0) \ (EA) \ 0100 \]

**Explanation**

Prohibits data from being transferred dynamically between C0 and EA. Dynamic data transfer between C0 and D4 is not affected.

![Figure 164. PROHIBIT Command Example](image)

**CONNECT**

**Example**

\[ C \ (C0) \ (EA) \ 0100 \]

**Explanation**

Statically connects C0 and EA for exclusive data transfer, despite the prohibition of a dynamic connection between these two ports. However, data cannot be transferred between C0 and D4.

![Figure 165. CONNECT Command Example](image)
**BLOCK**

**Example**

\`B (C0) 0100\`

**Explanation**

Blocks all data transfer through C0, regardless of its other connectivity. Data cannot be transferred between C0 and EA.

**Figure 166. BLOCK Command Example**

**UNBLOCK**

**Example**

\`U (C0) 0100\`

**Explanation**

Unblocks C0, so that the static connection is again in effect.

**Figure 167. UNBLOCK Command Example**

**DISCONNECT**

**Example**

\`N (C0) (EA) 0100\`

**Explanation**

Removes the static connection between C0 and EA, but data cannot be transferred between these two ports because they are still prohibited from having a dynamic connection. However, data can now be transferred between C0 and D4.
ALLOW

Example
A (C0) (EA) 0100

Explanation
Allows data to be transferred dynamically between C0 and EA.

Types Of Channels

I/O-Ops supports the following channel operational modes:
- ESCON Multiple Image Facility (EMIF) mode
- Logical Partition (LPAR) reconfigurable mode
- LPAR non-reconfigurable mode
- Basic mode

I/O-Ops can display, or return data on, all the channels known to it. For ESCON channels and for parallel channels operating in block mode, the I/O Ops program can be used to change the connectivity between these channel resources and others in their paths.

Remove (Quiesce) a Switch

When a service representative needs to have a switch quiesced and disabled in order to remove the unit, you can enter the I/O-Ops REMOVE SWITCH command to vary all the paths to and through the switch offline, including any paths to the switch (device) from the other hosts in its group and any chained paths leading to another switch. Optionally, you can specify the Disable option, so the service representative only has to turn power off to the unit. (You should not use the VARY DEV or the REMOVE DEV commands.)
I/O-Ops issues the appropriate vary path offline requests to the hosts. (It will not vary the path to the control unit port of the switch because it needs to communicate with the switch.)

The issuing I/O-Ops then sends the command “SET OFFLINE” to the switch and informs you that the command completed successfully. If the command completed unsuccessfully (for example, if a vary path request failed) you can analyze the data with the appropriate display command. If you cannot resolve the situation and the switch must be removed, you can rerun the REMOVE SWITCH command with the Force option. However, no paths will be varied offline.

For a discussion of why it sometimes isn't obvious that the status of a path or a device has changed after an operator has issued a connectivity command, see “Device and Path Status After Connectivity Commands” on page 562.

**Restore (Enable) a Switch**

When the system operator completes a maintenance procedure and turns power to the unit on, you enter the RESTORE SWITCH command and specify the switch device number.

The I/O-Ops at which you issued the command (the issuing I/O-Ops) first allocates, or attaches the switch. Then it activates the configuration for the switch that the unit is programmed to select whenever power is turned on to it. Finally, it ensures all the paths in the selected configuration are varied online.

I/O-Ops informs you the command completed successfully. (If it completed unsuccessfully, you can enter the DISPLAY RESULTS command to obtain more data.)

For a discussion of why it sometimes isn't obvious that the status of a path or a device has changed after an operator has issued a connectivity command, see “Device and Path Status After Connectivity Commands” on page 562.

**Using Port Names as a Tool in Switching Commands**

When you enter a switching command, you change the port attributes that control data transfer on the ports of the switch through which the path passes. The single switch or the pair of chained switches in a switchable path must be allocated to the issuing I/O-Ops (primary host) in the command. To facilitate the connectivity management of these switchable paths, I/O-Ops lets you assign, or write, a name of up to 24 characters to each addressable port on a switch.

A port name can consist of character strings, each of which identifies one of the resources in the path leading to or from the port. If you use the same character string in more than one port name, that character string effectively serves as a generic name.

To use a generic name instead of a full port name, you only need to mask off the irrelevant characters in the port name that precede or succeed or intervene between the generic character string you want to use. As a mask, you can place one asterisk (*) in each port name.
Chain and Unchain a Switchable Path (ESCON only)

A *chained path* is a path that passes statically (without dynamic connections) through one switch, which is called the *passthru switch* in the chain, and ends in a second, or *destination switch*. The two switches must be physically linked to each other. The ports in a chained path can be designated as the:

- **A** Port that the I/O interface of the channel is physically attached to in a CH chain and the I/O interface of the control unit is attached to in a CU chain.
- **M** Port that the A port is statically connected to in the passthru switch.
- **E** Port that ends the chain in the destination switch.
- **D** Is the destination link address of the chain as defined in the I/O definition file (IODF) or I/O control data set (IOCDS). As you can see, port D is simultaneously port E in a CU chain, but not in a CH chain.

![Diagram of Chained Paths](image)

*Figure 170. Examples of Chained Paths*

Chaining a Path (ESCON only)

1. You must define a chained path to I/O-Ops, and remove this definition when it is no longer needed.
2. The I/O-Ops that is used to define a chain or remove its definition must have dynamic access via a CNC channel to the control unit port (CUP) on both switches in the chained path. (In Directors, the CUP address is X'FE'.)
3. I/O-Ops will *not* permit you to define, or remove the definition of, a chain if it can foresee that the chain command would fail these precursory checks:
   - Remove the last path to the specified switch(es) in a Block, Connect, or Chain command
   - Affect an unimplemented port
   - Affect the control unit port (CUP) except to assign it a port name
   - Allow a port to connect dynamically to itself
   - Connect a port statically to itself or to more than one other port
   - Connect a port that is not installed.
   - Disconnect 2 ports that are not statically connected
- Chain a port that is already part of a different chain
- Chain ports so that they would create a loop

Chaining can affect the vary path requests that I/O-Ops issues. See "Processing a Vary Path Request" on page 556 for more information.

4. If you intend to define a chain that includes a parallel channel operating in block mode with a CV converter, such as an ESCON Converter Model 2 (9035) attached, use these guidelines for selecting noforce, nocheck, or force. The guidelines should help you decide whether to select NOForce, NOCheck, or Force as an option in a connectivity command that affects a CV converter.

*If the path is not chained and the foregoing requirements are met, you can enter a connectivity command with either the NOForce option or the NOCheck option.*

- If you specify NOForce, I/O-Ops will check for an operational CV converter in the path and process the command under the set of rules that apply.
  a. If the CV converter is operational, I/O-Ops will:
     1) Fail a Block, Connect, Remove Switch, or Chain command that affects the port.
     2) Fail a Prohibit command that affects the path to the CV converter.
  b. If the CV converter is not operational, I/O-Ops will:
     1) Issue a warning if an Unblock, Disconnect, RESTORE SWITCH, or Unchain command is entered that affects the port.
     2) Issue a warning if an Allow command is entered that affects the path to the CV converter.

- If you specify NOCheck, I/O-Ops will not check for a CV, and will process the command under the set of rules that apply.

*If the path is chained and the foregoing requirements are met, you must specify NOCheck or Force to unchain the path or to block any of the ports in the path.*

5. Make sure the channel I/O interface, the control unit I/O interface, and the link between the two switches are all attached properly. Ports A, M, E, and D in Figure 170 on page 551 are determined by these attachments. Also make sure the destination link address (port D) is defined properly in the IODF or IOCDS.

6. Remember that defining a chain and removing a defined chain can affect all the participants in the issuing I/O-Ops’ consensus vary-path processing.

**Two Scenarios for a CH Chain in a CNC Path**

*Configuration:*
Procedure:
VARY PATH(200,3C),OFFLINE,FORCE
CF CHP(3C),OFFLINE,FORCE
CONNECT (C1) (D4) 100
CF CHP(3C),ONLINE
VARY PATH(200,3C),ONLINE,FORCE
RESET S 200
CHAIN (F2) 200 (D4) (C1) 100

To remove the definition, enter:
UNCHAIN (F2) 200 (D4) (C1) 100 VARY FORCE NOB

The Force option is required because the Unchain command removes the last path to the CUP on the destination switch.

If you do not configure CHPID 3C offline, enter:
BLOCK (C1) 100

Configuration:
**Procedure:**

Enter from SYSA:

```
VARY PATH(200,3C),OFFLINE,FORCE
CF CHP(3C),OFFLINE,FORCE
```

Then, enter from SYSB:

```
CHAIN (F2) 200 (D4) (C1) 100
```

Finally, enter from SYSA:

```
CF CHP(3C),ONLINE
VARY PATH(200,3C),ONLINE,FORCE
RESET S 200
```

To remove the definition, enter from SYSB:

```
UNCHAIN (F2) 200 (D4) (C1) 100 VARY FORCE NOB
```

The Force option is required because the Unchain command removes the last path to the CUP on the destination switch. If you do not configure CHPID 3C offline, enter:

```
BLOCK (C1) 100
```

---

**A "Partially-ESCON" Path**

A partially-ESCON path is a path in which either the channel or the control unit in the I/O path is not an ESCON object.

To ensure system integrity when processing its switching commands, I/O-Ops relies on node descriptor (ND) or node element descriptor (NED) data from the ESCON objects involved. In a switchable partially-ESCON path, therefore, I/O-Ops cannot always ensure the same degree of system integrity that it can for "fully" ESCON paths.
An ESCON-in-Converted-Mode (CVC) Path

An ESCON converted channel (CVC) is an ESCON channel that has been initialized to operate in converted mode to allow data to be transferred between its host and a parallel control unit.

A protocol converter such as the ESCON Converter Model 1, or equivalent, must be attached to the control unit. This type of converter is transparent to I/O-Ops.

A Parallel Channel in Block (BL) Mode

Requirements for a Parallel Channel

- The parallel channel must be operating in block (BL) mode.
- The control unit in the path must be an ESCON-capable and “cooperating” control unit.
- The path can be switchable or nonswitchable.
- If the path is switchable:
  - A protocol converter, such as the ESCON Model 2 (9035), or equivalent, must be attached between the channel I/O interface and the switch.
  - The switch must be allocated to the issuing I/O-Ops.
Processing a Vary Path Request

When you have chosen Vary as a connectivity option, the issuing I/O-Ops and its active voting constituency issue the appropriate vary path requests to their host operating system images. The host images comply with the appropriate VARY PATH ON|OFF commands.

General Rules

I/O-Ops adheres to the following rules. When the vary path processing has been completed successfully, these rules are of only moderate interest to you. However, if one or more requests failed, you need to know these rules to help you analyze the cause of the failure.

- If I/O-Ops knows that the appropriate vary path requests have already been performed, it takes no action. For example, if a port is blocked, all paths are varied offline. If a Prohibit command is then entered for the blocked port, no further action is taken for that port.
- If two port attributes are to be changed with the same Writeport or Writeswch command, the hierarchy remains in effect. (See Switching Commands for more information on port hierarchy.) For example, if you specify that a port be both blocked and disconnected with the same command, I/O-Ops will perform the vary path processing needed to block that port and disconnect it.
- To keep I/O path disruption to a minimum should a command fail, I/O-Ops varies:
  - Offline before it sends a command to a switch.
  - Online after it sends a command to a switch.

A chained path adds another level of complexity to these rules. Paths through all three ports involved in a chain might be affected by the same connectivity command. For example, the command might affect the middle port’s connectivity. However, vary path processing might be required for the port on the destination switch that ends the chain.
If a CV converter is in the chained path, you must specify NOCheck or Force as a connectivity option if you want to block any of the 3 ports involved or you want to unchain the chain. (If you specify NOForce, the command will fail.)

**VTAM Application Name**

For communication among I/O-Ops base programs, I/O-Ops uses VTAM or TCP/IP, or both. To participate in this intersystem communication using VTAM, a base program must be defined as an application according to the rules for VTAM definitions. For information on these definitions, see the I/O-Ops program directory for the host operating system on which you intend to use the application.

VTAM returns the VTAM application name to a base program when it is started, thus identifying that base program. The base program uses its VTAM application name and TCP/IP host name as its *host names* to register itself to the members of its group when it allocates each dynamic switch.

**To get a list of the VTAM (application), or host names** known to an issuing I/O-Ops (primary host), enter the Display Host command as DH* or view the pages of that host's online notebook.

**To remove or restore a base program (host)** from participating in consensus vary-path processing, use the Reset Host command. (This command has three options: Off, Purge, and On.)

**TCP/IP Host Name**

For communication among I/O-Ops base programs, I/O-Ops uses VTAM and/or TCP/IP. To participate in this intersystem communication using TCP/IP, a base program must have defined the server port and optionally the client port. For information on these definitions, see the step “Perform TCP/IP Definitions for I/O Operations” in “Configuring SA z/OS on Host Systems” of *IBM Tivoli System Automation for z/OS Planning and Installation*.

**Note:** If server port is not defined the base program suppresses the TCP/IP communication.

If TCP/IP is configured to run multiple stacks (CINET) you should specify the host name on the PARM parameter to tell I/O-Ops what stack it has to use. For details on how to specify the PARM parameter see the step “Configuring I/O-Ops” in “Configuring SA z/OS on Host Systems” of *IBM Tivoli System Automation for z/OS Planning and Installation*. If you omit the parameter the base program connects to the next available stack.

TCP/IP returns its host name after the base program has connected to the stack. If the base program detects that the host name has more than 8 characters it looks for an alias of up to 8 characters. For details on how to specify an alias see the step “Perform TCP/IP Definitions for I/O Operations” in “Configuring SA z/OS on Host Systems” of *IBM Tivoli System Automation for z/OS Planning and Installation*. The base program uses its TCP/IP host name and VTAM application name as its host names to register itself to the members of its group when it allocates each dynamic switch.

**Note:** If the host name has more than 8 characters and an appropriate alias cannot be found the base program suppresses the TCP/IP communication.
To get a list of the TCP/IP (application), or host names known to an issuing I/O-Ops (primary host), enter the Display Host command as "D H *".

To remove or restore a base program (host) from participating in consensus vary-path processing, use the Reset Host command. (This command has three options: Off, Purge, and On.)
Appendix B. General Considerations for I/O Operations

Commands

I/O operations commands do not require access authorization, although it is highly advisable. For a cross-reference of the access authorization levels that they require see “Defining an RACF Profile for I/O operations” in the appendix “Security and Authorization” of IBM Tivoli System Automation for z/OS Planning and Installation.

General Information and Tips for Using Multisystem Commands

1. If you are entering an array or table, it must be fully contained in a single variable, by itself, and with no leading blanks (trailing blanks are acceptable because they will be ignored). Without this restriction, you could enter an array or table that has bytes that look like blanks, will be parsed as blanks, and will make the results unpredictable.

2. Array elements are checked for validity because they are all character data. Tables are not checked for validity because numeric data is assumed to be hexadecimal (and so can never be invalid). Tables are not subject to translation and only the RNUMs are taken from the table.

3. Host application names will be checked for validity. Valid names follow MVS member name rules.

   **Query commands only:** “HOST()” and “XSYS()” are interpreted as an error (no name is included between the beginning and ending parenthesis).

4. For table input, the responder must be the same as the host targeted to return the response.

5. A summary ROW for each host will be returned as the FIRST row for each responder host that is returning data (that is, that does not have a command start failure). This ROW is marked by RCODE=X'51000FFFF' and contains (currently) status bits.

6. For Query Interface commands (only), when the interface is not found on the entity1 switch/cu, row is marked by RCODE=X'51000016'.

7. Query Interface Switch/CntlUnit commands change their output structures to move the switch/control unit description into every row instead of having it reside in the header. With multi-host responses, there is no way to resolve which switch description would have stayed in the command header (each host could have a different description of the switch/control unit).

8. Query Interface Switch returns no rows when the switch being queried is closed. Also, Query Interface CntlUnit returns no rows when the PID for the control unit is not valid.

9. The lock owner field in the QEH command is only valid when the responder host is the same as the object/entity host. For example, if you issue a Query Entity Host Value HOST(*) SCOPE Value HOST(*) command, the QEH that is returned will contain #hosts x #hosts ROWs (because each host will return all the hosts it knows about) but the only valid lockowner fields will be where:

   QEH.ROW(i).RESPONDER.APPL_NAME = QEH.ROW(i).APPL_NAME.
General Considerations for Using the Query Entity|Interface|Relations Commands

1. **Input for Query Entity and Query Interface commands:** A Query Relation table can be input for all Query Entity and Query Interface Switch commands.

2. **Data returned:** For the Query Entity, Query Interface, and Query Relation commands, use the self-description data (header size, row size, format ID) that is returned with each output structure to parse the data.

3. **Duplicate rows:** No duplicate rows will be returned unless an array or table is used as the second entity asking for duplicate information to be returned. This implies that unless complete pathing (host-to-device or CHPID-to-device) is specified in your command, you could be getting summary data back.

   For example, assume your configuration contains CHPIDs 10, 11, and 12, each with a path to control unit 100. Additionally, control unit 100 has devices 100–107 on it.

   A **Query Relation CU 100 Chp Value *** *(tell me all CHPIDs that access device 100)* would return 3 data QRO.ROWs to describe the 3 paths (CHPID 10 to device 100, CHPID 11 to device 100 and CHPID 12 to device 100). These are examples of complete path definitions.

   If you issue **Query Relation CU 100 Dev Value 100** *(tell me whether device 100 is defined to be on CU 100)*, only 1 data QRO.ROW is returned (not 3) indicating that there is a logical relationship between the two entities but not complete details (paths) about the relation because the command did not ask for complete pathing.

4. **Multiple rows:** A single array entry could generate more than one row returned. For example, the relation between a device and a host will return as many rows as there are CHPIDs that access the given device.

5. **Rules for columns of data:** Although the same structure is returned for every Query Relation command, different columns of data are returned for the different variations of the commands. The following rules apply:
   a. **STATBITS** are set for every command.
   b. The columns between HOST and DEVICE are command-dependent; that is, Query Relation Host-Host commands return only the HOST column, Query Relation Host-Chp commands return both the HOST and CHPID columns, Query Relation Host-Switch commands return HOST, CHPID, PORTIN, SWDEVN, and LSN columns (note that STATBITS are required to validate PORTIN and SWDEVN as these might or might not be known), and so on.
   c. Like entities are not allowed except host-to-host relations, which indicate VTAM relations, and switch-to-switch relations, which indicate chaining relations.
   d. Data for a secondary host (voting I/O-Ops) will be marked INCOMPLETE because data bases are not shared among participating I/O-Ops.
   e. When no host is named in a command, the primary host (issuing I/O-Ops) is assumed.

6. **Sorting of output:** Sorting is done only if you specify Value or Range as options for the format of the output. If you specify Array or Table, I/O-Ops returns data in the same order as the input array or table.

7. **Chpid filtering of pathing information is handled as follows:**
   a. If a channel is reserved to a partition that is not the one you are running on, the channel will appear as not defined.
b. I/O-Ops only returns paths where the Chpid is not reserved to another partition.

8. **Reason codes that are related to Return codes:**
   In addition to the return codes (RCs) and reason codes that are also returned for the Display and the Query Switch commands, such as CHSC data refreshed, lock indicators, switch configurations, and so on, the following combinations are unique to the Query commands:
   - With RC = 4, a reason code of X'51001004' indicates that some host has at least one summary row bit set.
   - For RC = 8, a Getmain failure occurred.
   - With RC = 20, a reason code of X'51xxyyyy' indicates that I/O-Ops has detected an internal error. You should report this problem to IBM.

9. **Overflow conditions:** In the Query Entity, Query Interface, and Query Relation commands the header and summary rows of the output contains a bit to indicate that an overflow condition was detected in the response buffer.
   - **If you entered a Query Entity or Query Interface command:**
     Enter another (similar) command to retrieve the subset of data that has not been returned.
     For example, if the command that overflowed was **Query Entity Dev Value *** and the last device entry returned to you is (device number) 743, the next command you should issue is **Query Entity Dev Range 744-**.
     If you are using an array (or table) as input for a command, you should continue to use the array/table form of command but adjust your array/table input to begin at the (output_num_rows+1) element in your next array command.
   - **If you entered a Query Relation command:**
     When translating your original command to a new format due to an overflow return/reason code, you might need to begin your new command with the last value returned for the second entity (not the one after the last value returned) or some pathing information could be lost.
     For example, assume you have the following partial configuration:
     ```
     Host = IHVAPPL1
     ChPIDs = 1,2,3,4,5
     CntlUnits = 100 (accessed by ChPIDs 1,2,3), 200 (accessed by ChPIDs 3,4,5)
     Devices = 100-10F (off CU 100), 200-20F (off CU 200)
     ```
     Assume (for the purposes of this example only) that 7 is the maximum number of rows that can be returned with any Query Relation command (in reality this number is more on the order of 2700).
     You issue **Query Relation Host IHVAPPL1 Dev Value *** to request all the IOCDS paths defined for this host.
     I/O-Ops returns 7 rows describing paths (1,100), (2,100), (3,100), (1,101), (2,101), (3,101) and (1,102) with RC and RSN indicating overflow and the MORE_DATA bit in the output table set.
     Because you only received 1 of 3 rows (paths) back for device 102, you must now use the command **Query Relation Host IHVAPPL1 Dev Range 102-** to retrieve the pathing information for this host.

10. **Chaining:** Chaining is indicated in an AEAM for a port by virtue of the log_class=switch (port) and the LOGICAL bit being set.
    If the path is chained, the switch that appears in the row for a path is either:
    - a. The dynamic switch in the chain, or (if there is no dynamic switch)
b. The switch that the CHPID for the path is defined (in IOCDS) to be on (by the SWITCH= parameter).

11. “Physical” rules with respect to the EAM/AEAM are:
   a. EAM.LOGICAL or AEAM.LOGICAL = 1 (binary) means the entity is defined in the IOCDS. Additionally, Chain commands are considered logical definition data so they are also included in “logical” data.
   b. PHYSICAL (current, history and not valid) means “according to link level data” (node element descriptors [NEDs] or node descriptors [NDs]).
   c. Physical OTHERS bits (and NDs) are set when the interface for an entity being queried does not have the same validity as the entity itself (as represented in the physical ID). OTHER_NDs are validated by AEAM bits (OTHERS bits are never set in EAMs). The OTHER_ND that is returned contains a ND for the attached entity that is causing the PID to be “more valid” than the interface being queried.
   d. Logical OTHER bit (and value) is set when the interface for an entity being queried does not have an IOCDS path defined to it on the interface being queried (that is, a physical-only path to the entity) but the entity is defined in the IOCDS and is accessible by other paths. The OTHER_LOG field will be set to the smallest logical value for the entity (for example, if control units 100 and 200 are hung off a port in a physical-only relationship but are defined to have other (IOCDS) paths elsewhere, the LOG_OTHER bit would be set for the (QIS) port’s AEAM and the OTHER_LOG would contain 100).

12. P_AMB, when on, means at least one of the following:
   a. All the (IOCP/IODF) defined interfaces to the control unit do not have the same WWID (only valid NDs are checked).
   b. For switches only: Even if all the WWIDs in the interfaces for the CNTL match, if the CNTLPID.WWID is different from the SWITPID.WWID (which is obtained from the NED), the CNTLPID is marked P_AMB and remains REFLECTED.
   c. For CHPIDs only: When the attached ND’s WWID is not the same as the PID.WWID of the switch that it is supposed to be attached to.

---

**Device and Path Status After Connectivity Commands**

This section describes the results of I/O-Ops connectivity commands such as PROHIBIT/ALLOW, BLOCK/UNBLOCK, and REMOVE/RESTORE CHP/SWITCH. Sometimes it isn’t obvious why the status of a path or a device has not changed after an operator has issued one of these connectivity commands.

Generally, if the status of a path or device is OFFLINE before a PROHIBIT, BLOCK, or REMOVE command is issued, the status will not be changed when the opposite command is issued afterwards.

This is also true for the ALLOW, UNBLOCK, and RESTORE commands if the OFFLINE status of a path or device was not the result of a PROHIBIT, BLOCK, or REMOVE command.

The status of a path is also OFFLINE even if the path was ONLINE before issuing the connectivity commands when all of the following conditions are true:
- The path’s device is attached to a cascaded switch.
- The device of the path’s entry switch is OFFLINE, which makes the switch unknown to I/O-Ops.
The destination switch has been manipulated by two connectivity commands resulting in varying the path to the device first OFFLINE and then ONLINE.

**Note:** MVS handles the device status after varying ONLINE the very first path to a device differently, depending on the status that the device had before the last path to the device was varied OFFLINE:

- If the device status is ONLINE when the last path is forced to OFFLINE, the first path to the device that becomes ONLINE will also vary the device ONLINE.
- If the device status is OFFLINE when the last path is varied OFFLINE, the device is kept OFFLINE regardless of how many paths become ONLINE.

Table 5 shows a sample that uses the REMOVE SWITCH and RESTORE SWITCH commands to demonstrate how the status of a device and its paths are affected by an I/O-Ops connectivity command.

<table>
<thead>
<tr>
<th>System KEY3</th>
<th>System KEY4</th>
</tr>
</thead>
<tbody>
<tr>
<td>D M=DEV(297F)</td>
<td>D M=DEV(297F)</td>
</tr>
<tr>
<td>IEE174I 13.13.37 DISPLAY M 033</td>
<td>IEE174I 13.04.24 DISPLAY M 033</td>
</tr>
<tr>
<td>DEVICE 297F STATUS=ONLINE</td>
<td>DEVICE 297F STATUS=OFFLINE</td>
</tr>
<tr>
<td>CHP</td>
<td>CHP</td>
</tr>
<tr>
<td>AE</td>
<td>AE</td>
</tr>
<tr>
<td>ENTRY LINK ADDRESS 613C 6420</td>
<td>ENTRY LINK ADDRESS 613C 6420</td>
</tr>
<tr>
<td>DEST LINK ADDRESS 631C 6420</td>
<td>DEST LINK ADDRESS 631C 6420</td>
</tr>
<tr>
<td>PATH ONLINE Y N</td>
<td>PATH ONLINE Y N</td>
</tr>
<tr>
<td>CHP PHYSICALLY ONLINE Y Y</td>
<td>CHP PHYSICALLY ONLINE Y Y</td>
</tr>
<tr>
<td>PATH OPERATIONAL Y Y</td>
<td>PATH OPERATIONAL Y Y</td>
</tr>
<tr>
<td>MANAGED N N</td>
<td>MANAGED N N</td>
</tr>
<tr>
<td>DEVICE 297F STATUS=OFFLINE</td>
<td>DEVICE 297F STATUS=OFFLINE</td>
</tr>
<tr>
<td>CHP</td>
<td>CHP</td>
</tr>
<tr>
<td>AE</td>
<td>AE</td>
</tr>
<tr>
<td>ENTRY LINK ADDRESS 6257 6280</td>
<td>ENTRY LINK ADDRESS 6257 6280</td>
</tr>
<tr>
<td>DEST LINK ADDRESS 613C 6420</td>
<td>DEST LINK ADDRESS 613C 6420</td>
</tr>
<tr>
<td>PATH ONLINE Y Y</td>
<td>PATH ONLINE Y Y</td>
</tr>
<tr>
<td>CHP PHYSICALLY ONLINE Y Y</td>
<td>CHP PHYSICALLY ONLINE Y Y</td>
</tr>
<tr>
<td>PATH OPERATIONAL Y Y</td>
<td>PATH OPERATIONAL Y Y</td>
</tr>
<tr>
<td>MANAGED N N</td>
<td>MANAGED N N</td>
</tr>
<tr>
<td>U NUMBER 297C297F</td>
<td>U NUMBER 297C297F</td>
</tr>
</tbody>
</table>

Appendix B. General Considerations for I/O Operations Commands
### Table 5. Device and Path Status Before and After the REMOVE SWITCH and RESTORE SWITCH Commands (continued)

<table>
<thead>
<tr>
<th>System KEYS</th>
<th>System KEYS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device 297C</td>
<td>Device 297C</td>
</tr>
<tr>
<td>Status=OFFLINE</td>
<td>Status=OFFLINE</td>
</tr>
<tr>
<td>Device 297D</td>
<td>Device 297D</td>
</tr>
<tr>
<td>Status=ONLINE</td>
<td>Status=ONLINE</td>
</tr>
<tr>
<td>Device 297E</td>
<td>Device 297E</td>
</tr>
<tr>
<td>Status=OFFLINE</td>
<td>Status=ONLINE</td>
</tr>
<tr>
<td>Device 297F</td>
<td>Device 297F</td>
</tr>
<tr>
<td>Status=OFFLINE</td>
<td>Status=OFFLINE</td>
</tr>
</tbody>
</table>

THE FOLLOWING DEVICE PATHS ARE ONLINE THROUGH THIS PORT:

- **ATTACHED NODE = 002107.9A2.IBM.75.000000074011**
- **DCM STATUS=OFFLINE**
- **PATH(297F,AA) ONLINE**
- **PATH(297F,AB) ONLINE**
- **PATH(297C,AB) ONLINE**
- **PATH(297E,AB) ONLINE**

**IHVD344I Control unit 1013 status update.**

**IHVA201I Switch 1013 Port FE state changed.**

**IHVA201I Switch 1013 Port 05 state changed.**

**IEF237I 1013 ALLOCATED TO SWIT1013**

**IEF236I ALLOC. FOR IOZPS33H E**

**IHVD344I Control unit 1013 status update.**

**IHVA201I Switch 1013 Port FE state changed.**

**IHVA201I Switch 1013 Port 05 state changed.**

**IEF237I 1013 ALLOCATED TO SWIT1013**

**IEF236I ALLOC. FOR IOZPS33H E**

**IHVD344I Control unit 1013 status update.**

**IHVA201I Switch 1013 Port FE state changed.**

**IHVA201I Switch 1013 Port 05 state changed.**

**IEF237I 1013 ALLOCATED TO SWIT1013**

**IEF236I ALLOC. FOR IOZPS33H E**

**IHVA201I Switch 1013 Port FE state changed.**

**IHVA201I Switch 1013 Port 05 state changed.**

**IEF237I 1013 ALLOCATED TO SWIT1013**

**IEF236I ALLOC. FOR IOZPS33H E**

**IHVA201I Switch 1013 Port FE state changed.**

**IHVA201I Switch 1013 Port 05 state changed.**

**IEF237I 1013 ALLOCATED TO SWIT1013**

**IEF236I ALLOC. FOR IOZPS33H E**

**IHVA201I Switch 1013 Port FE state changed.**

**IHVA201I Switch 1013 Port 05 state changed.**

**IEF237I 1013 ALLOCATED TO SWIT1013**

**IEF236I ALLOC. FOR IOZPS33H E**

**IHVA201I Switch 1013 Port FE state changed.**

**IHVA201I Switch 1013 Port 05 state changed.**
### Table 5. Device and Path Status Before and After the REMOVE SWITCH and RESTORE SWITCH Commands (continued)

<table>
<thead>
<tr>
<th>System KEYS</th>
<th>System KEYS</th>
</tr>
</thead>
<tbody>
<tr>
<td>D-M-DEV(297)</td>
<td>D-M-DEV(297)</td>
</tr>
<tr>
<td>00177413 41 32 DISPLAY M 356</td>
<td>00177413 41 32 DISPLAY M 356</td>
</tr>
<tr>
<td>DEVICE 297</td>
<td>DEVICE 297</td>
</tr>
<tr>
<td>STATUS-ONLINE</td>
<td>STATUS-ONLINE</td>
</tr>
<tr>
<td>CHP</td>
<td>CHP</td>
</tr>
<tr>
<td>AE AB</td>
<td>AE AB</td>
</tr>
<tr>
<td>ENTRY LINK ADDRESS 6257 6280</td>
<td>ENTRY LINK ADDRESS 6257 6280</td>
</tr>
<tr>
<td>DEST LINK ADDRESS 613C 6420</td>
<td>DEST LINK ADDRESS 613C 6420</td>
</tr>
<tr>
<td>PATH ONLINE</td>
<td>PATH ONLINE</td>
</tr>
<tr>
<td>N Y</td>
<td>N Y</td>
</tr>
<tr>
<td>CHP PHYSICALLY ONLINE</td>
<td>CHP PHYSICALLY ONLINE</td>
</tr>
<tr>
<td>Y Y</td>
<td>Y Y</td>
</tr>
<tr>
<td>MANNED</td>
<td>MANNED</td>
</tr>
<tr>
<td>N N</td>
<td>N N</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>D-PATH(297 AE)-OFFLINE FORCE</th>
<th>D-PATH(297 AE)-OFFLINE FORCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHP</td>
<td>CHP</td>
</tr>
<tr>
<td>AE AB</td>
<td>AE AB</td>
</tr>
<tr>
<td>ENTRY LINK ADDRESS</td>
<td>ENTRY LINK ADDRESS</td>
</tr>
<tr>
<td>6257 6280</td>
<td>6257 6280</td>
</tr>
<tr>
<td>DEST LINK ADDRESS</td>
<td>DEST LINK ADDRESS</td>
</tr>
<tr>
<td>613C 6420</td>
<td>613C 6420</td>
</tr>
<tr>
<td>PATH ONLINE</td>
<td>PATH ONLINE</td>
</tr>
<tr>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>CHP PHYSICALLY ONLINE</td>
<td>CHP PHYSICALLY ONLINE</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>PATH OPERATIONAL</td>
<td>PATH OPERATIONAL</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>MANAGED</td>
<td>MANAGED</td>
</tr>
<tr>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>D-PATH(297 AE)-OFFLINE FORCE</th>
<th>D-PATH(297 AE)-OFFLINE FORCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHP</td>
<td>CHP</td>
</tr>
<tr>
<td>AE AB</td>
<td>AE AB</td>
</tr>
<tr>
<td>ENTRY LINK ADDRESS</td>
<td>ENTRY LINK ADDRESS</td>
</tr>
<tr>
<td>6257 6280</td>
<td>6257 6280</td>
</tr>
<tr>
<td>DEST LINK ADDRESS</td>
<td>DEST LINK ADDRESS</td>
</tr>
<tr>
<td>613C 6420</td>
<td>613C 6420</td>
</tr>
<tr>
<td>PATH ONLINE</td>
<td>PATH ONLINE</td>
</tr>
<tr>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>CHP PHYSICALLY ONLINE</td>
<td>CHP PHYSICALLY ONLINE</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>PATH OPERATIONAL</td>
<td>PATH OPERATIONAL</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>MANAGED</td>
<td>MANAGED</td>
</tr>
<tr>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>D-PATH(297 AE)-OFFLINE FORCE</th>
<th>D-PATH(297 AE)-OFFLINE FORCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHP</td>
<td>CHP</td>
</tr>
<tr>
<td>AE AB</td>
<td>AE AB</td>
</tr>
<tr>
<td>ENTRY LINK ADDRESS</td>
<td>ENTRY LINK ADDRESS</td>
</tr>
<tr>
<td>6257 6280</td>
<td>6257 6280</td>
</tr>
<tr>
<td>DEST LINK ADDRESS</td>
<td>DEST LINK ADDRESS</td>
</tr>
<tr>
<td>613C 6420</td>
<td>613C 6420</td>
</tr>
<tr>
<td>PATH ONLINE</td>
<td>PATH ONLINE</td>
</tr>
<tr>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>CHP PHYSICALLY ONLINE</td>
<td>CHP PHYSICALLY ONLINE</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>PATH OPERATIONAL</td>
<td>PATH OPERATIONAL</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>MANAGED</td>
<td>MANAGED</td>
</tr>
<tr>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>D-PATH(297 AE)-OFFLINE FORCE</th>
<th>D-PATH(297 AE)-OFFLINE FORCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHP</td>
<td>CHP</td>
</tr>
<tr>
<td>AE AB</td>
<td>AE AB</td>
</tr>
<tr>
<td>ENTRY LINK ADDRESS</td>
<td>ENTRY LINK ADDRESS</td>
</tr>
<tr>
<td>6257 6280</td>
<td>6257 6280</td>
</tr>
<tr>
<td>DEST LINK ADDRESS</td>
<td>DEST LINK ADDRESS</td>
</tr>
<tr>
<td>613C 6420</td>
<td>613C 6420</td>
</tr>
<tr>
<td>PATH ONLINE</td>
<td>PATH ONLINE</td>
</tr>
<tr>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>CHP PHYSICALLY ONLINE</td>
<td>CHP PHYSICALLY ONLINE</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>PATH OPERATIONAL</td>
<td>PATH OPERATIONAL</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>MANAGED</td>
<td>MANAGED</td>
</tr>
<tr>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>
Table 5. Device and Path Status Before and After the REMOVE SWITCH and RESTORE SWITCH Commands (continued)

<table>
<thead>
<tr>
<th>Device</th>
<th>297C</th>
<th>297D</th>
<th>297E</th>
<th>297F</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>KEY3</td>
<td>KEY4</td>
<td>KEY3</td>
<td>KEY4</td>
</tr>
<tr>
<td>Status before command REMOVE SWITCH 1013</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Device</td>
<td>ON</td>
<td>OFF</td>
<td>ON</td>
<td>OFF</td>
</tr>
<tr>
<td>Path AE (1013,3C)</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
</tr>
</tbody>
</table>
### Table 6. Path and Device Status Summary (continued)

<table>
<thead>
<tr>
<th>Device</th>
<th>297C</th>
<th>297D</th>
<th>297E</th>
<th>297F</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>KEY3</td>
<td>KEY4</td>
<td>KEY3</td>
<td>KEY4</td>
</tr>
<tr>
<td>Path AB (1010,20)</td>
<td>ON</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
</tr>
</tbody>
</table>

**Status after command REMOVE SWITCH 1013**

<table>
<thead>
<tr>
<th>Device</th>
<th>297C</th>
<th>297D</th>
<th>297E</th>
<th>297F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path AE (1013,3C)</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
</tr>
<tr>
<td>Path AB (1010,20)</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
</tr>
</tbody>
</table>

**Status after command RESTORE SWITCH 1013**

<table>
<thead>
<tr>
<th>Device</th>
<th>297C</th>
<th>297D</th>
<th>297E</th>
<th>297F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path AE (1013,3C)</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
</tr>
<tr>
<td>Path AB (1010,20)</td>
<td>ON</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
</tr>
</tbody>
</table>

**Status before command REMOVE SWITCH 1010**

<table>
<thead>
<tr>
<th>Device</th>
<th>297C</th>
<th>297D</th>
<th>297E</th>
<th>297F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path AE (1013,3C)</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
</tr>
<tr>
<td>Path AB (1010,20)</td>
<td>ON</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
</tr>
</tbody>
</table>

**Status after command REMOVE SWITCH 1010**

<table>
<thead>
<tr>
<th>Device</th>
<th>297C</th>
<th>297D</th>
<th>297E</th>
<th>297F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path AE (1013,3C)</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>Path AB (1010,20)</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
</tr>
</tbody>
</table>

**Status after command RESTORE SWITCH 1010**

<table>
<thead>
<tr>
<th>Device</th>
<th>297C</th>
<th>297D</th>
<th>297E</th>
<th>297F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path AE (1013,3C)</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
</tr>
<tr>
<td>Path AB (1010,20)</td>
<td>ON</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
</tr>
</tbody>
</table>

The device status changed from ONLINE to OFFLINE after forcing the last path to OFFLINE, yielding the same status as on KEY4.
Appendix C. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user’s responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Websites are provided for convenience only and do not in any manner serve as an endorsement of those Websites. The materials at those Websites are not part of the materials for this IBM product and use of those Websites is at your own risk.
IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Deutschland Research & Development GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

**Programming Interface Information**

This publication documents information that is not intended to be used as a programming interface of IBM Tivoli System Automation for z/OS.

**Trademarks**

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at www.ibm.com/legal/copytrade.shtml.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.
Glossary

This glossary includes terms and definitions from:

- The *Information Technology Vocabulary* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.

The following cross-references are used in this glossary:

- **Contrast with.** This refers to a term that has an opposed or substantively different meaning.
- **Deprecated term for.** This indicates that the term should not be used. It refers to a preferred term, which is defined in its proper place in the glossary.
- **See.** This refers the reader to multiple-word terms in which this term appears.
- **See also.** This refers the reader to terms that have a related, but not synonymous, meaning.
- **Synonym for.** This indicates that the term has the same meaning as a preferred term, which is defined in the glossary.
- **Synonymous with.** This is a backward reference from a defined term to all other terms that have the same meaning.

A

**ACF**  See [automation configuration file](#).

**ACF/NCP**  Advanced Communications Function for the Network Control Program. See [Advanced Communications Function](#) and [Network Control Program](#).

**ACF/VTAM**  Advanced Communications Function for the Virtual Telecommunications Access Method. Synonym for VTAM. See [Advanced Communications Function](#) and [Virtual Telecommunications Access Method](#).

**active monitoring**  In SA z/OS automation control file, the acquiring of resource status information by soliciting such information at regular, user-defined intervals. See also [passive monitoring](#).

**adapter**  Hardware card that enables a device, such as a workstation, to communicate with another device, such as a monitor, a printer, or some other I/O device.

**adjacent hosts**  Systems connected in a peer relationship using adjacent NetView sessions for purposes of monitoring and control.

**adjacent NetView**  In SA z/OS, the system defined as the communication path between two SA z/OS systems that do not have a direct link. An adjacent NetView is used for message forwarding and as a communication link between two SA z/OS systems. For example, the adjacent NetView is used when sending responses from a focal point to a remote system.

**Advanced Communications Function (ACF)**  A group of IBM licensed programs (principally VTAM, TCAM, NCP, and SSP) that use the concepts of Systems Network Architecture (SNA), including distribution of function and resource sharing.

**advanced program-to-program communication (APPC)**  A set of inter-program communication services that support cooperative transaction processing in a Systems...
Network Architecture (SNA) network.
APPC is the implementation, on a given system, of SNA's logical unit type 6.2.

**Advanced Workload Analysis Reporter (zAware)**
IBM analytics appliance which requires an exclusive user partition as runtime environment. The appliance is a combination of a LPAR running in a special mode, an embedded OS (Linux on System Z) and an application layer. Customers can use the appliance to monitor the console message streams of other LPARs running in the same System z cluster and create trend reports. Exploiting zAware and these trend reports can help to better predict OS outages or performance degradations and initiate proactive clusters.

**alert**
In SNA, a record sent to a system problem management focal point or to a collection point to communicate the existence of an alert condition.

In NetView, a high-priority event that warrants immediate attention. A database record is generated for certain event types that are defined by user-constructed filters.

**alert condition**
A problem or impending problem for which some or all of the process of problem determination, diagnosis, and resolution is expected to require action at a control point.

**alert focal-point system**
See [NPDA focal point system](#).

**alert threshold**
An application or volume service value that determines the level at which SA z/OS changes the associated icon in the graphical interface to the alert color. SA z/OS may also issue an alert. See [warning threshold](#).

**AMC**
See [Automation Manager Configuration](#).

**American Standard Code for Information Interchange (ASCII)**
A standard code used for information exchange among data processing systems, data communication systems, and associated equipment. ASCII uses a coded character set consisting of 7-bit coded characters (8-bit including parity check). The ASCII set consists of control characters and graphic characters. See also [Extended Binary Coded Decimal Interchange Code](#).

**APF**
See [authorized program facility](#).

**API**
See [application programming interface](#).

**APPC**
See [advanced program-to-program communication](#).

**application**
In SA z/OS, applications refer to z/OS subsystems, started tasks, or jobs that are automated and monitored by SA z/OS. On SNMP-capable processors, application can be used to refer to a subsystem or process.

**Application entry**
A construct, created with the customization dialogs, used to represent and contain policy for an application.

**application group**
A named set of applications. An application group is part of an SA z/OS enterprise definition and is used for monitoring purposes.

**application program**
A program written for or by a user that applies to the user's work, such as a program that does inventory or payroll.

A program used to connect and communicate with stations in a network, enabling users to perform application-oriented activities.

**application programming interface (API)**
An interface that allows an application program that is written in a high-level language to use specific data or functions of the operating system or another program.

**ApplicationGroup entry**
A construct, created with the customization dialogs, used to represent and contain policy for an application group.

**ARM**
See [automatic restart management](#).

**ASCB**
Address space control block.

**ASCB status**
An application status derived by SA z/OS running a routine (the ASCB...
checker) that searches the z/OS address space control blocks (ASCBs) for address spaces with a particular job name. The job name used by the ASCB checker is the job name defined in the customization dialog for the application.

ASCII  See American Standard Code for Information Interchange

ASF  See automation status file

authorized program facility (APF)
A facility that permits identification of programs that are authorized to use restricted functions.

automated console operations (ACO)
The use of an automated procedure to replace or simplify the action that an operator takes from a console in response to system or network events.

automated function
SA z/OS automated functions are automation operators, NetView autotasks that are assigned to perform specific automation functions. However, SA z/OS defines its own synonyms, or automated function names, for the NetView autotasks, and these function names are referred to in the sample policy databases provided by SA z/OS. For example, the automation operator AUTBASE corresponds to the SA z/OS automated function BASEOPER.

automatic restart management (ARM)
A z/OS recovery function that improves the availability of specified subsystems and applications by automatically restarting them under certain circumstances. Automatic restart management is a function of the Cross-System Coupling Facility (XCF) component of z/OS.

automatic restart management element name
In MVS 5.2 or later, z/OS automatic restart management requires the specification of a unique sixteen character name for each address space that registers with it. All automatic restart management policy is defined in terms of the element name, including SA z/OS’s interface with it.

automation
The automatic initiation of actions in response to detected conditions or events.

SA z/OS provides automation for z/OS applications, z/OS components, and remote systems that run z/OS. SA z/OS also provides tools that can be used to develop additional automation.

automation agent
In SA z/OS, the automation function is split up between the automation manager and the automation agents. The observing, reacting and doing parts are located within the NetView address space, and are known as the automation agents. The automation agents are responsible for:
• Recovery processing
• Message processing
• Active monitoring; they propagate status changes to the automation manager

automation configuration file
The SA z/OS customization dialogs must be used to build the automation configuration file. It consists of:
• The automation manager configuration file (AMC)
• The NetView automation table (AT)
• The NetView message revision table (MRT)
• The MPFLSTxx member

automation control file (ACF)
In SA z/OS, a file that contains system-level automation policy information. There is one master automation control file for each NetView system that SA z/OS is installed on. Additional policy information and all resource status information is contained in the policy database (PDB). The SA z/OS customization dialogs must be used to build the automation control files. They must not be edited manually.

automation flags
In SA z/OS, the automation policy settings that determine the operator functions that are automated for a resource and the times during which automation is active. When SA z/OS is running, automation is controlled by automation flag policy settings and override settings (if any) entered by the operator. Automation flags are set using the customization dialogs.
In SA z/OS, the automation function is split up between the automation manager and the automation agents. The coordination, decision making and controlling functions are processed by each sysplex's automation manager.

The automation manager contains a model of all of the automated resources within the sysplex. The automation agents feed the automation manager with status information and perform the actions that the automation manager tells them to.

The automation manager provides sysplex-wide automation.

Automation Manager Configuration

The Automation Manager Configuration file (AMC) contains an image of the automated systems in a sysplex or of a standalone system. See also automation configuration file.

Automation NetView

In SA z/OS the NetView that performs routine operator tasks with command procedures or uses other ways of automating system and network management, issuing automatic responses to messages and management services units.

Automation operator

NetView automation operators are NetView autotasks that are assigned to perform specific automation functions. See also automated function. NetView automation operators may receive messages and process automation procedures. There are no logged-on users associated with automation operators. Each automation operator is an operating system task and runs concurrently with other NetView tasks. An automation operator could be set up to handle JES2 messages that schedule automation procedures, and an automation statement could route such messages to the automation operator. Similar to operator station task. SA z/OS message monitor tasks and target control tasks are automation operators.

Automation policy

The policy information governing automation for individual systems. This includes automation for applications, z/OS subsystems, z/OS data sets, and z/OS components.

Automation policy settings

The automation policy information contained in the automation control file. This information is entered using the customization dialogs. You can display or modify these settings using the customization dialogs.

Automation procedure

A sequence of commands, packaged as a NetView command list or a command processor written in a high-level language. An automation procedure performs automation functions and runs under NetView.

Automation routines

In SA z/OS, a set of self-contained automation routines that can be called from the NetView automation table, or from user-written automation procedures.

Automation status file (ASF)

In SA z/OS, a file containing status information for each automated subsystem, component or data set. This information is used by SA z/OS automation when taking action or when determining what action to take. In Release 2 and above of AOC/MVS, status information is also maintained in the operational information base.

Automation table (AT)

See NetView automation table.

Autotask

A NetView automation task that receives messages and processes automation procedures. There are no logged-on users associated with autotasks. Each autotask is an operating system task and runs concurrently with other NetView tasks. An autotask could be set up to handle JES2 messages that schedule automation procedures, and an automation statement could route such messages to the autotasks. Similar to operator station task. SA z/OS message monitor tasks and target control tasks are autotasks. Also called automation operator.

Available

In VTAM programs, pertaining to a
logical unit that is active, connected, enabled, and not at its session limit.

**Base Control Program (BCP)**
A program that provides essential services for the MVS and z/OS operating systems. The program includes functions that manage system resources. These functions include input/output, dispatch units of work, and the z/OS UNIX System Services kernel. See also Multiple Virtual Storage and z/OS.

**basic mode**
A central processor mode that does not use logical partitioning. Contrast with logically partitioned mode.

**BCP** See Base Control Program.

**BCP Internal Interface**
Processor function of System z processor families. It allows for communication between basic control programs such as z/OS and the processor support element in order to exchange information or to perform processor control functions. Programs using this function can perform hardware operations such as ACTIVATE or SYSTEM RESET.

**beaconing**
The repeated transmission of a frame or messages (beacon) by a console or workstation upon detection of a line break or outage.

**blade**
A hardware unit that provides application-specific services and components. The consistent size and shape (or form factor) of each blade allows it to fit in a BladeCenter chassis.

**BladeCenter chassis**
A modular chassis that can contain multiple blades, allowing the individual blades to share resources such as management, switch, power, and blower modules.

**BookManager®**
An IBM product that lets users view softcopy documents on their workstations.

**C**

**central processor (CP)**
The part of the computer that contains the sequencing and processing facilities for instruction execution, initial program load (IPL), and other machine operations.

**central processor complex (CPC)**
A physical collection of hardware that consists of central storage, (one or more) central processors, (one or more) timers, and (one or more) channels.

**central site**
In a distributed data processing network, the central site is usually defined as the focal point for alerts, application design, and remote system management tasks such as problem management.

**CFR/CFS and ISC/ISR**
I/O operations can display and return data about integrated system channels (ISC) connected to a coupling facility and coupling facility receiver (CFR) channels and coupling facility sender (CFS) channels.

**channel**
A path along which signals can be sent; for example, data channel, output channel. See also link.

**channel path identifier**
A system-unique value assigned to each channel path.

**channel-attached**
Attached directly by I/O channels to a host processor (for example, a channel-attached device).

Attached to a controlling unit by cables, rather than by telecommunication lines. Contrast with link-attached. Synonymous with local.

**CHPID**
In SA z/OS, channel path ID; the address of a channel.

**CHPID port**
A label that describes the system name, logical partitions, and channel paths.

**CI** See console integration.

**CICS/VS**
Customer Information Control System for Virtual Storage. See Customer Information Control System.

**CLIST** See command list.

**clone**
A set of definitions for application.
instances that are derived from a basic application definition by substituting a number of different system-specific values into the basic definition.

**clone ID**
A generic means of handling system-specific values such as the MVS SYSCLONE or the VTAM subarea number. Clone IDs can be substituted into application definitions and commands to customize a basic application definition for the system that it is to be instantiated on.

**CNC**
A channel path that transfers data between a host system image and an ESCON control unit. It can be point-to-point or switchable.

**command**
A request for the performance of an operation or the execution of a particular program.

**command facility**
The component of NetView that is a base for command processors that can monitor, control, automate, and improve the operation of a network. The successor to NCCF.

**command list (CLIST)**
A list of commands and statements, written in the NetView command list language or the REXX language, designed to perform a specific function for the user. In its simplest form, a command list is a list of commands. More complex command lists incorporate variable substitution and conditional logic, making the command list more like a conventional program. Command lists are typically interpreted rather than being compiled.

In SA z/OS, REXX command lists that can be used for automation procedures.

**command procedure**
In NetView, either a command list or a command processor.

**command processor**
A module designed to perform a specific function. Command processors, which can be written in assembler or a high-level language (HLL), are issued as commands.

**command processor control block**
An I/O operations internal control block that contains information about the command being processed.

**Command Tree/2**
An OS/2-based program that helps you build commands on an OS/2 window, then routes the commands to the destination you specify (such as a 3270 session, a file, a command line, or an application program). It provides the capability for operators to build commands and route them to a specified destination.

**common commands**
The SA z/OS subset of the CPC operations management commands.

**Common User Access (CUA) architecture**
Guidelines for the dialog between a human and a workstation or terminal.

**communication controller**
A type of communication control unit whose operations are controlled by one or more programs stored and executed in the unit or by a program executed in a processor to which the controller is connected. It manages the details of line control and the routing of data through a network.

**communication line**
Deprecated term for telecommunication line.

**connectivity view**
In SA z/OS, a display that uses graphic images for I/O devices and lines to show how they are connected.

**console automation**
The process of having NetView facilities provide the console input usually handled by the operator.

**console connection**
In SA z/OS, the 3270 or ASCII (serial) connection between a PS/2 computer and a target system. Through this connection, the workstation appears (to the target system) to be a console.

**console integration (CI)**
A hardware facility that if supported by an operating system, allows operating system messages to be transferred through an internal hardware interface for
display on a system console. Conversely, it allows operating system commands entered at a system console to be transferred through an internal hardware interface to the operating system for processing.

**consoles**
Workstations and 3270-type devices that manage your enterprise.

**Control units**
Hardware units that control I/O operations for one or more devices. You can view information about control units through I/O operations, and can start or stop data going to them by blocking and unblocking ports.

**controller**
A unit that controls I/O operations for one or more devices.

**converted mode (CVC)**
A channel operating in converted (CVC) mode transfers data in blocks and a CBY channel path transfers data in bytes. Converted CVC or CBY channel paths can communicate with a parallel control unit. This resembles a point-to-point parallel path and dedicated connection, regardless whether the path passes through a switch.

**couple data set**
A data set that is created through the XCF couple data set format utility and, depending on its designated type, is shared by some or all of the z/OS systems in a sysplex. See also **sysplex**, **couple data set**, and **XCF couple data set**.

**coupling facility**
The hardware element that provides high-speed caching, list processing, and locking functions in a sysplex.

**CP**
See **central processor**.

**CPC**
See **central processor complex**.

**CPC operations management commands**
A set of commands and responses for controlling the operation of System/390 CPCs.

**CPC subset**
All or part of a CPC. It contains the minimum resource to support a single control program.

**CPCB**
See **command processor control block**.

**CPU**
Central processing unit. Deprecated term for **processor**.

**cross-system coupling facility (XCF)**
A component of z/OS that provides functions to support cooperation between authorized programs running within a sysplex.

**CTC**
The channel-to-channel (CTC) channel can communicate with a CTC on another host for intersystem communication.

**Customer Information Control System (CICS)**
A general-purpose transactional program that controls online communication between terminal users and a database for a large number of end users on a real-time basis.

**customization dialogs**
The customization dialogs are an ISPF application. They are used to customize the enterprise policy, like, for example, the enterprise resources and the relationships between resources, or the automation policy for systems in the enterprise. How to use these dialogs is described in IBM Tivoli System Automation for z/OS Customizing and Programming.

**CVC**
See **converted mode**.

**D**

**DataPower X150z**
See IBM Websphere DataPower Integration Appliance X150 for zEnterprise (DataPower X150z).

**DASD**
See **direct access storage device**.

**data services task (DST)**
The NetView subtask that gathers, records, and manages data in a VSAM file or a network device that contains network management information.

**data set**
The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

**data set members**
Members of partitioned data sets that are individually named elements of a larger file that can be retrieved by name.

**DBCS**
See **double-byte character set**.

---

**Display on a system console. Conversely, it allows operating system commands entered at a system console to be transferred through an internal hardware interface to the operating system for processing.**

**Consoles**
Workstations and 3270-type devices that manage your enterprise.

**Control units**
Hardware units that control I/O operations for one or more devices. You can view information about control units through I/O operations, and can start or stop data going to them by blocking and unblocking ports.

**Controller**
A unit that controls I/O operations for one or more devices.

**Converted mode (CVC)**
A channel operating in converted (CVC) mode transfers data in blocks and a CBY channel path transfers data in bytes. Converted CVC or CBY channel paths can communicate with a parallel control unit. This resembles a point-to-point parallel path and dedicated connection, regardless whether the path passes through a switch.

**Couple data set**
A data set that is created through the XCF couple data set format utility and, depending on its designated type, is shared by some or all of the z/OS systems in a sysplex. See also **sysplex**, **couple data set**, and **XCF couple data set**.

**Coupling facility**
The hardware element that provides high-speed caching, list processing, and locking functions in a sysplex.

**CP**
See **central processor**.

**CPC**
See **central processor complex**.

**CPC operations management commands**
A set of commands and responses for controlling the operation of System/390 CPCs.

**CPC subset**
All or part of a CPC. It contains the minimum resource to support a single control program.

**CPCB**
See **command processor control block**.

**CPU**
Central processing unit. Deprecated term for **processor**.

**Cross-system coupling facility (XCF)**
A component of z/OS that provides functions to support cooperation between authorized programs running within a sysplex.

**CTC**
The channel-to-channel (CTC) channel can communicate with a CTC on another host for intersystem communication.

**Customer Information Control System (CICS)**
A general-purpose transactional program that controls online communication between terminal users and a database for a large number of end users on a real-time basis.

**Customization dialogs**
The customization dialogs are an ISPF application. They are used to customize the enterprise policy, like, for example, the enterprise resources and the relationships between resources, or the automation policy for systems in the enterprise. How to use these dialogs is described in IBM Tivoli System Automation for z/OS Customizing and Programming.

**CVC**
See **converted mode**.

**D**

**DataPower X150z**
See IBM Websphere DataPower Integration Appliance X150 for zEnterprise (DataPower X150z).

**DASD**
See **direct access storage device**.

**Data services task (DST)**
The NetView subtask that gathers, records, and manages data in a VSAM file or a network device that contains network management information.

**Data set**
The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

**Data set members**
Members of partitioned data sets that are individually named elements of a larger file that can be retrieved by name.

**DBCS**
See **double-byte character set**.
DCCF
See disabled console communication facility.

DCF
See Document Composition Facility.

DELAY Report
An RMF report that shows the activity of each job in the system and the hardware and software resources that are delaying each job.

device
A piece of equipment. Devices can be workstations, printers, disk drives, tape units, remote systems or communications controllers. You can see information about all devices attached to a particular switch, and control paths and jobs to devices.

DEVR Report
An RMF report that presents information about the activity of I/O devices that are delaying jobs.

dialog
Interactive 3270 panels.

direct access storage device (DASD)
A device that allows storage to be directly accessed, such as a disk drive.

disabled console communication facility (DCCF)
A z/OS component that provides limited-function console communication during system recovery situations.

disk operating system (DOS)
An operating system for computer systems that use disks and diskettes for auxiliary storage of programs and data. Software for a personal computer that controls the processing of programs. For the IBM Personal Computer, the full name is Personal Computer Disk Operating System (PCDOS).

display
To present information for viewing, usually on the screen of a workstation or on a hardcopy device.

Deprecated term for panel.

distribution manager
The component of the NetView program that enables the host system to use, send, and delete files and programs in a network of computers.

Document Composition Facility (DCF)
An IBM licensed program used to format input to a printer.

domain
An access method and its application programs, communication controllers, connecting lines, modems, and attached workstations.

In SNA, a system services control point (SSCP) and the physical units (PUs), logical units (LUs), links, link stations, and associated resources that the SSCP can control with activation requests and deactivation requests.

double-byte character set (DBCS)
A character set, such as Kanji, in which each character is represented by a 2-byte code.

DP enterprise
Data processing enterprise.

DSIPARM
This file is a collection of members of NetView's customization.

DST
Data Services Task.

E

EBCDIC
See Extended Binary Coded Decimal Interchange Code.

ECB
See event control block.

EMCS
Extended multiple console support. See also multiple console support.

ensemble
A collection of one or more zEnterprise nodes (including any attached zBX) that are managed as a single logical virtualized system by the Unified Resource Manager, through the Hardware Management Console.

ensemble member
A zEnterprise node that has been added to an ensemble.

enterprise
The compositive of all operational entities, functions, and resources that form the total business concern and that require an information system.

enterprise monitoring
Enterprise monitoring is used by SA z/OS to update the NetView Management Console (NMC) resource status information that is stored in the Resource Object Data Manager (RODM).
Resource status information is acquired by enterprise monitoring of the Resource Measurement Facility (RMF) Monitor III service information at user-defined intervals. SA z/OS stores this information in its operational information base, where it is used to update the information presented to the operator in graphic displays.

**Enterprise Systems Architecture (ESA)**
A hardware architecture that reduces the effort required for managing data sets and extends addressability for system, subsystem, and application functions.

**entries**
Resources, such as processors, entered on panels.

**entry type**
Resources, such as processors or applications, used for automation and monitoring.

**environment**
Data processing enterprise.

**error threshold**
An automation policy setting that specifies when SA z/OS should stop trying to restart or recover an application, subsystem or component, or offload a data set.

**ESA**
See **Enterprise Systems Architecture**

**event**
In NetView, a record indicating irregularities of operation in physical elements of a network.

An occurrence of significance to a task; for example, the completion of an asynchronous operation, such as an input/output operation.

Events are part of a trigger condition, such that if all events of a trigger condition have occurred, a startup or shutdown of an application is performed.

**event control block (ECB)**
A control block used to represent the status of an event.

**exception condition**
An occurrence on a system that is a deviation from normal operation. SA z/OS monitoring highlights exception conditions and allows an SA z/OS enterprise to be managed by exception.

**Extended Binary Coded Decimal Interchange Code (EBCDIC)**
A coded character set of 256 8-bit characters developed for the representation of textual data. See also **American Standard Code for Information Interchange**

**extended recovery facility (XRF)**
A facility that minimizes the effect of failures in z/OS, VTAM, the host processor, or high availability applications during sessions between high availability applications and designated terminals. This facility provides an alternate subsystem to take over sessions from the failing subsystem.

**F**

**fallback system**
See **secondary system**

**field**
A collection of bytes within a record that are logically related and are processed as a unit.

**file manager commands**
A set of SA z/OS commands that read data from or write data to the automation control file or the operational information base. These commands are useful in the development of automation that uses SA z/OS facilities.

**focal point**
In NetView, the focal-point domain is the central host domain. It is the central control point for any management services element containing control of the network management data.

**focal point system**
A system that can administer, manage, or control one or more target systems. There are a number of different focal point system associated with IBM automation products.

**NMC focal point system**
The NMC focal point system is a NetView system with an attached workstation server and LAN that gathers information about the state of the network. This focal point system uses RODM to store the data it collects in the data model. The information stored in RODM can be accessed from any LAN-connected workstation with NetView Management Console installed.
NPDA focal point system. This is a NetView system that collects all the NPDA alerts that are generated within your enterprise. It is supported by NetView. If you have SA z/OS installed the NPDA focal point system must be the same as your NMC focal point system. The NPDA focal point system is also known as the alert focal point system.

SA z/OS Processor Operations focal point system. This is a NetView system that has SA z/OS host code installed. The SA z/OS Processor Operations focal point system receives messages from the systems and operator consoles of the machines that it controls. It provides full systems and operations console function for its target systems. It can be used to IPL these systems. Note that some restrictions apply to the Hardware Management Console for an S/390 microprocessor cluster.

SA z/OS SDF focal point system. The SA z/OS SDF focal point system is an SA z/OS NetView system that collects status information from other SA z/OS NetViews within your enterprise.

Status focal point system. In NetView, the system to which STATMON, VTAM and NLDLM send status information on network resources. If you have a NMC focal point, it must be on the same system as the Status focal point.

Hardware Management Console. Although not listed as a focal point, the Hardware Management Console acts as a focal point for the console functions of an S/390 microprocessor cluster. Unlike all the other focal points in this definition, the Hardware Management Console runs on a LAN-connected workstation.

frame For a System/390 microprocessor cluster, a frame contains one or two central processor complexes (CPCs), support elements, and AC power distribution.

full-screen mode
In NetView, a form of panel presentation that makes it possible to display the contents of an entire workstation screen at once. Full-screen mode can be used for fill-in-the-blanks prompting. Contrast with line mode.

gateway session
An NetView-NetView Task session with another system in which the SA z/OS outbound gateway operator logs onto the other NetView session without human operator intervention. Each end of a gateway session has both an inbound and outbound gateway operator.

generic alert
Encoded alert information that uses code points (defined by IBM and possibly customized by users or application programs) stored at an alert receiver, such as NetView.

group A collection of target systems defined through configuration dialogs. An installation might set up a group to refer to a physical site or an organizational or application entity.

group entry A construct, created with the customization dialogs, used to represent and contain policy for a group.

group entry type A collection of target systems defined through the customization dialog. An installation might set up a group to refer to a physical site or an organizational entity. Groups can, for example, be of type STANDARD or SYSPLEX.

Hardware Management Console (HMC) A user interface through which data center personnel configure, control, monitor, and manage System z hardware and software resources. The HMC communicates with each central processor complex (CPC) through the Support Element. On an IBM zEnterprise 196 (z196), using the Unified Resource Manager on the HMCs or Support Elements, personnel can also create and manage an ensemble.

Hardware Management Console Application (HWMCA) A direct-manipulation object-oriented graphical user interface that provides a single point of control and single system image for hardware elements. The HWMCA provides grouping support, aggregated and real-time system status.
using colors, consolidated hardware messages support, consolidated operating system messages support, consolidated service support, and hardware commands targeted at a single system, multiple systems, or a group of systems.

**heartbeat**
In SA z/OS, a function that monitors the validity of the status forwarding path between remote systems and the NMC focal point, and monitors the availability of remote z/OS systems, to ensure that status information displayed on the SA z/OS workstation is current.

**help panel**
An online panel that tells you how to use a command or another aspect of a product.

**hierarchy**
In the NetView program, the resource types, display types, and data types that make up the organization, or levels, in a network.

**high-level language (HLL)**
A programming language that provides some level of abstraction from assembler language and independence from a particular type of machine. For the NetView program, the high-level languages are PL/I and C.

HLL  See [high-level language](#)

**host (primary processor)**
The processor that you enter a command at (also known as the *issuing processor*).

**host system**
In a coupled system or distributed system environment, the system on which the facilities for centralized automation run. SA z/OS publications refer to target systems or focal-point systems instead of hosts.

**HWMCA**
See [Hardware Management Console Application](#)

**Hypervisor**
A program that allows multiple instances of operating systems or virtual servers to run simultaneously on the same hardware device. A hypervisor can run directly on the hardware, can run within an operating system, or can be imbedded in platform firmware. Examples of hypervisors include PR/SM, z/VM, and PowerVM Enterprise Edition.

**IBM blade**
A customer-acquired, customer-installed select blade to be managed by IBM zEnterprise Unified Resource Manager. One example of an IBM blade is a POWER7 blade.

**IBM Smart Analyzer for DB2 for z/OS**
An optimizer that processes certain types of data warehouse queries for DB2 for z/OS.

**IBM System z Application Assist Processor (zAAP)**
A specialized processor that provides a Java execution environment, which enables Java-based web applications to be integrated with core z/OS business applications and backend database systems.

**IBM System z Integrated Information Processor (zIIP)**
A specialized processor that provides computing capacity for selected data and transaction processing workloads and for selected network encryption workloads.

**IBM Websphere DataPower Integration Appliance X150 for zEnterprise (DataPower X150z)**
A purpose-built appliance that simplifies, helps secure, and optimizes XML and Web services processing.

**IBM zEnterprise 196 (z196)**
The newest generation of System z family of servers built on a new processor chip, with enhanced memory function and capacity, security, and on demand enhancements to support existing mainframe workloads and large scale consolidation.

**IBM zEnterprise BladeCenter Extension (zBX)**
A heterogeneous hardware infrastructure that consists of a BladeCenter chassis attached to an IBM zEnterprise 196 (z196). A BladeCenter chassis can contain IBM blades or optimizers.

**IBM zEnterprise BladeCenter Extension (zBX) blade**
Generic name for all blade types supported in an IBM zEnterprise BladeCenter Extension (zBX). This term includes IBM blades and optimizers.
IBM zEnterprise System (zEnterprise)
A heterogeneous hardware infrastructure that can consist of an IBM zEnterprise 196 (z196) and an attached IBM zEnterprise BladeCenter Extension (zBX) Model 002, managed as a single logical virtualized system by the Unified Resource Manager.

IBM zEnterprise Unified Resource Manager
Licensed Internal Code (LIC), also known as firmware, that is part of the Hardware Management Console. The Unified Resource Manager provides energy monitoring and management, goal-oriented policy management, increased security, virtual networking, and data management for the physical and logical resources of a given ensemble.

I/O operations
The part of SA z/OS that provides you with a single point of logical control for managing connectivity in your active I/O configurations. I/O operations takes an active role in detecting unusual conditions and lets you view and change paths between a processor and an I/O device, using dynamic switching (the ESCON director). Also known as I/O Ops.

I/O Ops
See I/O operations

I/O resource number
Combination of channel path identifier (CHPID), device number, etc. See Internal token

images
A grouping of processors and I/O devices that you define. You can define a single-image mode that allows a multiprocessor system to function as one central processor image.

IMS
See Information Management System

IMS/VS
See Information Management System/Virtual Storage

inbound
In SA z/OS, messages sent to the focal-point system from the PC or target system.

inbound gateway operator
The automation operator that receives incoming messages, commands, and responses from the outbound gateway operator at the sending system. The inbound gateway operator handles communications with other systems using a gateway session.

Information Management System (IMS)
Any of several system environments available with a database manager and transaction processing that are capable of managing complex databases and terminal networks.

Information Management System/Virtual Storage (IMS/VS)
A database/data communication (DB/DC) system that can manage complex databases and networks.
Synonymous with Information Management System

INGEIO PROC
The I/O operations default procedure name. It is part of the SYS1.PROCLIB.

initial microprogram load
The action of loading microprograms into computer storage.

initial program load (IPL)
The initialization procedure that causes an operating system to commence operation.
The process by which a configuration image is loaded into storage at the beginning of a workday or after a system malfunction.
The process of loading system programs and preparing a system to run jobs.

initialize automation
SA z/OS-provided automation that issues the correct z/OS start command for each subsystem when SA z/OS is initialized. The automation ensures that subsystems are started in the order specified in the automation control files and that prerequisite applications are functional.

input/output configuration data set (IOCDS)
A configuration definition built by the I/O configuration program (IOCP) and stored on disk files associated with the processor controller.

input/output support processor (IOSP)
The hardware unit that provides I/O support functions for the primary support processor and maintenance support functions for the processor controller.
Interactive System Productivity Facility (ISPF)
An IBM licensed program that serves as a full-screen editor and dialog manager. Used for writing application programs, it provides a means of generating standard screen panels and interactive dialogs between the application programmer and the terminal user. See also Time Sharing Option.

interested operator list
The list of operators who are to receive messages from a specific target system.

internal token
A logical token (LTK); name by which the I/O resource or object is known; stored in IODF.

IOCDS
See input/output configuration data set.

IOSP
See input/output support processor.

IPL
See initial program load.

ISPF
See Interactive System Productivity Facility.

ISPF console
You log on to ISPF from this 3270-type console to use the runtime panels for I/O operations and SA z/OS customization panels.

issuing host
The base program that you enter a command for processing with. See primary host.

J
JCL
See job control language.

JES
See job entry subsystem.

JES2
An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system. In an installation with more than one processor, each JES2 processor independently controls its job input, scheduling, and output processing. See also job entry subsystem and JES3.

JES3
An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system. In complexes that have several loosely coupled processing units, the JES3 program manages processors so that the global processor exercises centralized control over the local processors and distributes jobs to them using a common job queue. See also job entry subsystem and JES2.

job
A set of data that completely defines a unit of work for a computer. A job usually includes all necessary computer programs, linkages, files, and instructions to the operating system.

An address space.

job control language (JCL)
A problem-oriented language designed to express statements in a job that are used to identify the job or describe its requirements to an operating system.

job entry subsystem (JES)
An IBM licensed program that receives jobs into the system and processes all output data that is produced by jobs. In SA z/OS publications, JES refers to JES2 or JES3, unless otherwise stated. See also JES2 and JES3.

K
Kanji
An ideographic character set used in Japanese. See also double-byte character set.

L
LAN
See local area network.

line mode
A form of screen presentation in which the information is presented one line at a time in the message area of the terminal screen. Contrast with full-screen mode.

link
In SNA, the combination of the link connection and the link stations joining network nodes; for example, a System/370 channel and its associated protocols, a serial-by-bit connection under the control of synchronous data link control (SDLC). See synchronous data link control.

In SA z/OS, link connection is the physical medium of transmission.

link-attached
Describes devices that are physically connected by a telecommunication line. Contrast with channel-attached.
Linux on System z
UNIX-like open source operating system conceived by Linus Torvalds and developed across the internet.

local Pertaining to a device accessed directly without use of a telecommunication line. Synonymous with channel-attached

local area network (LAN) A network in which a set of devices is connected for communication. They can be connected to a larger network. See also token ring
A network that connects several devices in a limited area (such as a single building or campus) and that can be connected to a larger network.

logical partition (LP) A subset of the processor hardware that is defined to support an operating system. See also logically partitioned mode

logical switch number (LSN) Assigned with the switch parameter of the CHPID macro of the IOCP.

logical token (LTOK) Resource number of an object in the IODF.

logical unit (LU) In SNA, a port through which an end user accesses the SNA network and the functions provided by system services control points (SSCPs). An LU can support at least two sessions, one with an SSCP and one with another LU, and may be capable of supporting many sessions with other LUs. See also physical unit and system services control point

logical unit 6.2 (LU 6.2) A type of logical unit that supports general communications between programs in a distributed processing environment. LU 6.2 is characterized by:
• A peer relationship between session partners
• Efficient use of a session for multiple transactions
• A comprehensive end-to-end error processing
• A generic application program interface (API) consisting of structured verbs that are mapped to a product implementation

logically partitioned (LPAR) mode A central processor mode that enables an operator to allocate system processor hardware resources among several logical partitions. Contrast with basic mode

LOGR The sysplex logger.

LP See logical partition

LPAR See logically partitioned mode

LSN See logical switch number

LU See logical unit

LU 6.2 See logical unit 6.2

LU 6.2 session A session initiated by VTAM on behalf of an LU 6.2 application program, or a session initiated by a remote LU in which the application program specifies that VTAM is to control the session by using the APPCCMD macro. See logical unit 6.2

LU-LU session In SNA, a session between two logical units (LUs) in an SNA network. It provides communication between two end users, or between an end user and an LU services component.

M

MAT Deprecated term for NetView automation table

MCA See Micro Channel architecture

MCS See multiple console support

member A specific function (one or more modules or routines) of a multisystem application that is defined to XCF and assigned to a group by the multisystem application. A member resides on one system in the sysplex and can use XCF services to communicate (send and receive data) with other members of the same group.

message automation table (MAT) Deprecated term for NetView automation table

message class A number that SA z/OS associates with a message to control routing of the message. During automated operations, the classes associated with each message
issued by SA z/OS are compared to the classes assigned to each notification operator. Any operator with a class matching one of the message’s classes receives the message.

message forwarding
The SA z/OS process of sending messages generated at an SA z/OS target system to the SA z/OS focal-point system.

message group
Several messages that are displayed together as a unit.

message monitor task
A task that starts and is associated with a number of communications tasks. Message monitor tasks receive inbound messages from a communications task, determine the originating target system, and route the messages to the appropriate target control tasks.

message processing facility (MPF)
A z/OS table that screens all messages sent to the z/OS console. The MPF compares these messages with a customer-defined list of messages (based on this message list, messages are automated and/or suppressed from z/OS console display), and marks messages to automate or suppress. Messages are then broadcast on the subsystem interface (SSI).

message suppression
The ability to restrict the amount of message traffic displayed on the z/OS console.

Micro Channel architecture
The rules that define how subsystems and adapters use the Micro Channel bus in a computer. The architecture defines the services that each subsystem can or must provide.

microprocessor
A processor implemented on one or a small number of chips.

migration
Installation of a new version or release of a program to replace an earlier version or release.

MP
Multiprocessor.

MPF
See message processing facility.

MPFLSTxx
The MPFLST member that is built by SA z/OS.

multi-MVS environment
physical processing system that is capable of operating more than one MVS image. See also MVS image.

multiple console support (MCS)
A feature of MVS that permits selective message routing to multiple consoles.

Multiple Virtual Storage (MVS)
An IBM operating system that accesses multiple address spaces in virtual storage. The predecessor of z/OS.

multiprocessor (MP)
A CPC that can be physically partitioned to form two operating processor complexes.

multisystem application
An application program that has various functions distributed across z/OS images in a multisystem environment.

multisystem environment
An environment in which two or more systems reside on one or more processors. Or one or more processors can communicate with programs on the other systems.

MVS
See Multiple Virtual Storage

MVS image
A single occurrence of the MVS operating system that has the ability to process work. See also multi-MVS environment and single-MVS environment.

MVS/ESA
Multiple Virtual Storage/Enterprise Systems Architecture. See z/OS.

MVS/JES2
Multiple Virtual Storage/Job Entry System 2. A z/OS subsystem that receives jobs into the system, converts them to an internal format, selects them for execution, processes their output, and purges them from the system. In an installation with more than one processor, each JES2 processor independently controls its job input, scheduling, and output processing.

N
See network addressable unit.

NAU
See network addressable unit.
See network accessible unit.

NCCF  See Network Communications Control Facility.

NCP  See network control program (general term).

NCP/token ring interconnection
A function used by ACF/NCP to support token ring-attached SNA devices. NTRI also provides translation from token ring-attached SNA devices (PUs) to switched (dial-up) devices.

NetView
An IBM licensed program used to monitor a network, manage it, and diagnose network problems. NetView consists of a command facility that includes a presentation service, command processors, automation based on command lists, and a transaction processing structure on which the session monitor, hardware monitor, and terminal access facility (TAF) network management applications are built.

NetView (NCCF) console
A 3270-type console for NetView commands and runtime panels for system operations and processor operations.

NetView automation procedures
A sequence of commands, packaged as a NetView command list or a command processor written in a high-level language. An automation procedure performs automation functions and runs under the NetView program.

NetView automation table (AT)
A table against which the NetView program compares incoming messages. A match with an entry triggers the specified response. SA z/OS entries in the NetView automation table trigger an SA z/OS response to target system conditions. Formerly known as the message automation table (MAT).

NetView command list language
An interpretive language unique to NetView that is used to write command lists.

NetView Graphic Monitor Facility (NGMF)
Deprecated term for NetView Management Console.

NetView hardware monitor
The component of NetView that helps identify network problems, such as hardware, software, and microcode, from a central control point using interactive display techniques. Formerly called network problem determination application.

NetView log
The log that NetView records events relating to NetView and SA z/OS activities in.

NetView Management Console (NMC)
A function of the NetView program that provides a graphic, topological presentation of a network that is controlled by the NetView program. It provides the operator different views of a network, multiple levels of graphical detail, and dynamic resource status of the network. This function consists of a series of graphic windows that allows you to manage the network interactively. Formerly known as the NetView Graphic Monitor Facility (NGMF).

NetView message table
See NetView automation table.

NetView paths via logical unit (LU 6.2)
A type of network-accessible port (VTAM connection) that enables end users to gain access to SNA network resources and communicate with each other. LU 6.2 permits communication between processor operations and the workstation. See logical unit 6.2.

NetView-NetView task (NNT)
The task that a cross-domain NetView operator session runs under. Each NetView program must have a NetView-NetView task to establish one NNT session. See also operator station task.

NetView-NetView task session
A session between two NetView programs that runs under a NetView-NetView task. In SA z/OS, NetView-NetView task
sessions are used for communication between focal point and remote systems.

**network**
An interconnected group of nodes.

In data processing, a user application network. See SNA network.

**network accessible unit (NAU)**
In SNA networking, any device on the network that has a network address, including a logical unit (LU), physical unit (PU), control point (CP), or system services control point (SSCP). It is the origin or the destination of information transmitted by the path control network. Synonymous with network addressable unit.

**network addressable unit (NAU)**
Synonym for network accessible unit.

**Network Communications Control Facility (NCCF)**
The operations control facility for the network. NCCF consists of a presentation service, command processors, automation based on command lists, and a transaction processing structure on which the network management applications NLDM and NPDA are built. NCCF is a precursor to the NetView command facility.

**Network Control Program (NCP)**
An IBM licensed program that provides communication controller support for single-domain, multiple-domain, and interconnected network capability. Its full name is Advanced Communications Function for the Network Control Program.

**network control program (NCP)**
A program that controls the operation of a communication controller.

A program used for requests and responses exchanged between physical units in a network for data flow control.

**Network Problem Determination Application (NPDA)**
An NCCF application that helps you identify network problems, such as hardware, software, and microcode, from a central control point using interactive display methods. The alert manager for the network. The precursor of the NetView hardware monitor.

**Networking NetView**
In SA z/OS the NetView that performs network management functions, such as managing the configuration of a network. In SA z/OS it is common to also route alerts to the Networking NetView.

**NGMF**
Deprecated term for NetView Management Console.

**NGMF focal-point system**
Deprecated term for NMC focal point system.

**NIP**
See nucleus initialization program.

**NMC focal point system**
See focal point system.

**NMC workstation**
The NMC workstation is the primary way to dynamically monitor SA z/OS systems. From the windows, you see messages, monitor status, view trends, and react to changes before they cause problems for end users. You can use multiple windows to monitor multiple views of the system.

**NNT**
See NetView-NetView task.

**notification message**
An SA z/OS message sent to a human notification operator to provide information about significant automation actions. Notification messages are defined using the customization dialogs.

**notification operator**
A NetView console operator who is authorized to receive SA z/OS notification messages. Authorization is made through the customization dialogs.

**NPDA**
See Network Problem Determination Application.

**NPDA focal-point system**
See focal point system.

**NTRI**
See NCP/token ring interconnection.

**nucleus initialization program (NIP)**
The program that initializes the resident control program; it allows the operator to...
request last-minute changes to certain options specified during system generation.

Object value
An average Workflow or Using value that SA z/OS can calculate for applications from past service data. SA z/OS uses the objective value to calculate warning and alert thresholds when none are explicitly defined.

OCA In SA z/OS, operator console A, the active operator console for a target system. Contrast with OCB.

OCB In SA z/OS, operator console B, the backup operator console for a target system. Contrast with OCA.

OPC/A See Operations Planning and Control/Advanced.


Open Systems Adapter (OSA)
I/O operations can display the Open System Adapter (OSA) channel logical definition, physical attachment, and status. You can configure an OSA channel on or off.

Operating system (OS)
Software that controls the execution of programs and that may provide services such as resource allocation, scheduling, input/output control, and data management. Although operating systems are predominantly software, partial hardware implementations are possible.

Operations
The real-time control of a hardware device or software function.

Operations Planning and Control/Advanced (OPC/A)
A set of IBM licensed programs that automate, plan, and control batch workload. OPC/A analyzes system and workload status and submits jobs accordingly.

Operations Planning and Control/Enterprise Systems Architecture (OPC/ESA)
A set of IBM licensed programs that automate, plan, and control batch workload. OPC/ESA analyzes system and workload status and submits jobs accordingly. The successor to OPC/A.

Operator
A person who keeps a system running.
A person or program responsible for managing activities controlled by a given piece of software such as z/OS, the NetView program, or IMS.
A person who operates a device.
In a language statement, the lexical entity that indicates the action to be performed on operands.

Operator console
A functional unit containing devices that are used for communications between a computer operator and a computer. (T)
A display console used for communication between the operator and the system, used primarily to specify information concerning application programs and I/O operations and to monitor system operation.
In SA z/OS, a console that displays output from and sends input to the operating system (z/OS, LINUX, VM, VSE). Also called operating system console.
In the SA z/OS operator commands and configuration dialogs, OC is used to designate a target system operator console.

Operator station task (OST)
The NetView task that establishes and maintains the online session with the network operator. There is one operator station task for each network operator who logs on to the NetView program.

Operator view
A set of group, system, and resource definitions that are associated together for monitoring purposes. An operator view appears as a graphic display in the graphical interface showing the status of the defined groups, systems, and resources.

OperatorView entry
A construct, created with the
customization dialogs, used to represent and contain policy for an operator view.

optimizer
A special-purpose hardware component or appliance that can perform a limited set of specific functions with optimized performance when compared to a general-purpose processor. Because of its limited set of functions, an optimizer is an integrated part of a processing environment, rather than a stand-alone unit. One example of an optimizer is the IBM Smart Analytics Optimizer for DB2 for z/OS.

OS  See operating system
OSA  See Open Systems Adapter
OST  See operator station task

outbound
In SA z/OS, messages or commands from the focal-point system to the target system.

outbound gateway operator
The automation operator that establishes connections to other systems. The outbound gateway operator handles communications with other systems through a gateway session. The automation operator sends messages, commands, and responses to the inbound gateway operator at the receiving system.

P

page
The portion of a panel that is shown on a display surface at one time.
To transfer instructions, data, or both between real storage and external page or auxiliary storage.

panel
A formatted display of information that appears on a terminal screen. Panels are full-screen 3270-type displays with a monospaced font, limited color and graphics.
By using SA z/OS panels you can see status, type commands on a command line using a keyboard, configure your system, and passthru to other consoles. See also help panel.
In computer graphics, a display image that defines the locations and characteristics of display fields on a display surface. Contrast with screen.

parallel channels
Parallel channels operate in either byte (BY) or block (BL) mode. You can change connectivity to a parallel channel operating in block mode.

parameter
A variable that is given a constant value for a specified application and may represent an application, for example. An item in a menu for which the user specifies a value or for which the system provides a value when the menu is interpreted.
Data passed to a program or procedure by a user or another program, specifically as an operand in a language statement, as an item in a menu, or as a shared data structure.

partition
A fixed-size division of storage.
In VSE, a division of the virtual address area that is available for program processing.
On an IBM Personal Computer fixed disk, one of four possible storage areas of variable size; one can be accessed by DOS, and each of the others may be assigned to another operating system.

partitionable CPC
A CPC that can be divided into 2 independent CPCs. See also physical partition, single-image mode, MIP, and side.

partitioned data set (PDS)
A data set in direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

passive monitoring
In SA z/OS, the receiving of unsolicited messages from z/OS systems and their resources. These messages can prompt updates to resource status displays. See also active monitoring.

PCE
A processor controller. Also known as the support processor or service processor in some processor families.

PDB  See policy database
PDS  See partitioned data set
**Physical partition**
Part of a CPC that operates as a CPC in its own right, with its own copy of the operating system.

**Physical unit (PU)**
In SNA, the component that manages and monitors the resources (such as attached links and adjacent link stations) of a node, as requested by a system services control point (SSCP) through an SSCP-PU session. An SSCP activates a session with the physical unit to indirectly manage, through the PU, resources of the node such as attached links.

**Physically partitioned (PP) configuration**
A mode of operation that allows a multiprocessor (MP) system to function as two or more independent CPCs having separate power, utilities, and maintenance boundaries. Contrast with single-image mode.

**PLEXID Group**
PLEXID group or "extended XCF communication group" is a term used in conjunction with a sysplex. The PLEXID group includes System Automation Agents for a subset of a sysplex or for the entire sysplex. It is used to provide XCF communication beyond the SAplex boundaries. For a detailed description, refer to "Defining the Extended XCF Communication Group" in IBM Tivoli System Automation for z/OS Planning and Installation.

**POI**
See program operator interface.

**Policy**
The automation and monitoring specifications for an SA z/OS enterprise. See IBM Tivoli System Automation for z/OS Defining Automation Policy.

**Policy database**
The automation definitions (automation policy) that the automation programmer specifies using the customization dialog is stored in the policy database. Also known as the PDB. See also automation policy.

**POR**
See power-on reset.

**Port**
System hardware that the I/O devices are attached to.
In an ESCON switch, a port is an addressable connection. The switch routes data through the ports to the channel or control unit. Each port has a name that can be entered into a switch matrix, and you can use commands to change the switch configuration.

An access point (for example, a logical unit) for data entry or exit.

A functional unit of a node that data can enter or leave a data network through.

In data communication, that part of a data processor that is dedicated to a single data channel for the purpose of receiving data from or transmitting data to one or more external, remote devices.

**Power-on reset (POR)**
A function that re-initializes all the hardware in a CPC and loads the internal code that enables the CPC to load and run an operating system. See initial microprogram load.

**PP**
See physical partition.

**PPI**
See program to program interface.

**PPT**
See primary POI task.

**PR/SM**
See Processor Resource/Systems Manager.

**Primary host**
The base program that you enter a command for processing at.

**Primary POI task (PPT)**
The NetView subtask that processes all unsolicited messages received from the VTAM program operator interface (POI) and delivers them to the controlling operator or to the command processor. The PPT also processes the initial command specified to execute when NetView is initialized and timer request commands scheduled to execute under the PPT.

**Primary system**
A system is a primary system for an application if the application is normally meant to be running there. SA z/OS starts the application on all the primary systems defined for it.

**Problem determination**
The process of determining the source of a problem; for example, a program component, machine failure, telecommunication facilities, user or contractor-installed programs or
equipment, environment failure such as a power loss, or user error.

**processor**
A device for processing data from programmed instructions. It may be part of another unit.
In a computer, the part that interprets and executes instructions. Two typical components of a processor are a control unit and an arithmetic logic unit.

**processor controller**
Hardware that provides support and diagnostic functions for the central processors.

**processor operations**
The part of SA z/OS that monitors and controls processor (hardware) operations. Processor operations provides a connection from a focal-point system to a target system. Through NetView on the focal-point system, processor operations automates operator and system consoles for monitoring and recovering target systems. Also known as ProcOps.

**Processor Resource/Systems Manager™ (PR/SM)**
The feature that allows the processor to use several operating system images simultaneously and provides logical partitioning capability. See also logically partitioned mode.

**ProcOps**
See processor operations.

**ProcOps Service Machine (PSM)**
The PSM is a CMS user on a VM host system. It runs a CMS multitasking application that serves as "virtual hardware" for ProcOps. ProcOps communicates via the PSM with the VM guest systems that are defined as target systems within ProcOps.

**product automation**
Automation integrated into the base of SA z/OS for the products CICS, DB2, IMS, TWS (formerly called features).

**program operator interface (POI)**
A NetView facility for receiving VTAM messages.

**program to program interface (PPI)**
A NetView function that allows user programs to send or receive data buffers from other user programs and to send alerts to the NetView hardware monitor from system and application programs.

**protocol**
In SNA, the meanings of, and the sequencing rules for, requests and responses used for managing the network, transferring data, and synchronizing the states of network components.

**proxy resource**
A resource defined like an entry type APL representing a processor operations target system.

**PSM**
See ProcOps Service Machine.

**PU**
See physical unit.

**R**

**RACF**
See Resource Access Control Facility.

**remote system**
A system that receives resource status information from an SA z/OS focal-point system. An SA z/OS remote system is defined as part of the same SA z/OS enterprise as the SA z/OS focal-point system to which it is related.

**requester**
A workstation from that user can log on to a domain from, that is, to the servers belonging to the domain, and use network resources. Users can access the shared resources and use the processing capability of the servers, thus reducing hardware investment.

**resource**
Any facility of the computing system or operating system required by a job or task, and including main storage, input/output devices, the processing unit, data sets, and control or processing programs.
In NetView, any hardware or software that provides function to the network.
In SA z/OS, any z/OS application, z/OS component, job, device, or target system capable of being monitored or automated through SA z/OS.

**Resource Access Control Facility (RACF)**
A program that can provide data security for all your resources. RACF protects data from accidental or deliberate unauthorized disclosure, modification, or destruction.
**resource group**
A physically partitionable portion of a processor. Also known as a side.

**Resource Measurement Facility (RMF)**
A feature of z/OS that measures selected areas of system activity and presents the data collected in the format of printed reports, System Management Facility (SMF) records, or display reports.

**Resource Object Data Manager (RODM)**
In NetView for z/OS, a component that provides an in-memory cache for maintaining real-time data in an address space that is accessible by multiple applications. RODM also allows an application to query an object and receive a rapid response and act on it.

**resource token**
A unique internal identifier of an ESCON resource or resource number of the object in the IODF.

**restart automation**
Automation provided by SA z/OS that monitors subsystems to ensure that they are running. If a subsystem fails, SA z/OS attempts to restart it according to the policy in the automation configuration file.

**Restructured Extended Executor (REXX)**
A general-purpose, high-level, programming language, particularly suitable for EXEC procedures or programs for personal computing, used to write command lists.

**return code**
A code returned from a program used to influence the issuing of subsequent instructions.

**REXX** See Restructured Extended Executor

**REXX procedure**
A command list written with the Restructured Extended Executor (REXX), which is an interpretive language.

**RMF** See Resource Measurement Facility

**RODM** See Resource Object Data Manager

**S**

**SAF** See Security Authorization Facility

**SA IOM**
See System Automation for Integrated Operations Management

**SAplex**
SAplex or “SA z/OS Subplex” is a term used in conjunction with a sysplex. In fact, a SAplex is a subset of a sysplex. However, it can also be a sysplex. For a detailed description, refer to “Using SA z/OS Subplexes” in IBM Tivoli System Automation for z/OS Planning and Installation.

**SA z/OS**
See System Automation for z/OS

**SA z/OS customization dialogs**
An ISPF application through which the SA z/OS policy administrator defines policy for individual z/OS systems and builds automation control data and RODM load function files.

**SA z/OS customization focal point system**
See focal point system

**SA z/OS data model**
The set of objects, classes and entity relationships necessary to support the function of SA z/OS and the NetView automation platform.

**SA z/OS enterprise**
The group of systems and resources defined in the customization dialogs under one enterprise name. An SA z/OS enterprise consists of connected z/OS systems running SA z/OS.

**SA z/OS focal point system**
See focal point system

**SA z/OS policy**
The description of the systems and resources that make up an SA z/OS enterprise, together with their monitoring and automation definitions.

**SA z/OS policy administrator**
The member of the operations staff who is responsible for defining SA z/OS policy.

**SA z/OS satellite**
If you are running two NetViews on a z/OS system to split the automation and networking functions of NetView, it is common to route alerts to the Networking NetView. For SA z/OS to process alerts properly on the Networking NetView, you
must install a subset of SA z/OS code, called an SA z/OS satellite on the Networking NetView.

**SA z/OS SDF focal point system**
See focal point system.

**SCA**
In SA z/OS, system console A, the active system console for a target hardware. Contrast with SCB.

**SCB**
In SA z/OS, system console B, the backup system console for a target hardware. Contrast with SCA.

**screen**
Deprecated term for panel.

**screen handler**
In SA z/OS, software that interprets all data to and from a full-screen image of a target system. The interpretation depends on the format of the data on the full-screen image. Every processor and operating system has its own format for the full-screen image. A screen handler controls one PS/2 connection to a target system.

**SDF**
See status display facility.

**SDLC**
See synchronous data link control.

**SDSF**
See System Display and Search Facility.

**secondary system**
A system is a secondary system for an application if it is defined to automation on that system, but the application is not normally meant to be running there. Secondary systems are systems to which an application can be moved in the event that one or more of its primary systems are unavailable. SA z/OS does not start the application on its secondary systems.

**Security Authorization Facility (SAF)**
An MVS interface with which programs can communicate with an external security manager, such as RACF.

**server**
A server is a workstation that shares resources, which include directories, printers, serial devices, and computing powers.

**service language command (SLC)**
The line-oriented command language of processor controllers or service processors.

**service period**
Service periods allow the users to schedule the availability of applications. A service period is a set of time intervals (service windows), during which an application should be active.

**service processor (SVP)**
The name given to a processor controller on smaller System/370 processors.

**service threshold**
An SA z/OS policy setting that determines when to notify the operator of deteriorating service for a resource. See also alert threshold and warning threshold.

**session**
In SNA, a logical connection between two network addressable units (NAUs) that can be activated, tailored to provide various protocols, and deactivated, as requested. Each session is uniquely identified in a transmission header by a pair of network addresses identifying the origin and destination NAUs of any transmissions exchanged during the session.

**session monitor**
The component of the NetView program that collects and correlates session-related data and provides online access to this information. The successor to NLDM.

**shutdown automation**
SA z/OS-provided automation that manages the shutdown process for subsystems by issuing shutdown commands and responding to prompts for additional information.

**side**
A part of a partitionable CPC that can run as a physical partition and is typically referred to as the A-side or the B-side.

**Simple Network Management Protocol (SNMP)**
A set of protocols for monitoring systems and devices in complex networks. Information about managed devices is defined and stored in a Management Information Base (MIB).

**single image**
A processor system capable of being physically partitioned that has not been physically partitioned. Single-image systems can be target hardware processors.
single-MVS environment
An environment that supports one MVS image. See also MVS image.

double-image (SI) mode
A mode of operation for a multiprocessor (MP) system that allows it to function as one CPC. By definition, a uniprocessor (UP) operates in single-image mode. Contrast with physically partitioned (PP) configuration.

SLC See service language command

SMP/E See System Modification Program/Extended

SNA See Systems Network Architecture

SNA network
In SNA, the part of a user-application network that conforms to the formats and protocols of systems network architecture. It enables reliable transfer of data among end users and provides protocols for controlling the resources of various network configurations. The SNA network consists of network addressable units (NAUs), boundary function components, and the path control network.

SNMP See Simple Network Management Protocol

solicited message
An SA z/OS message that directly responds to a command. Contrast with unsolicited message.

SSCP See system services control point

SSI See subsystem interface

start automation
SA z/OS-provided automation that manages and completes the startup process for subsystems. During this process, SA z/OS replies to prompts for additional information, ensures that the startup process completes within specified time limits, notifies the operator of problems, if necessary, and brings subsystems to an UP (or ready) state.

startup
The point in time that a subsystem or application is started.

status
The measure of the condition or availability of the resource.

status display facility (SDF)
The system operations part of SA z/OS that displays status of resources such as applications, gateways, and write-to-operator messages (WTORs) on dynamic color-coded panels. SDF shows spool usage problems and resource data from multiple systems.

status focal-point system
See focal point system.

steady state automation
The routine monitoring, both for presence and performance, of subsystems, applications, volumes and systems. Steady state automation may respond to messages, performance exceptions and discrepancies between its model of the system and reality.

structure
A construct used by z/OS to map and manage storage on a coupling facility.

subgroup
A named set of systems. A subgroup is part of an SA z/OS enterprise definition and is used for monitoring purposes.

SubGroup entry
A construct, created with the customization dialogs, used to represent and contain policy for a subgroup.

subplex
See SAplex

subsystem
A secondary or subordinate system, usually capable of operating independent of, or asynchronously with, a controlling system.

In SA z/OS, an z/OS application or subsystem defined to SA z/OS.

subsystem interface (SSI)
The z/OS interface over which all messages sent to the z/OS console are broadcast.

support element
A hardware unit that provides communications, monitoring, and diagnostic functions to a central processor complex (CPC).

support processor
Another name given to a processor
controller on smaller System/370 processors. See service processor.

SVP
See service processor.

switch identifier
The switch device number (swchdevn), the logical switch number (LSN) and the switch name.

switches
ESCON directors are electronic units with ports that dynamically switch to route data to I/O devices. The switches are controlled by I/O operations commands that you enter on a workstation.

symbolic destination name (SDN)
Used locally at the workstation to relate to the VTAM application name.

synchronous data link control (SDLC)
A discipline for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or nonswitched links. The configuration of the link connection may be point-to-point, multipoint, or loop. SDLC conforms to subsets of the Advanced Data Communication Control Procedures (ADCCP) of the American National Standards Institute and High-Level Data Link Control (HDLC) of the International Standards Organization.

SYSINFO Report
An RMF report that presents an overview of the system, its workload, and the total number of jobs using resources or delayed for resources.

SysOps
See system operations.

sysplex
A set of z/OS systems communicating and cooperating with each other through certain multisystem hardware components (coupling devices and timers) and software services (couple data sets).

In a sysplex, z/OS provides the coupling services that handle the messages, data, and status for the parts of a multisystem application that has its workload spread across two or more of the connected processors, sysplex timers, coupling facilities, and couple data sets (which contains policy and states for automation).

A Parallel Sysplex® is a sysplex that includes a coupling facility.

sysplex application group
A sysplex application group is a grouping of applications that can run on any system in a sysplex.

sysplex couple data set
A couple data set that contains sysplex-wide data about systems, groups, and members that use XCF services. All z/OS systems in a sysplex must have connectivity to the sysplex couple data set. See also couple data set.

Sysplex Timer
An IBM unit that synchronizes the time-of-day (TOD) clocks in multiple processors or processor sides. External Time Reference (ETR) is the z/OS generic name for the IBM Sysplex Timer (9037).

system
In SA z/OS, system means a focal point system (z/OS) or a target system (MVS, VM, VSE, LINUX, or CF).

System Automation for Integrated Operations Management
An outboard automation solution for secure remote access to mainframe/distributed systems. Tivoli System Automation for Integrated Operations Management, previously Tivoli AF/REMOTE, allows users to manage mainframe and distributed systems from any location.

The full name for SA IOM.

System Automation for OS/390
The full name for SA OS/390, the predecessor to System Automation for z/OS.

System Automation for z/OS
The full name for SA z/OS.

system console
A console, usually having a keyboard and a display screen, that is used by an operator to control and communicate with a system.

A logical device used for the operation and control of hardware functions (for...
example, IPL, alter/display, and reconfiguration). The system console can be assigned to any of the physical displays attached to a processor controller or support processor.

In SA z/OS, the hardware system console for processor controllers or service processors of processors connected using SA z/OS. In the SA z/OS operator commands and configuration dialogs, SC is used to designate the system console for a target hardware processor.

**System Display and Search Facility (SDSF)**
An IBM licensed program that provides information about jobs, queues, and printers running under JES2 on a series of panels. Under SA z/OS you can select SDSF from a pull-down menu to see the resources' status, view the z/OS system log, see WTOR messages, and see active jobs on the system.

**System entry**
A construct, created with the customization dialogs, used to represent and contain policy for a system.

**System Modification Program/Extended (SMP/E)**
An IBM licensed program that facilitates the process of installing and servicing an z/OS system.

**System Operations**
The part of SA z/OS that monitors and controls system operations applications and subsystems such as NetView, SDSF, JES, RMF, TSO, RODM, ACF/VTAM, CICS, IMS, and OPC. Also known as SysOps.

**System Services Control Point (SSCP)**
In SNA, the focal point within an SNA network for managing the configuration, coordinating network operator and problem determination requests, and providing directory support and other session services for end users of the network. Multiple SSCPs, cooperating as peers, can divide the network into domains of control, with each SSCP having a hierarchical control relationship to the physical units and logical units within its domain.

**System/390 Microprocessor Cluster**
A configuration that consists of central processor complexes (CPCs) and may have one or more integrated coupling facilities.

**Systems Network Architecture (SNA)**
The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks.

**T**
**TAF** See terminal access facility.

**Target** A processor or system monitored and controlled by a focal-point system.

**Target Control Task**
In SA z/OS, target control tasks process commands and send data to target systems and workstations through communications tasks. A target control task (a NetView autotask) is assigned to a target system when the target system is initialized.

**Target Hardware**
In SA z/OS, the physical hardware on which a target system runs. It can be a single-image or physically partitioned processor. Contrast with target system.

**Target System**
In a distributed system environment, a system that is monitored and controlled by the focal-point system. Multiple target systems can be controlled by a single focal-point system.

In SA z/OS, a computer system attached to the focal-point system for monitoring and control. The definition of a target system includes how remote sessions are established, what hardware is used, and what operating system is used.

**Task** A basic unit of work to be accomplished by a computer.

In the NetView environment, an operator station task (logged-on operator), automation operator (autotask), application task, or user task. A NetView task performs work in the NetView environment. All SA z/OS tasks are NetView tasks. See also message monitor task and target control task.
telecommunication line
Any physical medium, such as a wire or microwave beam, that is used to transmit data.

terminal access facility (TAF)
A NetView function that allows you to log onto multiple applications either on your system or other systems. You can define TAF sessions in the SA z/OS customization panels so you don't have to set them up each time you want to use them.

In NetView, a facility that allows a network operator to control a number of subsystems. In a full-screen or operator control session, operators can control any combination of subsystems simultaneously.

terminal emulation
The capability of a microcomputer or personal computer to operate as if it were a particular type of terminal linked to a processing unit to access data.

threshold
A value that determines the point at which SA z/OS automation performs a predefined action. See alert threshold, warning threshold and error threshold.

time of day (TOD)
Typically refers to the time-of-day clock.

Time Sharing Option (TSO)
An optional configuration of the operating system that provides conversational timesharing from remote stations. It is an interactive service on z/OS, MVS/ESA, and MVS/XT.

Time-Sharing Option/Extended (TSO/E)
An option of z/OS that provides conversational timesharing from remote terminals. TSO/E allows a wide variety of users to perform many different kinds of tasks. It can handle short-running applications that use fewer sources as well as long-running applications that require large amounts of resources.

timers
A NetView instruction that issues a command or command processor (list of commands) at a specified time or time interval.

Tivoli Workload Scheduler (TWS)
A family of IBM licensed products that plan, execute and track jobs on several platforms and environments. The successor to OPC/A.

TOD
Time of day.

token ring
A network with a ring topology that passes tokens from one attaching device to another; for example, the IBM Token-Ring Network product.

TP
See transaction program

transaction program
In the VTAM program, a program that performs services related to the processing of a transaction. One or more transaction programs may operate within a VTAM application program that is using the VTAM application program interface (API). In that situation, the transaction program would request services from the applications program using protocols defined by that application program. The application program, in turn, could request services from the VTAM program by issuing the APPCCMD macro instruction.

transitional automation
The actions involved in starting and stopping subsystems and applications that have been defined to SA z/OS. This can include issuing commands and responding to messages.

translating host
Role played by a host that turns a resource number into a token during a unification process.

trigger
Triggers, in combination with events and service periods, are used to control the starting and stopping of applications in a single system or a parallel sysplex.

TSO
See Time Sharing Option

TSO console
From this 3270-type console you are logged onto TSO or ISPF to use the runtime panels for I/O operations and SA z/OS customization panels.

TSO/E
See Time-Sharing Option/Extended

TWS
See Tivoli Workload Scheduler

U
UCB  See unit control block

unit control block (UCB)
A control block in common storage that describes the characteristics of a particular I/O device on the operating system and that is used for allocating devices and controlling I/O operations.

unsolicited message
An SA z/OS message that is not a direct response to a command.

uniform resource identifier (URI)
A uniform resource identifier is a string of characters used to identify a name of a web resource. Such identification enables interaction with representations of the web resource over the internet, using specific protocols.

user task
An application of the NetView program defined in a NetView TASK definition statement.

Using An RMF Monitor III definition. Jobs getting service from hardware resources (processors or devices) are using these resources. The use of a resource by an address space can vary from 0% to 100% where 0% indicates no use during a Range period, and 100% indicates that the address space was found using the resource in every sample during that period.

V

view In the NetView Graphic Monitor Facility, a graphical picture of a network or part of a network. A view consists of nodes connected by links and may also include text and background lines. A view can be displayed, edited, and monitored for status information about network resources.

Virtual Server A logical construct that appears to comprise processor, memory, and I/O resources conforming to a particular architecture. A virtual server can support an operating system, associated middleware, and applications. A hypervisor creates and manages virtual servers.

Virtual Server Collection
A set of virtual servers that supports a workload. This set is not necessarily static. The constituents of the collection at any given point are determined by virtual servers involved in supporting the workload at that time.

virtual Server Image
A package containing metadata that describes the system requirements, virtual storage drives, and any goals and constraints for the virtual machine (for example, isolation and availability). The Open Virtual Machine Format (OVF) is a Distributed Management Task Force (DMTF) standard that describes a packaging format for virtual server images.

Virtual Server Image Capture
The ability to store metadata and disk images of an existing virtual server. The metadata describes the virtual server storage, network needs, goals and constraints. The captured information is stored as a virtual server image that can be referenced and used to create and deploy other similar images.

Virtual Server Image Clone
The ability to create an identical copy (clone) of a virtual server image that can be used to create a new similar virtual server.

Virtual Storage Extended (VSE)
A system that consists of a basic operating system (VSE/Advanced Functions), and any IBM supplied and user-written programs required to meet the data processing needs of a user. VSE and the hardware that it controls form a complete computing system. Its current version is called VSE/ESA.

Virtual Telecommunications Access Method (VTAM)
An IBM licensed program that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability. Its full name is Advanced Communications Function for the Virtual Telecommunications Access Method. Synonymous with ACF/VTAM.

VM Second Level Systems Support
With this function, Processor Operations is able to control VM second level
systems (VM guest systems) in the same way that it controls systems running on real hardware.

**VM/ESA**
Virtual Machine/Enterprise Systems Architecture. Its current version is called z/VM.

**volume**
A direct access storage device (DASD) volume or a tape volume that serves a system in an SA z/OS enterprise.

**VSE**
See Virtual Storage Extended

**VTAM**
See Virtual Telecommunications Access Method

**W**

**warning threshold**
An application or volume service value that determines the level at which SA z/OS changes the associated icon in the graphical interface to the warning color. See alert threshold

**workstation**
In SA z/OS workstation means the graphic workstation that an operator uses for day-to-day operations.

**write-to-operator (WTO)**
A request to send a message to an operator at the z/OS operator console. This request is made by an application and is handled by the WTO processor, which is part of the z/OS supervisor program.

**write-to-operator-with-reply (WTOR)**
A request to send a message to an operator at the z/OS operator console that requires a response from the operator. This request is made by an application and is handled by the WTO processor, which is part of the z/OS supervisor program.

**WTO**
See write-to-operator

**WTOR**
See write-to-operator-with-reply

**WWV**
The US National Institute of Standards and Technology (NIST) radio station that provides standard time information. A second station, known as WWVB, provides standard time information at a different frequency.

**X**

**XCF**
See cross-system coupling facility

**XCF couple data set**
The name for the sysplex couple data set prior to MVS/ESA System Product Version 5 Release 1. See also sysplex couple data set

**XCF group**
A set of related members that a multisystem application defines to XCF. A member is a specific function, or instance, of the application. A member resides on one system and can communicate with other members of the same group across the sysplex.

**XRF**
See extended recovery facility

**Z**

**z/OS**
An IBM mainframe operating system that uses 64-bit real storage. See also Base Control Program

**z/OS component**
A part of z/OS that performs a specific z/OS function. In SA z/OS, component refers to entities that are managed by SA z/OS automation.

**z/OS subsystem**
Software products that augment the z/OS operating system. JES and TSO/E are examples of z/OS subsystems. SA z/OS includes automation for some z/OS subsystems.

**z/OS system**
A z/OS image together with its associated hardware, which collectively are often referred to simply as a system, or z/OS system.

**z196**
See IBM Enterprise 196 (z196)

**zAAP**
See IBM System z Application Assist Processor (zAAP)

**zBX**
See IBM zEnterprise BladeCenter Extension (zBX)

**zBX blade**
See IBM zEnterprise BladeCenter Extension (zBX) blade

**zCPC**
The physical collection of main storage, central processors, timers, and channels within a zEnterprise mainframe. Although this collection of hardware resources is part of the larger zEnterprise central
processor complex, you can apply energy management policies to zCPC that are different from those that you apply to any attached IBM zEnterprise BladeCenter Extension (zBX) or blades. See also central processor complex.

zIIP See IBM System z Integrated Information Processor (zIIP)

zEnterprise See IBM zEnterprise System (zEnterprise)
Index

A

accessibility xi
ACF command dialog 15
ACF file manager command 15
ACTIVATE command
description 431
of a VM guest system 433
status checks 461
active, ACTIVE=SAVED on ESCD 543
adding your user ID to the interested
operator list 421
additional parameters for system
operations commands 10
allocating alternate couple data set
(CDS) 189
ALLOW command 295
alternate couple data set (CDS)
allocating 189
spare volumes 189
turn into primary CDS 190
AOCHELP command dialog 22
AOCTRACE command dialog 23
API
Query command rules 560
application
related events 43, 44
application group, setting status or
attribute of 156
application move, resource aware 196
ASF file manager command 28
ASFUSER file manager command 30
assigning names, WRITE command 349
attribute hierarchy 546
attribute of a resource or application
group, setting 156
attributes 556
authorization levels 559
automatic connectivity checks 551
automation
from status message 377
keywords 377
turning flags on or off 88
automation control file WARM start 15
automation manager
diagnostic functions 85
initiating diagnostic functions 86
showing details about 82
work item statistics 86
automation manager commands
(continued)
ingtopo 258
ingtrig 259
ingvote 270
Automation manager commands
ingsched 228
automation operators, determining
active 34
backout option 545
backout processing 546
BLOCK command 297

B
canceling a console lock
with ISQXUNL 428
canceling a console lock with
ISQXUNL 426
cascaded switch
FICON 294
CBU command 434
CCTRL 437
CFRM policy
pending changes 110, 189
policy switch 103
chain planning 551
ports 551
CHAIN command 299
chained path
examples 551
vary path requests 556
changing values with ISQVARS 377
channel
CVC 555
DISPLAY CHP command 304
parallel 555
REMOVE CHP command 332
closing a target system with
ISQXCLS 392
command
authorization level 559
availability 559
command failures 546
command output, port I/O
definition 540
command routing mechanism 235
commands
ACF 15
ACTIVATE 431
ACTIVATE, of a VM guest
system 433
ALLOW 295
AOCHELP 22
AOCTRACE 23
ASF 28
commands (continued)
ASFUSER 30
BLOCK 297
CU 434
CCNTL 437
CHAIN 299
common 353, 431
CONNECT 438
CTRLCONS 442
DEACTIVATE 443
DEACTIVATE, of a VM guest
system 445
DISCONNECT 303
DISPACF 32
DISPAOPS 34
DISPAPG 36
DISPASF 38
DISPAUTO 40
DISPERRS 41
DISPEVT 43
DISPEVT 44
DISPFALGS 47
DISPGW 50
DISPINFO 52
DISPLAY CHANGECHECK 304
DISPLAY CHP 304
DISPLAY DEV 306
DISPLAY HOST 308
DISPLAY NAME 311
DISPLAY HOST 314
DISPLAY RESULTS 316
DISPLAY SWITCH 321
DISPLAY TIMEOUT 325
DISPLAY VARY 325
DISPMDCS 54
DISPMTR 55
DISPSCHD 58
DISPSFLT 60
DISPSSTAT 63
DISPSYS 67
DISABLE 69
DISABLE 70
DISPINT 72
ensemble 354
EXPLAIN 73
EXTINT 46
EXTINT, for a VM guest system 446
GETCLUSTER 447
GETIINFO 448
GETISTAT 449
GETLOCK 328
GETS 450
GETSINFO 452
GETSSTAT 453
ICNTL 454
IDQCMD 360
INGAMS 74
INGAUTO 88
INGCF 92

© Copyright IBM Corp. 1996, 2014
commands (continued)
INGCF DRAIN 99
INGCF ENABLE 103, 106
INGCF MAINT 106
INGCF PATH 107
INGCF STRUCTURE 108
INGCFG 110
INGCFL 110
INGCICS 112
INGDLA 119
INGEVENT 121
INGFILTER 122
INGGROUP 129
INGHIST 137
INGHWSRV 139
INGIMS 140
INGINFO 146
INGLIST 148
INGLKP 158
INGMDFY 163
INGMOVE 166
INGMSGS 171
INGNTFY 174
INGPAC 177
INGPLEX 184
INGPLEX CDS 189
INGPLEX CONSOLE 197
INGPLEX IPL 200
INGPLEX SDUMP 201
INGPLEX SLIP 207
INGPLEX SVDUMP 204
INGPLEX SYTEM 195
INGPLEX SYTEM, WLM-related data 196
INGPW 180
INGRELS 208
INQREQ 211
INGRPT 221
INGRUN 224
INGSCH 228
INGSEND 235
INGSESS 239
INGST 243
INGSTR 247
INGTHRES 254
INGTOPO 258
INGTRIG 259
INGTWS 262
INGVOTE 270
ISQACT 533
ISQCCMD 355
ISQCLEAR 533
ISQCMON 359
ISQEXEC 362
ISQHELP 364
ISQIPS 365
ISQMSG 533
ISQOVRD 367
ISQPSM 354
ISQUERY 534
ISQROUTE 368
ISQSEND 370
ISQSDH 372
ISQSTART 374
ISQSTATUS 354
ISQSTOP 376
ISQTRACE 353

commands (continued)
ISQVARS 377, 397
ISQXLS 392
ISQXCONS 393
ISQXDRIL 396
ISQXDST 397
ISQXIII 418
ISQXLOC 420
ISQXOPT 424
ISQXPSM 426
ISQXUNL 428
LOAD 461
LOAD, for a VM guest system 463
LOGREC 329
MONITOR 274
OOCOD 464
OPCAQRY 275
POWERMOD 466
PROFILE 468
PROHIBIT 330
QSM 354
Query rules 560
REMOVE CHP 332
REMOVE SWITCH 334
RESERVE 480
RESET CHANGECHECK 336
RESET HOST 337
RESET SWITCH 340
RESET TIMEOUT 341
RESTART 278
RESTART, of a VM guest system 483
RESTART CHP 341
RESTART SWITCH 343
RESYNC 279
SECLOG 483
SETSTATE 283
SECTION 350
SETTIMER 286
START 484
START, of a VM guest system 485
STOP 485
STOP, of a VM guest system 486
STOPALL 491
STOP, of a VM guest system 486
SYSRESET 491
TCDATA 492
TCM 494
communications path
monitoring status 415
CONNECT command 301
common commands
GETCLUSTER 447
GETINFO 448
GETSTAT 449
GETSDIR 450
GETSINF 452
GETSSTAT 453
LOAD, for a VM guest system 463
processor operations commands 353
RESTART 482
RESTART, of a VM guest system 483
START 484
START, of a VM guest system 485
STOP 485
STOP, of a VM guest system 486
SYSRESET 491
SYSRESET, of a VM guest system 492
TCDATA 492
TCM 494
backout option 545
backout option 545
force option 544
making changes to 543
nobodybackout option 545
nobodycheck option 544
nocheck option 544
force | nocheck | force 544
port attributes 556
UNBLOCK command 345
varying paths 556
consensus processing, I/O operations 293
console
connecting a lock with ISQXUNL 426,
428
closing with ISQXLS 392
locking access 90
controlling secondary automation manager functions 81
converter in CVC path 555
converter, CVC channel 555
couple data set (CDS), alternate
allocating 189
spare volumes 189
turn into primary CDS 190
converter facility (CF)
channels 543
draining 99
enabling 103, 106
populating 104
receiver path 94
sender path 100, 103
states 94
CPCDATA command 440
CRM 235
CTRLCONS command 442
CU chain 551
CV converter, support 555
CVC channel 555
D

DEACTIVATE command
defined 443
of a VM guest system 445
reinitialize 445
deleting your user ID from the interested
operator list 421
destination switch 551
detail about automation manager,
showing 82
device, DISPLAY DEV command 306
diagnostic functions
automation manager 85
initiating for automation manager 86
disability xi
disabling a switch 549
DISCONNECT command 303
DISPACEF command dialog 32
DISPAOPS command 34
DISPAWG command 36
DISPASF command 38
DISPAUTO command 40
DISPERRS command 41
DISPEVT command 43
DISPEVTS command 44
DISPFLGS command 47
DISPGW command 50
DISPINFO command 52
DISPLAY CHANGECHECK
command 304
DISPLAY CHP command 304
DISPLAY HOST command 308
DISPLAY NAME 311
DISPLAY PORT command 314
DISPLAY RESULTS command 316
displaying
NetView connection path status 415
outstanding MVS requests 396
status panels with ISQXDST 397
sysplex timer information 67
target system information 424
DISPMSGS command 54
DISPMTR command 55
DISPNTFY command 174
DISPSCHD command 58
DISPSFILT command 60
DISPSSTAT command 63
DISPSYS command
defined 67
DISPTRREE command 69
DISPTRG command 70
distributed feature code, starting with
ISQXIII 418
DR,L command (MVS) 396
duplexing
system-managed 109
user-managed 109
dynamic connection, and PROHIBIT
command 330

E

enabling a switch 550
ensemble commands 354
Ensembles 408
enter a console command for
automation 370
error recovery 546
establish processor operations
environment 374
event and related applications 43, 44
EXPLAIN command 73
EXTINT command
defined 446
for a VM guest system 446

failed commands 546
FICON cascaded switches 294
FICON switches 294
file manager commands
ACF 15
ASF 28
ASUSER 30
flags 88
focal-point ports, disable 445
force option 544
functions of secondary automation
manager, controlling 81

G

gateway sessions 50
GETCLUSTER command 447
GETINFO command 448
GETISTAT command 449
GETLOCK command 450
GETSINFO command 452
GETSTAT command 453
getting help, ISQHELP command 364

hardware status 541
help
EXPLAIN command dialog 73
for commands 22, 364
ISQHELP 364
hierarchy of port attributes 546
host response timeout 341
host, primary 546

I

I/O operations
consensus processing 293
safe switching 293
I/O operations commands
ALLOW 295
BLOCK 297
CHAIN 299
CONNECT 301
DISCONNECT 303
DISPLAY CHANGECHECK 304
I/O operations commands (continued)
DISPLAY CHP 304
DISPLAY DEV 306
DISPLAY HOST 308
DISPLAY NAME 311
DISPLAY PORT 314
DISPLAY RESULTS 316
DISPLAY SWITCH 321
DISPLAY TIMEOUT 325
DISPLAY VARY 325
GETLOCK 328
LOGREC 329
PROHIBIT 330
REMOVE CHP 332
REMOVE SWITCH 334
RESET CHANGECHECK 336
RESET HOST 337
RESET SWITCH 340
RESET TIMEOUT 341
RESTORE CHP 341
RESTORE SWITCH 343
SYNC SWITCH 344
UNBLOCK 345
UNCHAIN 347
UNLOCK 348
WRITE 349
I/O operations, inter-I/O operations
communication 336, 337
ICNTL command 454
identifier, switch 540
IEA101A message 377, 391
IEA212A message 377
IEA213A message 377
IEA347A message 377, 391
INGAMS command 74
INGAUTO command 88
INGCF command 92
authorizations 93
DRAIN 99
ENABLE 103, 106
MAINT 106
PATH 107
STRUCTURE 108
INGCFG commands 110
INGCFL command 110
INGCICS command 112
INGDB2 system utility 116
INGDLA command 119
INGEVENT command 121
INGFILTER command 122
INGGROUP command 129
INGHIST command 137
INGHWSRV command 139
INGIMS command 140
INGINFO command 146
INGLIST command 148
INGLKUP 158
INGMDFY command 163
INGMOVE command 166
INGMSGS command 171
INGNTFY command 174
INGPAC command 177
INGPLEX command 184
CDS 189
CONSOLE 197
IPL 200
SDUMP 201

Index 605
INGPLEX command (continued)
SLIP 207
SVCDUMP 204
SYSTEM 195
SYSTEM, WLM-related data 196
INGPW command 180
INGRELS command 208
INGREQ command 211
INGRPT command 221
INGRUN command 224
INGSCHED command 228
INGSEND command 235
INGSEND command dialog 235
INGSESS command 239
INGSET command 243
INGSTR command 247
INGTHRES command 254
INGTOPO command 258
INGTRIG command 259
INGTWS command 262
INGVOTE command 264
initial status determination 279
initializing a target system with ISQXI1 418
initializing automation manager diagnostic functions 86
interested operator list
adding or deleting your userid 421
ISQSTART command 359
viewing 397
intersystem communication
DISPLAY HOST command 308
DISPLAY TIME-OUT 325
RESET CHANGECHECK 336
RESET HOST 337
IODF 94
IPL
ACTIVATE common command 431
LOAD common command 461
ISQACT command 533
ISQCCMD command 355
ISQCHK command 358
ISQCLEAR command 533
ISQCMON command defined 359
issued by ISQSTART 374
restrictions 359
ISQCMOD command 360
ISQEXEC command defined 362
usage with ISQXLOC 420
ISQHELP command 364
ISQIPSWT command 365
ISQLOG command 366
ISQMON command defined 359
ISQQUERY command 368
ISQSEND command defined 370
limited by ISQXCLS 392
service language commands 370
ISQSDH command 372
ISQSTART command 374
ISQSTATUS command 534
ISQSTOP command defined 376
restrictions 376
ISQTRAC command 355
ISQVAR command called by ISQXOPT 424
change status
target hardware 397
target system 397
target system attention 397
defined 377
keywords and status messages 377
ISQXCLS command 392
ISQXCON command managing connections 393
ISQXDRL command 396
ISQXDST command defined 397
restrictions 397
target system status values 397
ISQXII command 418
ISQXPL command check status 461
ISQXLOC command defined 420
usage with ISQXUNL 428
ISQXMON command 421
ISQXOPT command 424
ISQXPSM command 426
ISQXUNL command defined 428
usage with ISQXLOC 420
J
JES2, draining resources 72
K
keyboard xi
keywords
ISQVARS 377
sorted by keyword name 377
keywords and status messages 377
L
LOAD command defined 461
for a VM guest system 463
locking console access 420
locks
canceling a console lock 428
locks, canceling a console lock 426
logical switch number 539, 540
logical tokens 541
logical-to-physical coordination 344
LOGREC command 329
LSN 540
M
mask, allow|prohibit 541
message holding parameters 281
messages
IEA101A 377, 391
messages (continued)
IEA212A 377
IEA213A 377
IEA347A 377, 391
ISQCMON command 359
status 377
waiting for IEA101A start message 397
waiting for IEA347A start message 397
waiting for VM start message 397
MONITOR command 274
monitoring communications paths 415
MVS commands
DR,L command 396
QUESCE 445
MVS, displaying outstanding requests 396
MVSESA.RELOAD.ACTION flag 15
MVSESA.RELOAD.CONFIRM flag 15
N
name
generic 550
port 550
TCP/IP host 557
VTAM 557
WRITE command 349
naming a port 550
NetView commands, View 424
establishing processor operations 374
RMTCMD 235
NetView connection path 415
displaying status 415
NVC 415
SNMP 415
nobackout option 545
nocheck option 544
notforce option 544
O
OOCCOD command 464
OPCQRY command 275
OUTDSN parameter 13
OUTMODE parameter 12
P
panels
AOCHLP 22
AOCTRACT 23
authorized message receivers 54
Automation Operators 34
autoterm status 73
Command Dialogs 11
Command Response 32, 38
DISPAOPS 34
DISPAPO 36
DISPASF 38
DISPAUTO 40
DISPERRS 41
DISPEVT 43
safe switching, I/O operations 293
saved, ACTIVE=SAVED on ESCD 543
SECLOG command
  turning security logging ON/OFF 483
secondary automation manager, 
  controlling functions of 81
security logging
  turning security logging ON/OFF 483
sender path 100, 103
service language commands, entering 
  with ISQSEND 370
SETHOLD command 281
STATE command 283
SETTIMER command 286
setting
  attribute of a resource or application group 156
  status of a resource or application group 156
shortcut keys  xi
showing details about the automation manager 82
shutdown procedure 376
simulate pressing a console key 370
SLIP TRAP 207
spare volumes, alternate couple data set (CDS) 189
specifying resources 4
START command
  defined 484
  of a VM guest system 485
  starting a target processor 484
static connections
  CONNECT command 301
  DISCONNECT command 303
  establishing 301
status
  alert 397
  changing with SETSTATE 283
  clear to send 415
  closed
    target system 397
  DCCF 397
  degraded 397
  disabled wait 397
  displaying status panels 397
  environment alert 397
  gateways 50
  initialized 397
  IPL complete 397
  IPL failed 397
  last significant message 415
  load failed 397
  NetView connection path 415
  not active 397
  not operating 397
  of a resource or application group, 
    setting 156
  path problem 415
  processor operations
    attention status 397
    path detail 415
    path status 415
  target system attention status 397
  Processor operations
    target system status 397
    serious alert 397
    service 397
    service required 397
    stage-1 activation complete 397
    target hardware problem 397
    target systems 397
    undecided 397
    unknown 397
    values for target systems 397
    waiting for IEA101A start message 397
    waiting for IEA104A start message 397
    waiting for IEA104A start message 397
    waiting for VM start message 397
  status messages 377
STOP command
  defined 485
  of a VM guest system 486
STOPALL command 535
stopping
  processor operations 376
  target processors 485
STP command 486
STPDATA command 489
structure
  conditions 96, 110
  deallocation 100, 108
  duplexing
    system-managed 109
    user-managed 109
  pending policy changes 110, 189
structure (continued)
  rebuild
    system-managed 100, 109
    user-managed 100, 109
switch
  chained 551
  disabling 549
  FICON 294
  FICON cascaded 294
  logical switch number (LSN) 539, 
    540
  port 540
  removing 549
  restoring (enabling) 550
  switch identifier 540
switch chain 299
switch data, DISPLAY SWITCH 321
switch device number 540
switch log, LOGREC command 329
switching, safe, I/O operations 293
SYNC SWITCH command 344
syntax diagrams, format 5
sysexplexwide, refreshing configuration information 67
sysplex timer, displaying information 67

system operations commands (continued)

INGINFO 146
INGLIST 148
INGLKUP 158
INGMDFY 163
INGMOVE 166
INGMSGS 171
INGNTFY 174
INGPAC 177
INGPW 180
INGRELS 208
INGREQ 211
INGRPT 221
INGRUN 224
INGSEND 235
INGSESS 239
INGSET 243
INGSTR 247
INGTHRES 254
INGTOPO 257
INGVOTE 270
OPCAQRY 275
RESTART 278
TARGET 283
SETSTATE 283

System operations commands
INGSCHED 228

System Operations commands
SETHOLD 281

system operations commands
DISPGW 50
RESYNC 279
SETTIMER 286

system parameters 67

system utility
INGDB2 116

system-managed duplexing 109

system-managed rebuild 100, 109

tokens
logical 541
physical 542
trace, with AOCTRACE command 23

U
UNBLOCK command 345
UNCHAIN command 347
UNLOCK command 348

V
vary path requests
chained path 556
displaying failed varies 325
port attributes 556
viewing failed requests 325
varying paths 556
View command (NetView) 424
VM guest system
ACTIVATE command 433
DEACTIVATE command 445
EXTINT command 446
LOAD command 463
RESTART command 483
START command 485
STOP command 486
SYSRESET command 492
VTAM application name 557

W
wildcards 4
WLM-related data
INGPLEX command 196
work item statistics 86
WRITE command 349

Z
zEnterprise Ensembles
ISQECMD command 360

T

target console lock, removing with ISQOVRD 367
target control task
ISQEXEC 362
target control task, ISQEXEC 362
target hardware
status 397
target hardware summary 406
TARGET parameter 10
target processors
resetting 491
starting 484
stopping 485
target system
status 397
target system processor, restarting 482
target system, displaying and changing information 424
TCDATA command 492
TCM command 494
TCP/IP host name 557
thresholds, setting 254
timeout value, changing 341

Index 609