



**User's Guide**





**User's Guide**

**Note!**

Before using this information and the product it supports, be sure to read the general information under "Notices" on page xi.

This edition applies to IBM Tivoli System Automation for z/OS (Program Number 5698-SA3) Version 3 Release 2, an IBM licensed program, and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

IBM welcomes your comments. A form for readers' comments appears at the back of this publication. If the form has been removed, address your comments to:

IBM Deutschland Research & Development GmbH  
Department 3248  
Schoenaicher Strasse 220  
71032 Boeblingen  
Germany

If you prefer to send comments electronically, use one of the following methods:

FAX (Germany): 07031 16-3456  
FAX (Other Countries): +49 7031 16-3456  
Internet: s390id@de.ibm.com

**Internet**

Visit our homepage at <http://www.ibm.com/servers/eserver/zseries/software/sa>

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

© Copyright International Business Machines Corporation 1996, 2008. All rights reserved.

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

---

# Contents

<b>Figures</b> . . . . .	<b>vii</b>
<b>Tables</b> . . . . .	<b>ix</b>
<b>Notices</b> . . . . .	<b>xi</b>
Web Site Disclaimer . . . . .	xi
Programming Interface Information . . . . .	xi
Trademarks . . . . .	xii
<b>About This Book</b> . . . . .	<b>xiii</b>
Who Should Use This Book. . . . .	xiii
Where to Find More Information . . . . .	xiii
The System Automation for z/OS Library . . . . .	xiii
Related Product Information . . . . .	xiv
Using LookAt to look up message explanations . . . . .	xiv
Related Online Information . . . . .	xv
Summary of Changes for SC33-8263-05 . . . . .	xv
New Information . . . . .	xv

---

## Part 1. Introduction and Concepts . . 1

<b>Chapter 1. Introducing SA z/OS</b> . . . . .	<b>3</b>
Overview . . . . .	3
Component Description. . . . .	4
System Operations . . . . .	4
Processor Operations . . . . .	4
I/O Operations . . . . .	4
Hardware Overview . . . . .	4
SA z/OS Resources . . . . .	6
Channels . . . . .	8
Processors, Systems. . . . .	10
I/O Devices . . . . .	11
Automation Policy . . . . .	11
<b>Chapter 2. Concepts</b> . . . . .	<b>13</b>
Automation Concepts . . . . .	13
How Requests Become Goals . . . . .	19
How Request Propagation Works . . . . .	19
Conflicting Relationships and Requests . . . . .	20
How the Automation Manager and Automation Agents Cooperate in a Sysplex . . . . .	21
Locked Votes and Requests . . . . .	22
Controlling Application Move and Server Management . . . . .	23
Move Functions . . . . .	23
Controlling Application Move with Preference Values . . . . .	24
Resource Aware Application Move . . . . .	31
SA z/OS Automation Flags . . . . .	33
Automation Agent Flags . . . . .	33
The Automation Manager Global Automation Flag . . . . .	34
The Hold Flag . . . . .	35
Statuses Supplied by the Automation Manager . . . . .	35

Observed Status . . . . .	36
Automation Status . . . . .	37
Desired Status . . . . .	38
Startability Status . . . . .	38
Compound Status . . . . .	39
Health Status . . . . .	39
Statuses Supplied by Automation Agents . . . . .	40
Automation Agent Statuses . . . . .	40
Application Monitor Status . . . . .	46
Automatic Restart Manager Statuses . . . . .	47
Mapping the Automation Agent Statuses to the Automation Manager Observed Status . . . . .	48
SA z/OS Processes Involved in Status Determination . . . . .	50
Using SA z/OS for Monitoring. . . . .	52
Resynchronization . . . . .	52

## Chapter 3. Starting and Stopping SA z/OS . . . . . 53

Starting, Stopping, and Maintaining the Automation Manager . . . . .	53
Starting the Automation Manager: START (S) . . . . .	53
Stopping the Automation Manager: STOP (P) . . . . .	54
Making an Automation Manager the Primary . . . . .	54
Stopping the Automation Manager with Different Options. . . . .	54
Obtaining Automation Manager Diagnostic Information . . . . .	55
Takeover File Repair . . . . .	55
Starting NetView (System Operations) . . . . .	57
Starting and Stopping I/O Operations . . . . .	57
Starting and Stopping Processor Operations . . . . .	58
Starting and Stopping Applications . . . . .	58

## Chapter 4. Refreshing Automation Policy . . . . . 61

Refreshing an Automation Manager Configuration . . . . .	62
Identifying Current Configuration Files . . . . .	62

---

## Part 2. Using SA z/OS on the Host 65

<b>Chapter 5. Getting Started</b> . . . . .	<b>67</b>
Issuing Commands . . . . .	67
Using Command Dialogs. . . . .	68

## Chapter 6. How to Get Resource Information Sysplexwide . . . . . 69

Commands That Are Processed by the Automation Manager . . . . .	69
Using INGLIST to View Resources. . . . .	69
Using INGINFO to View a Selected Resource . . . . .	71
Displaying Application Group Information . . . . .	72
Displaying Monitor Information . . . . .	72

Displaying Relationship Information with INGRELS . . . . .	73
Commands That Are Processed by Automation Agents . . . . .	73
Displaying the Automation Agent View with DISPSTAT . . . . .	73
Using DISPINFO to Display Detailed Information	76
Using DISPTREE to Display Dependency Information . . . . .	76
Displaying Statistical Information . . . . .	77
Displaying Application Group Information . . . . .	77
Displaying Monitor Information . . . . .	77

**Chapter 7. How to Monitor and Control Resources . . . . . 79**

Starting and Stopping Resources . . . . .	80
Using INGREQ . . . . .	80
Using INGVOTE . . . . .	82
Using INGHIST . . . . .	83
Using Schedules . . . . .	84
Schedule Override . . . . .	84
Resource Override . . . . .	84
Displaying and Setting Trigger Information . . . . .	85
Displaying and Setting Events . . . . .	85
Determining Automation Agent Flag Values . . . . .	86
Displaying Actual and Effective Automation Agent Flags . . . . .	87
Displaying Automation Agent Flag Override Settings . . . . .	88
Setting Automation On or Off: INGAUTO . . . . .	88
Working with Application Groups . . . . .	88
Using INGGROUP to Change Policy for Applications . . . . .	89
Moving Sysplex Application Groups . . . . .	92
Setting Timers . . . . .	94
How to Create a Timer . . . . .	94
How to Modify a Timer . . . . .	94
How to Delete a Timer . . . . .	94
Interacting with the Automation Manager . . . . .	94
Changing the Automation Agent Status . . . . .	95
Monitoring Resources . . . . .	97
Monitoring Messages . . . . .	97
Monitoring Hardware . . . . .	100
Monitoring Health Performance . . . . .	102
Monitoring OMEGAMON Sessions . . . . .	103
Monitoring Processor Status . . . . .	107
Monitoring with the Status Display Facility . . . . .	114
SDF Focal Point Monitoring . . . . .	115
Interpreting Colors in SDF and DISPSTAT . . . . .	116
Using SDF . . . . .	118
How SDF Works . . . . .	118
How to Monitor WTORs with SDF . . . . .	121
How to Monitor Application Groups with SDF	121
Monitoring and Controlling a Sysplex . . . . .	122
How to Work with INGPLEX . . . . .	122
How to Work with INGSTR . . . . .	138
How SA z/OS Uses Error Thresholds . . . . .	139
Displaying Threshold Settings and Occurrences	139
How to Set or Change Error Thresholds . . . . .	140
Handling Jobs That Are Not Controlled by SA z/OS . . . . .	140

Communicating with Other Systems . . . . .	140
Sending Commands within a Sysplex . . . . .	140
Sending Commands to a Linux Target System	142
Terminal Access Facility Full-Screen Sessions	143
Gateway Sessions . . . . .	144

**Chapter 8. Scenario: Finding Out Why a Resource Is Not Available . . . . . 147**

**Chapter 9. Using I/O Operations . . . . . 157**

Making Connectivity Changes . . . . .	158
Display Commands To Obtain Data About I/O Resources . . . . .	158
Connectivity Commands That Affect a Switch . . . . .	158
Display and Utility Commands Related to Connectivity Commands . . . . .	159
Utility Commands for General Processing . . . . .	159
Using the I/O Operations ISPF Dialog . . . . .	160
Using the z/OS Operator Console for I/O Operations Commands . . . . .	160

**Chapter 10. SA z/OS Discovery Library Adapter . . . . . 163**

Overview . . . . .	163
Using the SA z/OS Discovery Library Adapter . . . . .	164

**Chapter 11. VM Second Level Systems Support . . . . . 165**

ProcOps Dialog . . . . .	165
Status Summary Panel (ISQESUM) . . . . .	166
Target Hardware Summary Panel (ISQEHARP)	166
Target System Summary Panel (ISQETARG) . . . . .	168
Guests Summary Panel (ISQETRGG) . . . . .	168
PATH Detail Panel (ISQEPNVC) . . . . .	169
PSM Console Mode . . . . .	169
Starting the PSM Control Program . . . . .	170
Stopping the PSM Control Program . . . . .	171
Restarting the PSM Control Program . . . . .	172

**Chapter 12. LPAR Management Functions . . . . . 173**

Functional Overview . . . . .	173
LPAR Management . . . . .	173
Activation Profile Management . . . . .	173
CPC and LPAR Information Queries . . . . .	173
Enabling the Functions . . . . .	174
Processor Operations . . . . .	174
BCP Internal Interface . . . . .	174
Usage Considerations . . . . .	175

**Part 3. Using SA z/OS on Workstations . . . . . 177**

**Chapter 13. Using the NetView Management Console for SA z/OS . . . . . 179**

Introduction . . . . .	179
Multiple Focal Point Support . . . . .	180
Starting the NMC . . . . .	180

Monitoring SA z/OS on the NMC . . . . .	181
Regular Resources . . . . .	181
Minor Resources, Anchors and Tokens . . . . .	182
Properties of SA z/OS Objects on the NMC Workstation . . . . .	185
Dynamic Behavior of the SA z/OS Topology Manager . . . . .	187
SA z/OS Topology Manager Initialization:	
INGTOPO INIT . . . . .	187
System Detection . . . . .	188
The Heartbeat Function . . . . .	188
Takeover when a System Goes Down . . . . .	188
Issuing Commands from the NMC . . . . .	188
Direct Commands . . . . .	188
Dialog Commands . . . . .	189
Interactive Commands . . . . .	189
The INGRICMD Command . . . . .	191
Creating Views . . . . .	192
Network Views and Exception Views . . . . .	192
Names . . . . .	193
SA z/OS Topology Manager Commands . . . . .	195
Processor Operations on the NMC . . . . .	195
Monitoring Sysplex Resources . . . . .	196
Problem Management . . . . .	196
Exception Views . . . . .	196
Marked and Unmarked Resources . . . . .	197
Updating Operator Data. . . . .	197
Problem Management of Minor Resources. . . . .	197

---

## Part 4. TEC Notification by SA z/OS . . . . . 199

### Chapter 14. Using TEC Notification by SA z/OS . . . . . 201

Understanding the Use of TEC Notification by SA z/OS. . . . .	201
Introduction of TEC Notification by SA z/OS . . . . .	202
SA z/OS Messages, Alerts and Tivoli Events . . . . .	202
Mapping SA z/OS Messages to Tivoli Events . . . . .	203
Mapping NetView Alerts to Tivoli Events . . . . .	209
TEC Rules for SA z/OS Events . . . . .	209
Operation of TEC Notification by SA z/OS . . . . .	211
Startup of TEC Notification by SA z/OS . . . . .	211
How TEC Notification by SA z/OS Works . . . . .	212
Termination of TEC Notification by SA z/OS . . . . .	214

---

## Part 5. Appendixes . . . . . 217

## Appendix A. Automation Initialization Options . . . . . 219

### Appendix B. Using the I/O Operations ISPF Dialog . . . . . 223

Getting Started . . . . .	223
Starting with the I/O Operations ISPF Main Menu . . . . .	223
The Modify Menu . . . . .	225
A Sample Way to Modify a Switch Matrix. . . . .	226
The View Menu . . . . .	228
The Options Menu . . . . .	230
The Help Menu . . . . .	232

### Appendix C. Problem Determination 233

Concurrent User Access within the Customization Dialog. . . . .	233
Maintaining System Status during a Status Forwarding Path Failure. . . . .	233
Debugging and Tracing . . . . .	234
Automation Manager State Trace Table . . . . .	234
Using Trace Services for the Automation Manager . . . . .	234
Using AOCTRACE . . . . .	236
How to Use a Log with INGHIST . . . . .	236
How to Use the Diagnostic Option of INGAMS . . . . .	237
Tracing and Debugging for ProcOps and the BCP Internal Interface . . . . .	242
Using Trace Services for I/O Operations . . . . .	243

### Appendix D. Hardware Interfaces: Additional Recovery Information . . . 247

BCP Internal Interface Recovery Information . . . . .	247
Determine BCP Internal Interface Communication Task Names . . . . .	247
BCP Internal Interface Recovery Actions . . . . .	247
How to Switch the IP Address for Target Hardware Used by Processor Operations. . . . .	249
How to Terminate BCP Internal Interface Connections or Clear Hardware Control Blocks after a Connection Failure . . . . .	249

## Glossary . . . . . 251

## Index . . . . . 271





## Figures

1. A Basic Hardware Configuration . . . . .	5	38. Checking Automation Flags with INGLIST	153
2. Conflicting Relationships and Requests	20	39. Checking Triggers with INGLIST . . . . .	153
3. Automation Manager and Automation Agents Cooperation in a Sysplex using XCF . . . . .	22	40. Complete Automation Environment for MSCSK1 . . . . .	154
4. Automation Manager and Automation Agents Cooperation in a Sysplex using MQSeries . . . . .	22	41. DISPTRG Command Dialog. . . . .	154
5. Non-Disruptive and Disruptive Restoration	27	42. Trigger Condition List . . . . .	155
6. Resource Aware Application Move Example Sysplex . . . . .	32	43. Verifying Automation Status . . . . .	155
7. Resource Aware Application Move. . . . .	32	44. SA z/OS Discovery Library Adapter	163
8. Automation Flag Span of Control . . . . .	33	45. ProcOps Panel Flow . . . . .	165
9. Observed Status Transitions . . . . .	37	46. Status Summary Panel ISQESUM . . . . .	166
10. Automations Status Transitions . . . . .	38	47. Target Hardware Summary Panel for PSMs ISQEHARP . . . . .	166
11. Application Startup Under SA z/OS's Control	59	48. Target Hardware Summary Panel for Non-PSMs ISQEHARP . . . . .	167
12. Application Stop Under SA z/OS's Control	60	49. Target System Summary Panel ISQETARG	168
13. DISPSYS Command Dialog . . . . .	63	50. PSMs/Guests Summary Panel ISQETRGG	168
14. INGAMS Command Dialog . . . . .	63	51. PATH Detail Panel ISQEPNVC. . . . .	169
15. Transitions for a Normal Subsystem . . . . .	96	52. SA z/OS-NMC Communication Overview	180
16. Transitions for an Abending Subsystem	96	53. Tivoli NetView for z/OS Management Console . . . . .	181
17. ISQXDST Command on NetView Console	101	54. Network View for Regular Resources	182
18. Target Hardware Summary Panel for OCF-Based Processors . . . . .	101	55. Subview with Applications of an Application Group . . . . .	182
19. ProcOps Target Status Summary Panel	107	56. Anchor View. . . . .	184
20. Target System Summary Panel . . . . .	108	57. View with 'Lost&Found'-Anchors. . . . .	185
21. Path Detail Panel . . . . .	113	58. Resource Properties Window . . . . .	187
22. Example SDF System Summary Panel	118	59. Parameter Window for INGTHRES . . . . .	189
23. Example SDF Hierarchy for KEY1. . . . .	119	60. NetView Console with INGREQ Panel	190
24. Example SDF Detail Status Display Panel for HUGC2APL0 . . . . .	120	61. Actualized View . . . . .	191
25. SDF Summary Status Panel . . . . .	121	62. Processor Operations View . . . . .	195
26. SDF Detail Status Display Panel for an Application Group . . . . .	122	63. Exception View . . . . .	197
27. Sending a Command to a Linux Target System. . . . .	143	64. Tivoli Event Inheritance . . . . .	206
28. TAF Fullscreen Display Panel . . . . .	144	65. Invoking the TEC Task Execution Dialog for Events . . . . .	213
29. INGLIST Output . . . . .	147	66. TEC Task Execution Dialog . . . . .	214
30. INGVOTE Output . . . . .	148	67. Sample AOF767I Message . . . . .	221
31. Refreshed INGLIST Output . . . . .	149	68. I/O operations ISPF Main Menu . . . . .	224
32. INGINFO Output . . . . .	149	69. INGHIST Logger Messages . . . . .	236
33. INGLIST Output for MSCSK* Resources	150	70. INGAMS Diagnostic Functions Panel	237
34. INGVOTE Output for MSCSK2 . . . . .	151	71. INGAMS Command Dialog, Show Details Panel . . . . .	238
35. INGSCHED for MSCSK2. . . . .	151	72. INGAMS Work Item Statistics Display	238
36. Schedule Override for MSCSK2 . . . . .	152	73. INGAMS Work Item History Display	239
37. Refreshed INGLIST Output . . . . .	152		



---

## Tables

1. System Automation for z/OS Library	xiii	7. Operator Tasks and Related Commands	79
2. Hierarchy of Requests . . . . .	15	8. Table of Default Subsystem Status Colors and Priorities . . . . .	116
3. Preference Values and Behavior of Members	25	9. Mapping SA z/OS Messages to SA z/OS Events . . . . .	203
4. Automation Agent Statuses . . . . .	40	10. SA z/OS Events and Corresponding Mapped SA z/OS Messages . . . . .	206
5. Regular Monitoring: Automation Status Determination. . . . .	46	11. Mapping SA z/OS Alerts to SA z/OS Events	209
6. Automation Agent to Automation Manager Status Mapping . . . . .	48		



---

## Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM product, program, or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

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

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  
71032 Boeblingen  
Germany

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

---

## Web Site Disclaimer

Any pointers in this publication to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement. IBM accepts no responsibility for the content or use of non-IBM Web sites specifically mentioned in this publication or accessed through an IBM Web site that is mentioned in this publication.

---

## Programming Interface Information

This publication documents information that is NOT intended to be used as a Programming Interface of System Automation for z/OS.

---

## Trademarks

The following terms are trademarks or service marks of the IBM Corporation in the United States or other countries or both:

CICS	DB2
ESCON	FICON
GDPS	IBM
IMS	MQSeries
MVS	NetView
OMEGAMON	OS/390
Parallel Sysplex	PR/SM
RACF	REXX
RMF	Sysplex Timer
System z	Tivoli
Tivoli Enterprise Console	VTAM
z/OS	z/VM
zSeries	

The following terms are trademarks of other companies:

- Linux is a registered trademark of Linus Torvalds.

---

## About This Book

This book describes IBM® Tivoli® System Automation for z/OS® (SA z/OS) and contains information on how to use SA z/OS to monitor and control your systems.

This document provides an overview of the operator tasks. It points you to ways to manage your systems. For further detail, refer to the help panels or online documentation available on both 3270 panels and workstation windows.

---

## Who Should Use This Book

This book is intended primarily for operators, but may also be useful for others. The information provided is specifically about using SA z/OS and assumes the following knowledge and experience:

- An understanding of z/OS
- An understanding of basic network concepts and terminology
- Using NetView®
- Using the NetView Management Console (NMC)
- Responding to messages and some troubleshooting

System administrators, programmers and help desk personnel may find the information helpful for installation, maintenance, and investigating and correcting problems. It also helps educate users about the SA z/OS functions.

---

## Where to Find More Information

### The System Automation for z/OS Library

The following table shows the information units in the System Automation for z/OS library:

*Table 1. System Automation for z/OS Library*

<b>Title</b>	<b>Order Number</b>
<i>IBM Tivoli System Automation for z/OS Planning and Installation</i>	SC33-8261
<i>IBM Tivoli System Automation for z/OS Customizing and Programming</i>	SC33-8260
<i>IBM Tivoli System Automation for z/OS Defining Automation Policy</i>	SC33-8262
<i>IBM Tivoli System Automation for z/OS User's Guide</i>	SC33-8263
<i>IBM Tivoli System Automation for z/OS Messages and Codes</i>	SC33-8264
<i>IBM Tivoli System Automation for z/OS Operator's Commands</i>	SC33-8265
<i>IBM Tivoli System Automation for z/OS Programmer's Reference</i>	SC33-8266
<i>IBM Tivoli System Automation for z/OS CICS Automation Programmer's Reference and Operator's Guide</i>	SC33-8267
<i>IBM Tivoli System Automation for z/OS IMS Automation Programmer's Reference and Operator's Guide</i>	SC33-8268
<i>IBM Tivoli System Automation for z/OS TWS Automation Programmer's Reference and Operator's Guide</i>	SC23-8269
<i>IBM Tivoli System Automation for z/OS End-to-End Automation Adapter</i>	SC33-8271

Table 1. System Automation for z/OS Library (continued)

Title	Order Number
IBM Tivoli System Automation for z/OS Monitoring Agent Configuration and User's Guide	SC33-8337

The System Automation for z/OS books are also available on CD-ROM as part of the following collection kit:

IBM Online Library z/OS Software Products Collection (SK3T-4270)

#### SA z/OS Home Page

For the latest news on SA z/OS, visit the SA z/OS home page at <http://www.ibm.com/servers/eserver/zseries/software/sa>

## Related Product Information

You can find books in related product libraries that may be useful for support of the SA z/OS base program by visiting the z/OS Internet Library at <http://www.ibm.com/servers/eserver/zseries/zos/bkserv/>

## Using LookAt to look up message explanations

LookAt is an online facility that lets you look up explanations for most of the IBM messages you encounter, as well as for some system abends and codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can use LookAt from these locations to find IBM message explanations for z/OS elements and features, z/VM<sup>®</sup>, VSE/ESA<sup>™</sup>, and Clusters for AIX<sup>®</sup> and Linux<sup>™</sup>:

- The Internet. You can access IBM message explanations directly from the LookAt Web site at <http://www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/>.
- Your z/OS TSO/E host system. You can install code on your z/OS or z/OS.e systems to access IBM message explanations using LookAt from a TSO/E command line (for example: TSO/E prompt, ISPF, or z/OS UNIX<sup>®</sup> System Services).
- Your Microsoft<sup>®</sup> Windows<sup>®</sup> workstation. You can install LookAt directly from the z/OS Collection (SK3T-4269) or the z/OS and Software Products DVD Collection (SK3T4271) and use it from the resulting Windows graphical user interface (GUI). The command prompt (also known as the DOS > command line) version can still be used from the directory in which you install the Windows version of LookAt.
- Your wireless handheld device. You can use the LookAt Mobile Edition from <http://www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/lookatm.html> with a handheld device that has wireless access and an Internet browser (for example: Internet Explorer for Pocket PCs, Blazer or Eudora for Palm OS, or Opera for Linux handheld devices).

You can obtain code to install LookAt on your host system or Microsoft Windows workstation from:

- A CD-ROM in the z/OS Collection (SK3T-4269).
- The z/OS and Software Products DVD Collection (SK3T4271).



- The LookAt Web site (click **Download** and then select the platform, release, collection, and location that suit your needs). More information is available in the LOOKAT.ME files available during the download process.

---

## Related Online Information

SA z/OS provides online help for all parts of its operator interfaces. From any terminal display, press PF1 to obtain help for that display.

SA z/OS also provides online help panels to support the tasks of installation and customization.

---

## Summary of Changes for SC33-8263-05

This document contains information previously presented in System Automation for z/OS V3R2.0 User's Guide, SC33-8263-04.

### New Information

| The SA z/OS Discovery Library Adapter (DLA) discovers automated resources  
| and their relationships within a sysplex (that is, the scope of the automation  
| manager) and generates output XML files. See Chapter 10, "SA z/OS Discovery  
| Library Adapter," on page 163 for more details about the SA z/OS DLA and how  
| to use it.

| "How to Terminate BCP Internal Interface Connections or Clear Hardware Control  
| Blocks after a Connection Failure" on page 249.

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. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.



---

## Part 1. Introduction and Concepts

This part provides an overview of System Automation for z/OS and its concepts. It contains the following:

- Chapter 1, "Introducing SA z/OS," on page 3
- Chapter 2, "Concepts," on page 13
- Chapter 3, "Starting and Stopping SA z/OS," on page 53
- Chapter 4, "Refreshing Automation Policy," on page 61



---

## Chapter 1. Introducing SA z/OS

System Automation for z/OS (SA z/OS) is a NetView-based application designed to provide a single point of control for a full range of systems management functions. SA z/OS plays a key role in supplying high-end automation solutions. SA z/OS functions include monitoring, controlling and automating a large range of system elements spanning both the hardware and software resources of your enterprise.

---

### Overview

SA z/OS is a systems management program with a single point of control. You see a single system image for a full range of essential systems management functions.

- Monitor** Monitor your resources to respond before they affect end users:
- Monitor hardware components
  - Monitor software products and applications
  - Monitor automated processes
  - Monitor messages and alerts
- Control** Take action to control conditions:
- Start and stop your entire enterprise system; initiate hardware and software startup and shutdown sequences
  - Manage both remote and local operations and support any zSeries® and 390-CMOS processor within a Parallel Sysplex
  - Manage several operating systems: z/OS, OS/390®, MVS™, VM, VSE, and Linux for zSeries
  - Control a coupling facility as a target system with coupling links in a Parallel Sysplex environment
  - React to errors and unscheduled events
- Automate** Automate many repetitive and complex tasks:
- Start and shut down software resources
  - Start and shut down hardware resources
  - Control channels and channel paths
  - Control availability of I/O devices
  - Control switching of I/O device ports
  - Detect and respond to system messages
  - Perform initial program load (IPL)
  - Perform system power-on reset (POR)
  - Build automation policy for your enterprise
  - Extend the built-in automation routines by writing your own automation policies

You monitor and control hardware and software resources from a NetView Management Console (NMC) workstation or from a NetView console.

## Component Description

### System Operations

System operations monitors and controls system operations applications and subsystems such as NetView, SDSF, JES, RMF™, TSO, RODM, ACF/VTAM, DB2®, CICS®, IMS™, OMEGAMON®, TBSM, and TWS.

With system operations, you can automate Parallel Sysplex applications. SA z/OS can automate applications distributed over a sysplex by virtually removing system boundaries for automation through its automation manager/automation agent design. SA z/OS reduces the complexity of managing a Parallel Sysplex through its goal driven automation and its concepts, such as grouping and powerful dependency support, which enable you to model your configuration. Single systems are also fully supported; the automation scope is then just one system.

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), and update the health status information that is displayed on the Tivoli Enterprise Portal (TEP) via the IBM Tivoli Monitoring infrastructure.

### Processor Operations

Processor operations monitors and controls processor hardware operations. It provides a connection from a focal point processor to a target processor. With NetView on the focal point system, processor operations automates operator and system consoles for monitoring and recovering target processors.

Processor operations allows you to power on and off multiple target processors and reset them, perform IPLs, set the time of day clocks, respond to messages, monitor status, and detect and resolve wait states.

### I/O Operations

I/O operations provides a single point of control for managing connectivity in your active I/O configurations. It takes an active role in detecting unusual I/O conditions and lets you view and change paths between a processor and an input/output device, which can involve using dynamic switching: the enterprise systems connection (ESCON®) or fiber channel connection (FICON®) switch.

I/O operations changes paths by letting you control channels, ports, switches, control units, and input/output devices. You can do this through an operator console or API.

---

## Hardware Overview

Figure 1 on page 5 illustrates a basic hardware configuration.

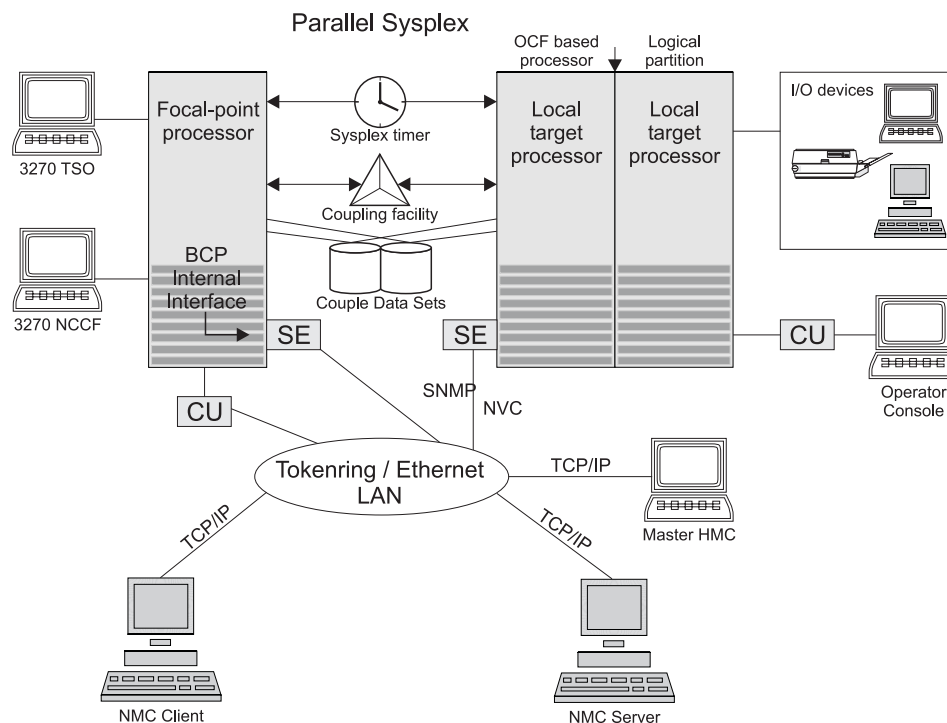


Figure 1. A Basic Hardware Configuration

It shows a two-processor Parallel Sysplex configuration with systems running on it. One is playing the role of a SA z/OS focal point. For example, the role of the SA z/OS NMC focal point with information about all the systems and applications in the sysplex, running under the control of SA z/OS.

Operators can use a workstation with the SA z/OS NMC client code installed, to work with graphical views of the SA z/OS controlled resources stored on the focal point. The NMC server component receives status changes from the NMC focal point and distributes them to the registered clients to update their dynamic resource views. Sysplex specific facilities, like the coupling facility hardware can be managed and controlled using the NMC's client graphical interface, as well as the 3270 NCCF based SA z/OS operator interfaces.

With the same interfaces, processor operations, another SA z/OS focal point function can be operated. With processor operations it is possible to manage and control the complete processor hardware in a sysplex. Operator tasks such as re-IPLing a sysplex member or activating a changed processor configuration can be accomplished. Processor operations uses the processor hardware infrastructure, consisting of the CPC Support Element (SE), or the Hardware Management Console (HMC) interconnected in a processor hardware LAN, to communicate with the own, other local, or remote located Support Elements of other CPCs. The Support Elements provide the Systems Management Interface OCF (Operations Command Facility) to perform hardware commands like LOAD or SYSTEM RESET to control the hardware and hardware images. SA z/OS processor operations can be customized to use SNA-based NetView connections (NVC), or IP based SNMP for communication. For Parallel Sysplex environments, SA z/OS provides an additional processor hardware interface, the BCP (basic control program) internal interface. This interface is independent from processor operations. It allows processor hardware operation in a sysplex, without requiring external network CUs (control units). From a system in the sysplex, the SE of the own CPC as well as the SEs of the other processors in the sysplex can be accessed.

### SA z/OS Resources

The following describes the functions and resources used by SA z/OS and its components.

<b>Focal point processor</b>	In a multisystem environment, the primary (or <i>issuing</i> ) host processor has SA z/OS installed. You might think of it as SA z/OS's <i>command center</i> . This is where you enter commands you want to process. It includes a central processor complex, which is a collection of hardware that consists of central storage, one or more central processors, timers, and channels. NetView and SA z/OS are installed on this processor.
<b>Target processor</b>	The processor that is controlled by a focal point processor. It can be a local or remote processor, or a logical partition of a processor. When you send a command from a focal point system, the target is the processor you want the command to affect. The operating system can be z/OS, OS/390, VM, VSE, or a coupling facility. If the operating system is z/OS, SA z/OS should be installed on the target processor for full SA z/OS functionality. If the operating system is VM, or VSE, or if the target processor is a coupling facility, only SA z/OS processor operations functions are available for the target.
<b>OCF-based processor</b>	CMOS-S/390, zSeries operations command facility (OCF) for interacting with human operators or external programs to perform operations management functions on the central processor. Processor operations uses this facility to manage and control the processors.
<b>Parallel Sysplex</b>	<p>A set of z/OS systems communicating and cooperating with each other through certain multisystem hardware components (coupling devices and sysplex timers) and software services (couple data sets). In a Parallel 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 containing policy and states for basic simplex functions are all parts of a sysplex.</p> <p>An z/OS sysplex can be viewed and managed as a single system image on the NMC workstation.</p>
<b>Logically Partitioned (LPAR) mode</b>	<p>A processor with the PR/SM™ feature that can be divided into partitions with separate logical system consoles that allocates hardware resources among several logical partitions. (It is called <i>logical</i> because the processor is not physically divided, but divided only by definition.) The partitions are defined, monitored, and activated separately by processor operations.</p> <p>A processor that does not use logical partitions is in “basic mode”.</p>



<b>Images</b>	<p>Processors can operate as <i>single images</i> or <i>partitioned images</i>. A single image allows a multiprocessor system to function as one central processor complex.</p> <p>A multiprocessor system that functions as two or more independent processors (with separate power, water, and maintenance boundaries) is a <i>physically partitioned configuration image</i>.</p>
<b>Communications links</b>	<p>Paths that connect the focal point system processor to target processors so that commands, messages, and alerts can flow. Two types of communication links can be selected for communication between the focal point and the target processor hardware: SNA based NetView connections (NVC) and IP based SNMP connections.</p>
<b>Communications Task</b>	<p>An SA z/OS user task responsible for all communications with the target processor hardware. Communications tasks receive commands from target control tasks or operator tasks and send them to the hardware. The number of communications tasks to be used is defined using the customization dialog.</p>
<b>OSA</b>	<p>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.</p>
<b>ESCON/FICON Director</b>	<p>The enterprise systems connection (ESCON) and fiber channel connection (FICON) Directors (also known as <i>switches</i> or <i>dynamic switches</i>) are hardware electronic switching units with ports that dynamically switch to route data to input/output devices. Each port on the switch is an addressable connection, controlled by I/O operations commands. Two switches (one dynamic and one static) can be linked or chained together to increase the path length.</p>
<b>Ports</b>	<p>On an ESCON/FICON switch, a port is an addressable connection. The switch routes data through the ports to the channel or control unit. Each port can be given a name that can be entered into a switch matrix, and you can use commands to change the switch configuration.</p>
<b>Control units</b>	<p>Control units are hardware units that control input/output 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. For example, if a control unit needs service, you can temporarily block all I/O paths going to it.</p>

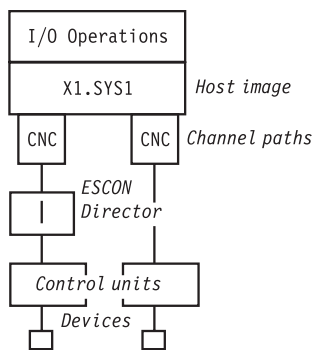
## Hardware Overview

<b>Coupling Facilities</b>	A hardware storage element with a high-speed cache, list processor, and locking functions that provides high performance random access to data for one system image or data that is shared among system images in a sysplex. With I/O operations you can see standalone coupling facilities. It handles them as control units with up to eight devices, all defined by the user. With SA z/OS system operations, you can display the status of coupling facilities from a single system's point of view or you can display sysplexwide status.
<b>Devices</b>	Input/output devices include hardware such as printers, tape drives, direct access storage devices (DASD), displays, or communications controllers. You can access them through multiple processors. You can see information about all devices and control paths to devices. You can vary devices, or groups of devices, online or offline.
<b>Sysplex Timer<sup>®</sup></b>	An IBM unit that synchronizes the time-of-day (TOD) clocks in a multiprocessor or in processor sides. External Time Reference (ETR) is the generic name for the IBM Sysplex Timer (9037).
<b>NMC Workstation</b>	A NetView feature running on a personal computer that provides a graphic series of windows controlled by NetView allowing you to monitor the SA z/OS enterprise graphically.
<b>Automation policy</b>	Operations and procedures (or <i>policy</i> ) information is kept in files that describe routine repetitive procedures, as well as the resources in your enterprise and their relationships. Automation automatically starts, restarts, monitors, or stops resources. You define the policy using the customization dialog. The resulting files are the system operations control files (automation control file and automation manager configuration file) (see also "Automation Policy" on page 11).
<b>Tivoli Enterprise Portal</b>	Tivoli Enterprise Portal (TEP) is the Java-based interface to the data monitoring and management resources of the IBM Tivoli Monitoring infrastructure. It displays information about the status of automation on z/OS systems and z/OS sysplexes that has been collected by the SA z/OS monitoring agent.

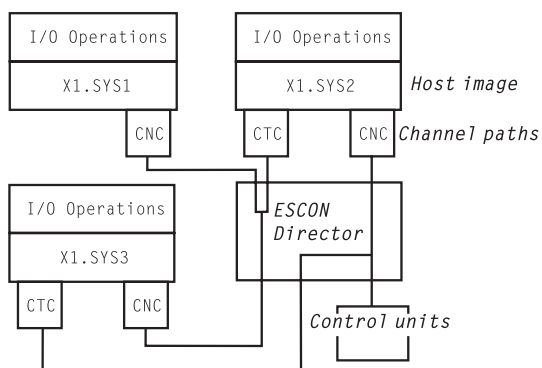
## Channels

A channel is a path that transmits signals and data. It can be a standard channel or a special purpose channel such as OSA (open system adapter) or a coupling facility sender or receiver channel. For channels operating in block mode, you can display and change the state of the channels. The types of channels with ESCON/FICON switches supported by I/O operations are shown in the following list:

**CNC** A channel path that transfers data between a host processor image and an ESCON/FICON control unit. This path from a channel to a device may be point to point or switchable.

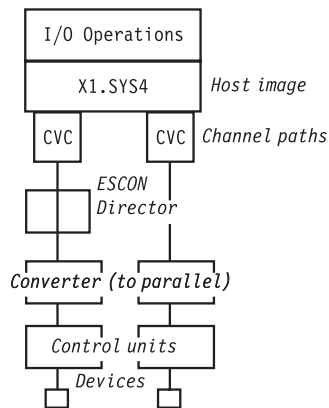


*CTC* The channel-to-channel (CTC) channel can communicate with a CNC on another host for intersystem communication.



*CVC and CBY*

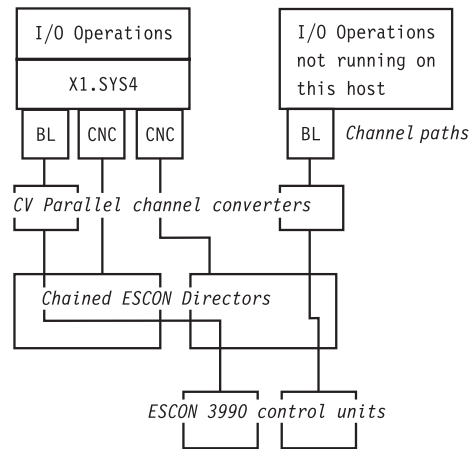
A channel operating in converted (CVC) mode transfers data in blocks and a CBY channel path transfers data in bytes. Converted CVC or CBY channel paths can communicate with a parallel control unit. This resembles a point-to-point parallel path and dedicated connection, regardless of whether it passes through a switch.



*Parallel*

Parallel channels operate in either byte (BY) or block (BL) mode. You can change connectivity to a parallel channel operating in block mode.

## Hardware Overview



### CFR/CFS and ISC/ISR

The I/O operations component of SA z/OS can display connectivity data and control connectivity for integrated system channels (ISC) connected to a coupling facility. This includes coupling facility receiver (CFR) channels and coupling facility sender (CFS) channels.

## Processors, Systems

The SA z/OS processor operations component monitors the status of target processors (including logical partitions) and target systems. The status is determined based upon messages and alerts from the target hardware that processor operations has received at the focal point system.

Alerts are high-priority events that require operator action. Alerts are issued for situations in the target hardware and target system, and also for situations in the supporting processor operations environment. The following alerts will result in a status change of a resource, which will in turn cause a change in the representation of that resource in the SA z/OS displays, thus alerting you to a potential problem.

- Target hardware problem
- Wait state detected
- Disabled Console Communication Facility (DCCF) message detected
- Console lost
- Console access lost
- Communications task failed
- System console could not be obtained

For OCF-based processors, processor operations automates support element-generated alerts to update status of resources viewable on SA z/OS panels and graphic displays.

Messages received by processor operations at the focal point system allow processor operations to update the status of a target system as it initializes. You can view the initialization status of a target system using processor operations panels. Examples of some of the initialization statuses that can be seen are: Stage-1 Activate complete, IPL complete, IPL failed, Load failed, Waiting for IEA101A Start Message (z/OS target systems), and Waiting for VM Start Message (VM target systems).

## I/O Devices

On a dynamically updated resource display, the status of an I/O device is updated when a message to the operator console or to the z/OS system log is trapped at the target system.

---

## Automation Policy

### Resource

Any target system, application, job, or associated hardware capable of being monitored or automated by SA z/OS.

### Automation operators

Autotasks or unattended NetView tasks that do not require a terminal or logged-on operator. Automation operators may receive messages and process automation procedures.

### (NetView) automation procedures

A sequence of commands, packaged as a NetView command list, a REXX™ procedure, or a command processor written in a high-level language. An automation procedure performs automation functions and runs under the NetView program. Automation procedures may exploit SA z/OS generic routines.

### NetView automation table

A table that the NetView program compares incoming messages against. A match with an entry triggers the specified response.

### Command processor

A module designed to perform a specific function. Command processors (that can be written in assembler or a high-level language) are invoked as commands.

### Command list (CLIST)

A list of commands and statements written in the NetView command list language or REXX. The list performs specific functions for a user. It may be as simple as a list of commands, or complex and include variable substitution and conditional logic (more like a conventional program). Command lists are typically interpreted rather than compiled.

### REXX procedure

A command list written with the Restructured Extended Executor (REXX), which is an interpretive language.

### Policy database

The automation definitions (automation policy) that the automation programmer specifies using the customization dialog is stored in the policy database. The automation programmer also uses the customization dialog functions to build the system operations control files (ACF and AMC file) from the policy database. Another file built from the information contained in the policy database is the processor control file. These files that are derived from the policy database define how SA z/OS applies automation to the resources.

### Applications

In SA z/OS, applications refer to z/OS subsystems, started tasks or jobs automated and monitored by SA z/OS. On SNMP-capable processors, "application" may be used to refer to a subsystem or process.

### Application groups

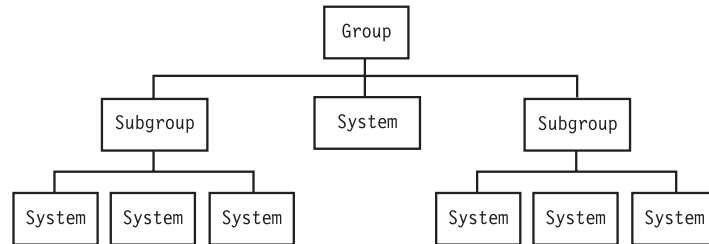
Using the SA z/OS customization dialog, you can group applications into

## Automation Policy

application groups. Application groups are linked to a system or sysplex. Thus you can have SA z/OS automate applications that belong or depend on each other in one unit.

### Grouped resource objects

Systems can be organized into groups or subgroups to simplify working with them and displaying them on panels. They can be grouped this way:



System groups can either be *standard* or *sysplex*, but processor operations treats them both the same. Applications can be grouped for being automated as an entity.

### Status display facility (SDF)

An SA z/OS facility that provides a 3270-based display of automated systems and resources using assigned status colors. An operator can monitor the status of automated systems and resources by viewing the SDF main panel.

### Monitor Resource

Monitor Resources enable you to monitor the performance and health of an application.

---

## Chapter 2. Concepts

This chapter contains information that you need to know while working with SA z/OS. The following topics are discussed:

- “Automation Concepts”
- “Controlling Application Move and Server Management” on page 23
- “SA z/OS Automation Flags” on page 33
- “Statuses Supplied by the Automation Manager” on page 35
- “Statuses Supplied by Automation Agents” on page 40
- “Using SA z/OS for Monitoring” on page 52

---

### Automation Concepts

In SA z/OS, the automation function is split up, as follows:

- The observing, reacting and doing parts are located in 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
- Within each sysplex, the coordination, decision making and controlling functions are gathered into a single address space outside of NetView. This address space is called the *automation manager*.

You define the resources that you want to automate with SA z/OS using the customization dialog. The automation manager contains a model of all of the automated resources within the sysplex. The automation agents are the automation manager’s eyes and arms. They feed the automation manager with status information and perform the actions that the automation manager tells them to. The automation manager is the brain of automation. It does all of the decision making that involves interaction between one or more resources.

The automation manager provides *sysplexwide* automation. Its main purpose is to provide one central point of book-keeping of all resources within a sysplex, that is, the automation manager knows about the following:

- Grouping of resources
- Dependencies between resources
- Statuses of resources
- Goals for resources, that is, what resources are to be made available or unavailable and when

According to the available information, the automation manager makes decisions and instructs the corresponding automation agent where the resource is located to put the resource into the desired state.

The automation manager knows several different statuses for each resource:

- The observed status
- The desired status

## Automation Concepts

- The automation status
- The startability status
- The compound status
- The health status

More detail about these statuses is provided in “Statuses Supplied by the Automation Manager” on page 35.

Decision making is done by the automation manager with the help of *goals*. Goals can be defined either permanently by the automation programmer who creates an automation policy using the customization dialog (see *IBM Tivoli System Automation for z/OS Defining Automation Policy*), or interactively by operators who issue commands.

In either case, the automation manager is informed about the goals of a certain application or resource, that is, when the resource should be up or down (available or unavailable). Goals also take into account the dependencies of resources as defined via relations in the customization dialog. They also check whether a trigger is defined for a resource that determines whether the availability of a resource depends upon some external events outside SA z/OS automation.

Automation performed by SA z/OS is thus also called *goal driven automation*.

Information about goals is provided to the automation manager in the following ways:

### **1 OPERATOR**

By the operator command INGREQ (see “How Requests Become Goals” on page 19). These requests are labelled with originator OPERATOR(USERID) in the INGVOTE command dialog.

### **2 AUTOOPS**

By the INGREQ command issued from automation procedure (REXX exec, CLIST). These requests are labelled with originator AUTOOPS in the INGVOTE command dialog.

### **3 E2EMGR**

By requests from the end-to-end automation manager. These requests are handled by the SA z/OS end-to-end automation adapter, which maps the requests to SA z/OS commands.

### **4 EXTERNAL**

By TWS Automation; if TWS Automation interfaces with SA z/OS, TWS will take care of the startup of supporting resources required for applications that are scheduled with TWS. These requests are labelled with originator EXTERNAL in the INGVOTE command dialog.

### **5 SCHEDULE**

There are three different ways that schedules can be used:

#### **Resource override**

The INGSCHED operator command with a resource parameter will interactively override a service period specifically for the specified resource (see “Resource Override” on page 84). These requests are labeled with the originator SCHEDULE in the INGVOTE command dialog.

#### **Schedule override**

With the INGSCHED operator command, which can be used to



pseudo-interactively override service periods for all resources that use this service period (see “Schedule Override” on page 84). These requests are labeled with the originator SCHEDULE in the INGVOTE command dialog.

### Automation policy

By means of this as defined in the customization dialog. The automation programmer uses *service periods* for applications or application groups as an easy method to let SA z/OS automatically schedule up and down times for these resources. Thus these service periods are also referred to as *schedules*. These requests are labeled with the originator SCHEDULE in the INGVOTE command dialog.

For most of these methods, you can specify one of the following priorities for a goal that is checked in the case of two conflicting goals:

1. LOW
2. HIGH
3. FORCE (not applicable for schedules)

Table 2 shows the hierarchy of all possible requests. The strongest goal that you can specify for a resource is interactively to issue an INGREQ command to make a resource unavailable, such as:

```
INGREQ AGFG2/APL/AOC7 REQ=STOP PRI=FORCE
```

The weakest goal you can define for a resource is to make it available with low priority via a service period.

A request to make a resource available is in most cases preferred to a request to make the same resource unavailable.

Table 2. Hierarchy of Requests

Rank	Priority	Method	Goal
1	FORCE	<b>1</b> OPERATOR	Unavailable
2	FORCE	<b>1</b> OPERATOR	Available
3	FORCE	<b>2</b> AUTOOPS	Unavailable
4	FORCE	<b>2</b> AUTOOPS	Available
5	FORCE	<b>3</b> E2EMGR	Unavailable
6	FORCE	<b>3</b> E2EMGR	Available
7	FORCE	<b>4</b> EXTERNAL	Unavailable
8	FORCE	<b>4</b> EXTERNAL	Available
9	HIGH	<b>1</b> OPERATOR	Available
10	HIGH	<b>1</b> OPERATOR	Unavailable
11	HIGH	<b>2</b> AUTOOPS	Available
12	HIGH	<b>2</b> AUTOOPS	Unavailable
13	HIGH	<b>3</b> E2EMGR	Available
14	HIGH	<b>3</b> E2EMGR	Unavailable
15	HIGH	<b>4</b> EXTERNAL	Available
16	HIGH	<b>4</b> EXTERNAL	Unavailable
17	HIGH	<b>5</b> SCHEDULE	Available

## Automation Concepts

Table 2. Hierarchy of Requests (continued)

Rank	Priority	Method	Goal
18	HIGH	<b>5</b> SCHEDULE	Unavailable
19	LOW	<b>1</b> OPERATOR	Available
20	LOW	<b>1</b> OPERATOR	Unavailable
21	LOW	<b>2</b> AUTOOPS	Available
22	LOW	<b>2</b> AUTOOPS	Unavailable
23	LOW	<b>3</b> E2EMGR	Available
24	LOW	<b>3</b> E2EMGR	Unavailable
25	LOW	<b>4</b> EXTERNAL	Available
26	LOW	<b>4</b> EXTERNAL	Unavailable
27	LOW	<b>5</b> SCHEDULE	Available
28	LOW	<b>5</b> SCHEDULE	Unavailable

### Notes:

1. A more recent request from the same source will however always overwrite an earlier request to the same resource without regard to the earlier request's priority. For example an operator can overwrite a HIGH priority INGREQ request to make a resource available with a later request of only LOW priority to make the same resource unavailable.

2. Within the same method, for all priorities except FORCE, the request or goal to make a resource available has a higher priority than to make it unavailable.

However, a *FORCE UnAvailable* request that is issued by an operator has the highest priority.

This is really the only method to beat SA z/OS's attempts to make a resource available if for any reason you want to have this resource down.

You can easily override your own FORCE Unavailable request for a resource with even a LOW priority request against the same resource. Thus this implementation of priorities ensures high flexibility and a quick reaction in emergency cases.

The automation manager transforms a goal (for example, the request that a certain resource (application) should be up) into an order to the corresponding automation agent where the application should run.

The automation agents therefore execute orders that come from the automation manager. While carrying out the automation, the automation agents also take information from the policy that is defined for the resources. This information is available in the automation control file on each automation agent. For example, for an order to start a resource that comes from the automation manager, the automation agents retrieve information about the appropriate startup command from the automation control file.

For enterprise monitoring, the automation manager has the task of gathering and controlling information about what resources are available, what the status of the resources is and what status updates occur during automation. The SA z/OS topology manager extracts and transfers this information to the enterprise-wide

RODM data base from where this information is picked up by the NMC workstation to display user-defined graphical views.

The communication between the automation manager and automation agents can be implemented with WebSphere MQ. This allows you to restart of the automation manager on a different system in case the automation manager terminates unexpectedly. This ensures that no requests are lost.

The main commands you can use to retrieve information from the automation manager are:

- INGLIST displays detail information about one or more resources (subsystem, application group, etc.)
- INGINFO displays lots of details for an individual resource or application group
- INGVOTE displays the requests that have been issued and are currently pending for a specified resource
- INGSCHED displays information about the current UP and DOWN service periods for resources
- INGGROUP displays the members of a group and their settings.
- INGRELS displays the relationships that are defined for a resource.

You can issue a request to the automation manager with the INGREQ command, such as that the goal for the specified resource should now be available or be changed to unavailable.

Requests that are sent to the automation manager with the INGREQ command are persistent. That is, if the automation manager terminates and is restarted later, it then remembers all requests (goals) that were valid when it terminated. The automation manager will then continue to pursue all these goals for the resources. This means that if a resource should run on a certain system, and this system fails and is restarted later, the automation manager will continue to pursue the specified goals across IPL times unless they conflict with the IPL schedule times.

### Summary of terminology:

**Goal:** There are only two goals for a resource: available or unavailable. Note that the success of carrying out a goal is also influenced by the dependencies of resources (defined as relationships in the customization dialog) and by triggers.

**Note:** If no goal is explicitly specified, the default goal is: make the resource available.

**Request:** If an operator specifies a goal interactively using the INGREQ command, this is called a request. According to the two goals for a resource, there are the two requests to achieve these: MakeAvailable and MakeUnavailable.

**Order:** An order is the result of a specified goal. An operator can specify goals interactively using the INGREQ command, or an automation programmer implicitly defines goals as automation policy by means of service periods.

The automation manager transforms such a goal, for example, the request that a certain resource (application) should be up, into an order to the corresponding automation agent where the application should run.

**Vote:** Resources can have relationships to other resources. As an example, a dependent resource needs a supporting resource before it can be made available. See *IBM Tivoli System Automation for z/OS Defining Automation Policy* for information on how to define relationships. If an operator generates a request to start a dependent resource, the automation manager internally generates another request to make the supporting resource available as well. Or, alternatively in an INGREQ command to shut down a resource, the operator can also request the shutdown of the dependent resources, which internally generates a request on these dependent resources. Such an internally generated request is called a *vote*.

It may happen that votes conflict or compete with each other. Such a scenario is illustrated in "Conflicting Relationships and Requests" on page 20. In such a case, the vote with the highest priority will win. Even if votes or requests with the same priority come into conflict, the requested action decides which request will win: the request to make a resource available is higher than the request to make that resource unavailable. To make a resource unavailable when it has the request to make it available, you must send it a higher priority request or revoke the MakeAvailable request.

Using the INGVOTE command, you will see all currently active requests with their priorities, may they have been generated explicitly by an operator, implicitly because of the automation policy or internally as a consequence of relationships.

One automation manager must exist on each sysplex and one on each standalone system. In a sysplex, you can have one or more secondary automation managers (SAM) for backup purposes. That is, if the system that the primary automation manager (PAM) runs on, abends or if the automation manager itself abends, the first secondary automation manager recognizing this, will become the new PAM

and will take over all current requests from the former PAM. The automation agents in the sysplex will not notice the switch from one automation manager to the other.

**Note:**  
No request will be lost.

## How Requests Become Goals

A request asks that one specific resource (its *target resource*) should be moved to a specific state (its *goal*). A resource may, for example, be an application that should run on a certain system.

Consider the following example:

```
INGREQ abc REQ=START
```

This specifies that SA z/OS should start the resource *abc*, that is, SA z/OS should set the desired status of this resource to AVAILABLE on a system.

SA z/OS knows the dependencies for each of the actions that can be taken against the resource through the resource's relationships with its supporting resources. Using these, it propagates the request onto those resources. These internally created requests are called *votes*. Let us have a look at a CICS scenario: It makes sense to start a Terminal Owning Region (TOR\_1) only after an Application Owning Region (AOR\_1) is up. So you could define TOR\_1 (running on a certain system) in our example to become a dependent resource and AOR\_1 to become the supporting resource. So if you request to start TOR\_1, SA z/OS would generate a vote to start AOR\_1.

Start requests are generally propagated from dependent resources up to supporting resources so that the supporting resource is started before the dependent resource is started. Stop requests are propagated down to dependent resources in order to stop dependent resources first.

## How Request Propagation Works

If the request is a higher priority than whatever request the supporting resource currently has, the new request wins and the supporting resource's goal is changed to comply with the request. The process then repeats itself for the supporting resource to its supporting resources.

Eventually the propagation will stop if either SA z/OS does not encounter another supporting resource or it encounters higher priority requests. At this point, SA z/OS knows which resources are not in their goal state. SA z/OS will issue start or stop commands for those resources whose prerequisites for being started or stopped are met.

Once a resource's goal has been set, SA z/OS will continue trying to achieve that goal. This means that if the resource abends and its desired status is still AVAILABLE, SA z/OS will try to restart it. If the goal changes to UNAVAILABLE, SA z/OS will stop the resource. If a resource starts when its goal is UNAVAILABLE, SA z/OS will also stop it.

## Conflicting Relationships and Requests

One of the most effective automation capabilities of SA z/OS is the concept of *dependency relationships*. Using the customization dialog, the automation programmer defines relationships and conditions between dependent resources and supporting resources. For information on how to achieve this, refer to *IBM Tivoli System Automation for z/OS Defining Automation Policy*.

In complex environments, it can happen that relationships exist that are opposed in both direction and meaning, as illustrated in Figure 2. The following abbreviations are used in this diagram:

- For dependency relationships:

**MaWa** denotes MAKEAVAILABLE/WhenAvailable

**MuWu**

denotes MAKEUNAVAILABLE/WhenUnavailable

- For statuses:

**Unav** denotes Unavailable

**Avail** denotes Available

- For requests or votes:

**Av** denotes a request or vote to become available

**Un** denotes a request or vote to become unavailable

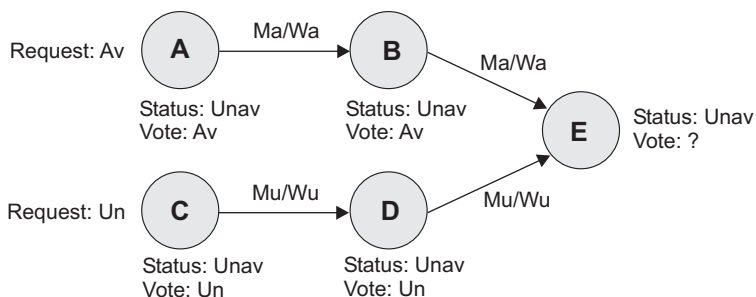


Figure 2. Conflicting Relationships and Requests

In Figure 2, resource A is in status Unavailable (Unav) and gets the request to become available (which internally creates the corresponding vote).

So the relationship MAKEAVAILABLE/WhenAvailable propagates requests to make resources available and the relationship MAKEUNAVAILABLE/WhenUnavailable propagates requests to make resources unavailable.

**Note:** The first part of a dependency relationship always applies to the resource where the arrow starts, and the condition always applies to the resource where the arrow ends.

As requests are internally stored and propagated as votes, in Figure 2, you see two conflicting votes that would be propagated to resource E. From A we have the vote to be available, while from resource C we have the vote to be unavailable. E has received both of these votes, but cannot satisfy them at the same time. You could see the conflicting votes for C by issuing the INGVOTE command against this resource.

In cases when two separate streams of propagation meet, the priority of votes/requests is evaluated and used to determine the winning vote. A priority is assigned to a request either by the operator interactively using the INGREQ command or by the automation programmer setting priorities for service periods.

A request to make a resource available wins over a request to make it unavailable with the same priority. This is only valid due to the *propagation* of requests. However, if an operator sends a request to start a resource, and the same or another operator later sends another request to stop this resource, the latter one will win. The difference is, that this is not a *propagation* of requests.

### How the Automation Manager and Automation Agents Cooperate in a Sysplex

As described in “Automation Concepts” on page 13, SA z/OS’s automation logic in a sysplex is divided between the automation agents and the automation manager.

For reasons of high availability, SA z/OS also offers the concept of the primary and secondary automation manager (PAM and SAM), where the SAM can take over the work of the PAM without loss of any information or without any requests being processed twice.

In a sysplex, the automation agents communicate with each other using XCF. Automation managers communicate with the automation agents using WebSphere MQ or XCF.

SA z/OS provides two options for establishing communication between the automation manager and the automation agents, using:

- WebSphere MQ queues for both communication and backup
- XCF for communication and a VSAM data set for backup

Figure 3 on page 22 and Figure 4 on page 22 show a typical configuration in a sysplex, showing the communication paths and methods between its members. Automation managers and automation agents communicate via WebSphere MQ using the coupling facility. Automation agents communicate via XCF. XCF is only used at initialization time or to determine if an automation agent leaves or joins the sysplex. For more detailed information on how WebSphere MQ queues are used to perform the communication refer to *IBM Tivoli System Automation for z/OS Planning and Installation*.

## Automation Concepts

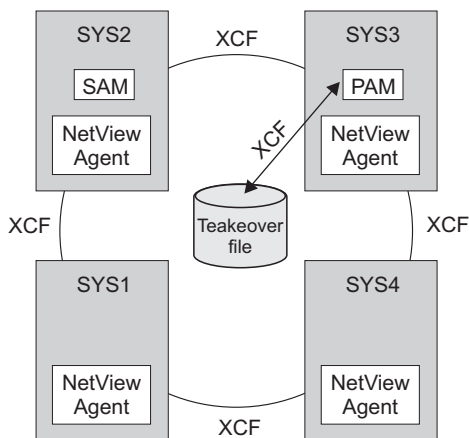


Figure 3. Automation Manager and Automation Agents Cooperation in a Sysplex using XCF

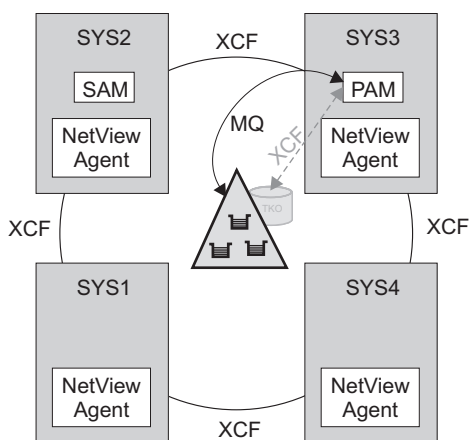


Figure 4. Automation Manager and Automation Agents Cooperation in a Sysplex using MQSeries®

## Locked Votes and Requests

When the automation manager tells an automation agent to start or stop a resource, internally, it locks the current winning vote on the resource. The automation agent is not generally expected to be able to stop an application from halfway through its start sequence, nor to be able to start an application if it is halfway through its stop sequence. This is why the winning vote is locked: To prevent it from changing and thus causing the automation manager to send the automation agent an order for a contrary action.

Occasionally the request that originated a locked vote will be deleted. In this case the automation manager automatically generates a copy of it (a LOCK request) that then becomes the owner of the remaining locked requests from the original vote. Votes from these requests show up on INGVOTE with a LOCK- prefix.

Once the action has been completed (or has failed) the lock on the winning vote is released, allowing it to be removed and replaced by a vote from the currently active requests. Once all locked votes from a locked request have been unlocked and removed, the locked request is automatically deleted.



You can use the interrupt option of the INGREQ command (that is, issue INGREQ with INTERRUPT=YES) for any start processes that are running. This forces the lock on the winning MakeAvailable vote to be released, allowing it to be replaced by your newly added request. If you use this to turn a start into a stop, it is essential that you have configured the subsystem's shutdown commands to be able to correctly shut down the subsystem from whatever state the startup may have left it in.

---

## Controlling Application Move and Server Management

*Application move* refers to the process of stopping an active set of resources and starting an alternative set in a coordinated fashion.

*Server management* refers to the starting and stopping of equivalent resources to maintain a desired level of server capacity. Although it will work with any type of resource, it is designed to work with servers that are loosely coupled to the applications using them. Loosely coupled in this sense means that applications using the servers will not be adversely impacted if one instance of the server group is stopped and another one is started.

SA z/OS uses the mechanism of grouping resources as described in *IBM Tivoli System Automation for z/OS Defining Automation Policy* to implement the move mechanism, namely the groups of nature MOVE and SERVER obey all move instructions described later in this section, along with an availability target instruction that specifies, how many members of a group should be available, if not all are required.

### Move Functions

The move functions that you can specify against a group are:

- Whether you want new members to be started at the same time as old members are being stopped (*parallel move mode*), or the old member should be stopped completely before the new member is started (*serial move mode*).
- Automatic group management to maintain application availability. This includes the activation of backup/alternate resources in response to resource or system failure.
- Moving resources away from a system en masse, at either a group or subgroup level or a system level.
- Querying and dynamically changing a group's policy to stop one instance of a resource and start another.
- Estimating the impact of those changes before committing them.
- Performing actions resulting from the above changes in a delayed, non-disruptive fashion.
- Returning resources to their normal systems in a delayed, non-disruptive fashion.

The INGGROUP command offers the following methods to apply move functions to a group:

#### EXCLUDE/AVOID/INCLUDE

This indicates that you want to remove movable members from a system (exclude), prohibit SA z/OS from activating movable members on a system (avoid) or that you wish to undo either of the other two actions (include).

## Controlling Application Move and Server Management

### Availability Target (AVTGT)

This specifies the availability target that is to be set for the group. The value is the number of the group members that should be made available when the group has to be available. You can dynamically overwrite a group's availability target that is statically defined in the automation policy.

### Preference (PREF)

This parameter specifies preference values for being available for the members of a group. You can dynamically overwrite the preference values that are statically defined in the automation policy for each member. More information about the preference value is provided in "Controlling Application Move with Preference Values."

### Satisfactory Target (SATTGT)

This parameter 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 dynamically overwrite a group's satisfactory target that is statically defined in the automation policy.

## Controlling Application Move with Preference Values

Controlling application move and server management is done with preference values. Preference values range from 0 through 3200. They express the importance of each individual member of the group.

Note that only MOVE groups and SERVER groups support preference values. With BASIC groups you cannot specify preferences, because all members need to be available. For MOVE and SERVER groups, the preference defines which resource members of the group should preferably run. Remember that a MOVE group is designed for backup solutions. A MOVE group can have many components, but one and only one is supposed to be active. All other components are for backup only. Thus MOVE groups usually contain members of one kind (where one application can replace the others).

SERVER groups are created when only a certain number of their resources need to be available for the application group to be available. The number of desired available resources is controlled by the *Availability Target* parameter in the customization dialog. You can use the preference value in the customization dialog to control which resources you want to be available and which you want to keep as backups.

The effective preference of each group member is calculated to determine which group members get picked to be activated by the group. The primary inputs into this are the base preference (as specified through the customization dialogs) and the preference adjustment (specified indirectly through the INGGROUP interface). These are added together to produce the resource's adjusted preference (this is the same as the preference value you actually enter the INGGROUP panels).

The following modifiers are then applied to the resource's adjusted preference if it is >1.

Bonus	Reason
25	Member is currently selected and has an adjusted preference $\leq 1000$ .
175	Member is currently selected and has an adjusted preference $> 1000$ .

## Controlling Application Move and Server Management

Bonus	Reason
220	Member has observed status of Starting or Degraded and the system is not being stopped.
225	Member has observed status of Available or Stopping and the system is not being stopped.
-400	Member has an adjusted preference >1500 and either the system is being stopped or the system has already been stopped and the member thus has observed status of SysGone.

The working value for the preference may then be set to an absolute value if any of the following conditions are satisfied:

1. A member with an effective preference of 1 may never be picked. It must always be sent a MakeUnavailable vote.
2. A member with an effective preference <600 may only be picked if the group is in recovery mode (that is, it is recovering from a failure).
3. A member requiring a resource on an excluded system must be sent a MakeUnavailable vote, even if it has been picked. This can occur with resources that have adjusted preferences >2600.

### Available Preference Values

The preference values shown in the following tables are adjusted preference values (the sum of the base preference from the ACF and any preference adjustment specified through INGGROUP).

*Table 3. Preference Values and Behavior of Members*

Adjusted Preference	Behavior
3100	The resource will always be selected, even if it has an observed status of HardDown, or SysGone, its system is stopping or its system has been excluded.
2900	The resource will always be selected unless it has an observed status of SysGone or its system is stopping and there is a viable alternative resource.  It will still be selected if its system has been excluded or it has an observed status of HardDown.
2400	The resource will always be selected unless it has an observed status of SysGone or HardDown, or its system is stopping and there is a viable alternative resource. It will still be selected if its system has been excluded.
1000	The member is always selected when the group has been started and the system associated to this member comes up. Other resources from the same group with preference 700 are stopped in favor of this member.
900	The member is always selected when the group has been started and the system associated to this member comes up. Members with preference 500 will be stopped in favor of this group.
700	The member is always selected when the group has been started and the system associated to this member comes up. Members with preference 300 will be stopped in favor of this group. It is not stopped unless a very high-preference alternative (1000) becomes available. This value is the default. Use it in a MOVE group for the resource that normally runs.
<600	This value is used as a threshold. A resource with preference 600 and lower will not be started at group startup. Only picked if the group is in recovery mode.

## Controlling Application Move and Server Management

Table 3. Preference Values and Behavior of Members (continued)

Adjusted Preference	Behavior
500	This resource is not selected but will be selected when the group enters recovery mode. Once the problem is over, it will be stopped as soon as a primary member with a calculated preference >750 is available.
300	This resource is not selected but will be selected when the group enters recovery mode. Once the problem is over, it will be stopped as soon as a primary member with a calculated preference >550 is available.
100	This resource is not selected but will be selected when the group enters recovery mode. As soon as a preference 500 backup member is available, it will be stopped in favor of the backup member.
1	A resource with this preference will never be selected by SA z/OS automation; it is always sent an Offline vote. Operators may make such resources available interactively by using the INGGROUP command and giving it a higher preference.  Use this value to provide completely manually controlled application availability.
0	This value indicates that the member is passive and will never have a vote propagated to it. If a member is made passive, any votes that have been propagated to it will be withdrawn.

### Preference Scenarios

To understand the interactions of the various flavours of preference values and the concept of the recovery mode, the following sections describe some scenarios of how SA z/OS reacts according to the specified and automatically applied preferences.

**Backup with Non-Disruptive Restoration:** In this scenario you have an application ABC that can run on one of two systems: SYS1 and SYS2. Only one instance of the application should normally be active. The instance on SYS1 is the primary instance (set preference 700), while the instance on SYS2 is the backup instance (set preference 500).

The preferences 700 and 500 mean that the applications will behave as follows:

1. If SYS2 is IPLd first, the application will not be started (preference 500, not in recovery mode).  
When SYS1 is IPLd, its instance of the application will be started (preference 700, highest startable instance).
2. If SYS1 should fail, the instance on SYS2 will be started (preference 500, in recovery mode, highest startable instance).
3. When SYS1 is IPLd again, it will not start its instance (preference 700, but active instance with preference 750 (500+250 for being available)).
4. At the end of the service period the instance of SYS2 will be stopped.
5. At the start of the next service period, the instance of SYS1 will be started (preference 700, highest startable instance).

**Backup with Disruptive Restoration:** This scenario is like the above, but you want the application to return to SYS1 as soon as SYS1 is available, even if this causes an additional service interruption. You can use preferences of 700 for SYS1 and 300 for SYS2, or 900 for SYS1 and 500 for SYS2. As these are the only two

## Controlling Application Move and Server Management

applications involved, either combination will work (the important thing is that they are more than 250 apart). The example uses the 700 and 300 values.

1. If SYS2 is IPLd first, the application will not be started (preference 300, not in recovery mode).  
When SYS1 is IPLd, its instance of the application will be started (preference 700, highest startable instance).
2. If SYS1 should fail, the instance on SYS2 will be started (preference 300, in recovery mode, highest startable instance).
3. When SYS1 is IPLd again, its instance will be started (preference 700, highest startable instance) and the SYS2 instance will be stopped (preference 550 (300 + 250 for being available), lower than 700). This causes a short service interruption as the application moves back to SYS1.
4. At the end of the service period the instance of SYS1 will be stopped.
5. At the start of the next service period, the instance of SYS1 will be started (preference 700, highest startable instance).

Figure 5 illustrates the difference of non-disruptive and disruptive restoration between the steps 2 and 3 in each scenario: in the non-disruptive scenario, the application ABC continues to run on SYS2 until the end of the service period for the group, because this resource now has a higher priority of 750 than the resource on SYS1, which would only have priority 700. In the disruptive restoration on the other hand, SYS1 takes over for the rest of the service period after the problem has been fixed.

	Non-Disruptive Restoration		Disruptive Restoration	
Initial	SYS1 700	SYS2 500	SYS1 700	SYS2 300
1. IPL	SYS1 950	SYS2 500	SYS1 950	SYS2 300
2. SYS1 fails	SYS1 700	SYS2 750	SYS -	SYS2 550
3. SYS1 IPLd	SYS1 700	SYS2 750	SYS1 950	SYS2 300
4. End of service period	SYS1 700	SYS2 500	SYS1 700	SYS2 300
5. New service period	SYS1 700	SYS2 500	SYS1 700	SYS2 300

Figure 5. Non-Disruptive and Disruptive Restoration

**Real Hierarchy of Preferences:** In this scenario, you have three systems where the application can be run. SYS1 is where you normally run it, SYS2 is a back up and SYS3 is where it can run in extreme emergency only. You would specify preferences as 700, 500 and 100.

1. If either SYS2 or SYS3 are IPLd first, their instances will not be started (preferences 500 and 100, not in recovery mode). The operator can, of course,

## Controlling Application Move and Server Management

explicitly request one of the instances to be started (which should be done if they want the application to be available and know that SYS1 will not be IPLd for a while).

2. When SYS1 comes up its instance will be started (preference 700, highest startable instance).
3. If SYS1 fails, the SYS2 instance will be started (preference 500, in recovery mode, highest startable instance).
4. If SYS1 were to return at this point, we would be back in the first scenario, a non-disruptive backup.
5. Instead, let us suppose that SYS2 now fails. This causes the instance on SYS3 to be started (preference 100, in recovery mode, highest startable instance).
6. If either of the other two systems came back (let us assume it is SYS2), its instance will be started (preference 500, in recovery mode, highest preference instance) and the instance on SYS3 will be stopped (preference 350 [100 plus 250 for being available], lower than 500). This would cause an application outage, but would get the application off SYS3.
7. If SYS1 then comes back, its instance will not be started: Preference 700, lower than 750 (500+250 for being available).
8. After the next service period stop, we would be back to running the instance on SYS1.

**Note:** If SYS1 was down at the end of the start of the next service period, the application would not be restarted. Using preferences of 900, 700, 100 (or 300) would get around this, but also means that the application could be automatically started on SYS2 if it was IPLd before SYS1.

**Server Battery with Backups:** In this scenario your sysplex consists of four systems (SYS1, SYS2, SYS3 and SYS4). You defined a Sysplex Application Group of nature SERVER with an availability target of 3 that should normally run on systems SYS1, SYS2 and SYS3. A backup resource is defined on SYS4, but it should only be used when one or more of the main servers is unavailable. Preference of 700 should be used for the resources on SYS1, SYS2 and SYS3, while the backup resource on SYS4 should have a preference of 300.

1. When all systems are IPLd, the resources on SYS1, SYS2 and SYS3 will be started (preference 700, 3 highest startable resources). The resource on SYS4 will not be started (preference 300, not in recovery mode).
2. If either SYS1, SYS2 or SYS3 fails, the resource on SYS4 will be started (preference 300, now in recovery mode).
3. When the failed system is restored, its resources will be restarted (preference 700, three highest startable resources) and the resource on SYS4 will be stopped (preference 550 [300+250 for being available]).

**Note:** Setting a preference of 500 for the resource on SYS4 would cause it remain running when the failed system returned, effectively turning the resource on the returned system into the backup. It is also possible to set the preferences as 700, 700, 700, 690, which will let automation start the resources on whichever systems it needs to make the application available, but to prefer not to use the SYS4 resource if the other three are available (which they may not be at IPL time).

**Move Groups and Recovery Mode:** In this scenario your sysplex consists of four systems (SYS1, SYS2, SYS3 and SYS4). You defined a Sysplex Application Group of nature MOVE (availability target of 1) that should normally run on system SYS1 (that is, preferences of 700 for SYS1, 450 for SYS2, 250 for SYS3, and 1 for SYS4).

## Controlling Application Move and Server Management

1. When all systems are IPLd, the resource on SYS1 will be started (preference 700, highest startable resource). The resources on SYS2, SYS3, and SYS4 will not be started (availability target of 1 and APG not in recovery mode with preferences  $\leq 600$ ).
2. If SYS1 should fail then we enter recovery mode, which means that SYS2 and SYS3 become eligible to be started. In this case SYS2 will be chosen (highest eligible preference 450). If however, SYS1 was a planned shutdown (SHUTSYS ALL) then we would not be in recovery mode and neither SYS2 nor SYS3 would be eligible, so the application would not be moved automatically. To move the resource to another system use INGGROUP to raise that system's preference to  $>600$ .
3. If SYS2 should fail, as well as SYS1, the resource will be started on SYS3 without further adjustment (minimum startable preference 250).

**Note:** SYS4 has not participated in any of these MOVE examples (preference value 1, non-startable instance).

**Absolute Manual Control:** Member to be activated gets preference 3200, all others get preference 1.

The member with the 3200 preference will be selected. All others will always have a MakeUnavailable vote propagated to them.

This does not change even if the system with the selected member leaves the sysplex.

**Move Only After System Failure or on System Shutdown:** Primary member defined preference 2800, backups' preference 500.

As long as the primary is neither SYSGONE nor being shut down it will be selected. If the primary is SYSGONE or is being stopped, its preference drops to 2575 ( $2800-400+175$ ), allowing an alternative viable member to be selected.

The backups will only be viable if the resource was previously active because of the groups recovery mode.

**Move Only If System Down or Being Stopped:** Primary member defined preference 2800, backups' preference 700.

As long as the primary member is neither SYSGONE nor being shut down it will be selected. If the primary member is SYSGONE or is being stopped, its preference drops to 2575 ( $2800-400+175$ ), allowing an alternative viable resource to be selected.

The backups are always viable.

**Sticky Move Only After System Down or System Shutdown:** Primary member defined preference 2801, backups' preference 2800.

As long as the primary member is neither SYSGONE nor being shut down, it will be selected. If the primary becomes SYSGONE or is being stopped, its preference drops to 2576 ( $2801-400+175$ ), allowing one of the backups to be selected. The sticky bonus raises its preference to 2975 ( $2800+175$ ), meaning that even when the primary returns the selected backup system will remain selected.

## Controlling Application Move and Server Management

It will only be deselected if it goes to SYSGONE or is shut down, in which case the primary will be re-selected if it is available otherwise another backup would be picked.

**Move Only After System or Application Failure:** Primary member defined with preference 2400, backups' preference 500.

As long as the primary member is not SYSGONE or HARDDOWN it remains selected. If it is HARDDOWN or SYSGONE its preference is still <2600, so a viable alternative can be selected.

The backups are only viable if you are in recovery mode (that is, the member was active when the failure occurred).

This will not initiate a move if the primary system is being shut down. This is because the group will not enter recovery mode.

**Move Only If System Down, Being Shut Down or Application Failure:** Primary member defined with preference 2400, backups' preference 700.

As long as the primary member is not neither SYSGONE, HARDDOWN, nor being shut down it remains selected. If it is HARDDOWN or SYSGONE or is being stopped, its preference is still <2600, so a viable alternative can be selected.

**Sticky Move If System Down or Stopping, Move After Application Failure:** Primary member defined with preference 2401, backups' preference 2400.

As long as the primary is not neither SYSGONE, HARDDOWN, nor being shut down it remains selected. If it is SYSGONE or HARDDOWN or is being stopped, its preference is <2600, so a viable alternative may be chosen.

The sticky bonus raises the preference of the selected member to 2575 (2400+175), so it will remain selected until gets to a state of SYSGONE or HARDDOWN or is being stopped, at which point another viable alternative will be selected, with the primary member having preference.

The individual shutdown of the active member will not trigger a move.

**Sticky Move Group, Unless Member Stopped:** Primary member defined with preference 701, backups' preference 700.

As long as the primary member is startable, it remains selected. If it becomes non-startable, another member is selected and that remains selected as long as it is startable.

Individually stopping the active member will trigger a move. If the members were defined with preferences >1500, the bonus would be 175 points rather than 25 points.

**Sticky Move Group, Until Told Otherwise:** Primary member defined with preference 3200, backups' with preference 1600.

With base preferences of 3200, you get it picking a member when the sysplex is IPLd and retaining that one until an operator uses INGGROUP to make it pick another one.



Beware of timing problems as the first system with a member to register will probably get it.

### Resource Aware Application Move

Before carrying out an application move, the automation manager can now take into account the *workload* on systems. This enables SA z/OS to start members on systems with the highest amount of free capacity and will normally result in better performance of your applications.

Starting members on a system increases its workload and thus the free capacity of that system changes while SA z/OS makes its decisions. When multiple members are affected SA z/OS is able to revise its decisions while it processes start requests. This prevents a system that initially has the most free capacity being flooded with a large workload.

**Note:** Resource aware application move only takes place if the applications in a system or sysplex have the *same* effective preference values. It also only considers the *workload* in its decision making; it does not affect any preference values that you have defined for your resources.

This also means that if you have defined *different* preference values for applications in a system or sysplex, resource aware application move will not take place even if it has been enabled.

### Overview

Resource aware application move is achieved by the automation manager querying WLM for the capacity data of all systems in a sysplex. Capacity information is returned in SUs. These are hardware-independent numbers that describe the amount of capacity a system can provide or an application uses.

WLM provides the number of SUs used per system. This is divided by the number of resources on the system to give an average. The average number of SUs that a resource uses is known as the *resource weight*.

It should be noted that the resource weight is an estimate. This is because the capacity data that is returned by WLM relates only to those workloads (that is, resources that consume SUs) that are known to SA z/OS. The accuracy of the estimate is however only seriously affected when a system is full.

SA z/OS uses the capacity data and the resource weight to predict how a system's capacity will change before the next update from WLM is received. Whenever an application is started or stopped on a system, the total available free capacity of that system is recalculated. This new value is then used when SA z/OS decides where to start the next application. When an update is received from WLM this replaces the interim value that SA z/OS calculates. Thus the longer the query interval to WLM, the less accurate the free capacity calculation may become. A query interval of 5 minutes is considered to be a reasonable default.

Whenever SA z/OS moves an application, it also moves any dependent or related resources that you may have defined for it.

If members of a MOVE or SERVER group that are to be moved are *also* groups then the decision-making is slightly different:

1. If all the group's members are on the *same* system, the automation manager uses the expected free capacity of the system divided by the number of

## Controlling Application Move and Server Management

members of the group as the basis to decide which group to move first. This tends to favor groups with fewer members.

2. If the group's members are spread over *multiple* systems, it is not possible to calculate a reasonable capacity value and resource aware application move is not possible.

**Example Scenario:** Consider the sysplex in Figure 6. (Note that the SU values are for illustration only, normally they are 8- or 9-digit numbers.)

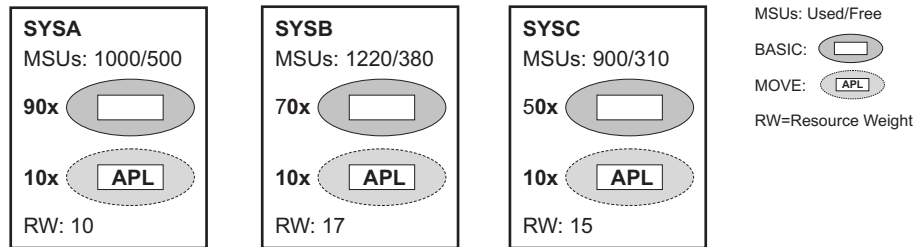


Figure 6. Resource Aware Application Move Example Sysplex

On each system a certain number of resources are members of BASIC groups. There are also 10 MOVE groups, each with one resource, on each system. These MOVE groups use the default preference of 700 for all of their members. Now suppose SYSA crashes. SA z/OS must move each of the 10 resources from the MOVE groups on SYSA to either SYSB or SYSC.

The initial free SU value of SYSB is 380 and of SYSC it is 300. SA z/OS first starts 5 resources, one after the other, on SYSB (see Figure 7). Now there is more free capacity on SYSC, so SA z/OS then starts alternately one of the remaining 5 resources on SYSC and SYSB. Each resource that is started reduces the free expected SUs by the resource weight (17 on SYSB and 15 on SYSC). This results in free SUs of 261 on SYSB and 255 on SYSC. Thus neither system is overloaded.

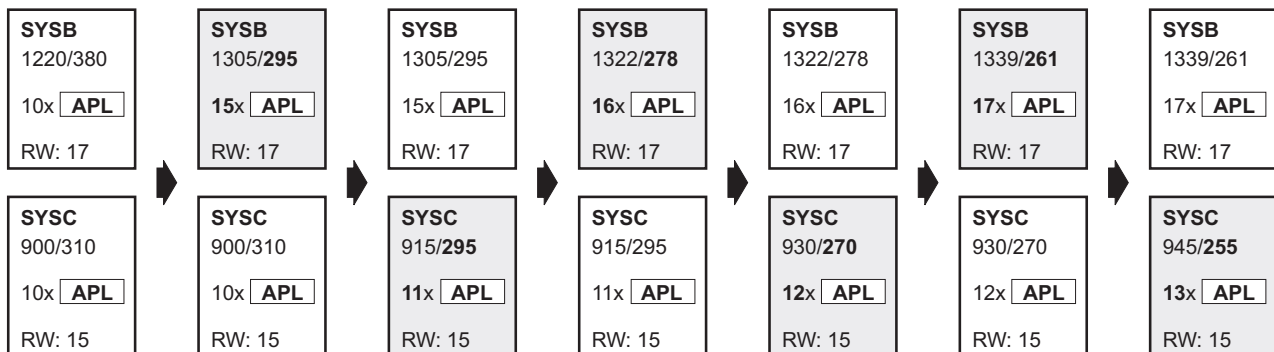


Figure 7. Resource Aware Application Move

### Enabling Resource Aware Application Move

To enable resource aware application move, you must customize the automation manager's PARMLIB member by adding a statement for the WLMQUERYINTERVAL keyword. For example, to have the automation manager query WLM every 5 minutes:

```
:
WLMQUERYINTERVAL=5
```

You can specify an interval between 0 (zero) and 600 minutes. If you specify 0 this disables WLM querying.

If you omit the WLMQUERYINTERVAL keyword, a default of 0 is used (that is, no querying of WLM is done).

### SA z/OS Automation Flags

Automation flags are automation policy settings defined in the customization dialog that specify whether certain types of automation are on or off for a resource. For information on how to set automation flags, refer to *IBM Tivoli System Automation for z/OS Defining Automation Policy*.

### Automation Agent Flags

If an event occurs that triggers automation, SA z/OS checks the appropriate flag to determine whether automation is currently on. If it is, SA z/OS proceeds with the automation actions that are associated with the flag. For example, if TSO fails and the Restart flag is on for TSO, SA z/OS restarts TSO according to the policy that is specified in the customization dialog.

Figure 8 shows how specific automation flags cover the phases in the lifetime of an application.

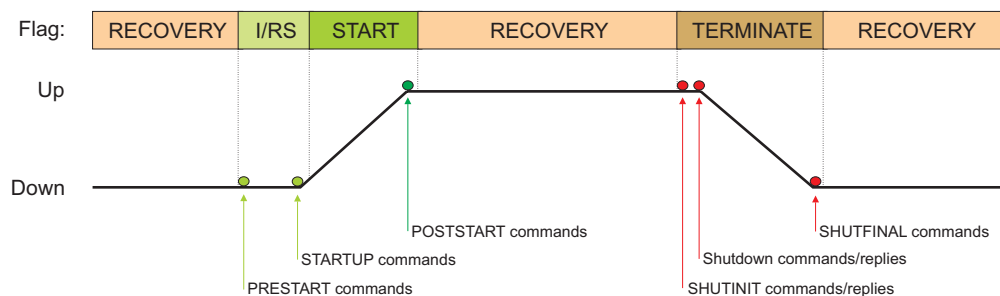


Figure 8. Automation Flag Span of Control

There is one flag that controls all the other flags. This is the Automation flag:

#### Automation

If this flag is set to N (No), no automation is allowed, 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.

The other flags that apply to each resource are:

#### Initstart

If this flag is turned on and the SA z/OS automation status for a subsystem is DOWN, SA z/OS can try and start the subsystem when all other conditions necessary for its startup (as defined in your automation policy) are met.

**Start** If this is on, automation manages and completes the startup process of the subsystem.

#### Recovery

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 corrective

## SA z/OS Automation Flags

actions in response to messages that indicate shortages of WTO buffers and JES spool space. Automation also invokes offload routines when the following MVS data sets are about to fill up: SYSLOG, LOGREC, SMF, MVS dump.

### Terminate

If this flag is turned on, SA z/OS can process automated shutdown requests for this subsystem. The setting of this flag can be overridden when the shutdown is requested. Also, the flag can be used to control secondary automation that occurs during the shutdown process. Secondary automation covers, for example, replying asynchronously to WTORS and issuing commands in response to messages from the subsystem.

### Restart

If this flag is turned on and the SA z/OS automation status for a subsystem is RESTART, SA z/OS can try and start the subsystem when all other conditions necessary for its startup (as defined in your automation policy) are met.

When you set an automation flag to Y in the customization dialog, that automation flag is on all the time, unless someone does one of the following:

- Turns the flag off for a period by scheduling an override in the customization dialog. For example, you could schedule an override that turns initialization and shutdown automation off on Sundays from 12:00 to 16:00, so you can do your system backups.

You can display these scheduled overrides using the DISPSCHD command dialog, as explained in “Displaying Automation Agent Flag Override Settings” on page 88.

- Turns the automation off using the INGAUTO command dialog. These command dialogs are discussed later in the chapter.
- Causes a higher level flag to be turned off, for example the subsystem’s Automation flag or the SUBSYSTEM Restart flag.

If a flag is set to N, you cannot schedule an override in the customization dialog to set it on for a particular time. You must use the INGAUTO command dialog.

If a flag is set to E, an exit is executed to determine whether automation is allowed. See *IBM Tivoli System Automation for z/OS Customizing and Programming* for details of flag exits.

If a flag is set to L, triggered automation actions such as commands or replies that this automation flag is checked for by SA z/OS are written to the NetView Log instead of being issued.

## The Automation Manager Global Automation Flag

Using the INGLIST or the INGSET command (see *IBM Tivoli System Automation for z/OS Operator’s Commands*) you can set an automation flag for the individual resources, which is checked by the automation manager before it sends any order to the automation agent to start or stop the specific resource.

The purpose of this flag is to prevent (if flag is set to NO) or enable (YES) the starting or stopping of resources from the point of view of the automation manager. This can be done for resources that reside on systems that are currently inactive, for example, to prevent the startup of the resource at IPL time of the system.

## The Hold Flag

As the default for all resources is to be available, you can use the *hold flag* for resources to prevent their startup at IPL time of the system.

The hold flag is primarily used for the NOSTART reply to the AOF603D WTOR.

You can manipulate the hold flag with the INGLIST or the INGSET command or just view it with INGINFO.

---

## Statuses Supplied by the Automation Manager

The automation manager keeps track of various statuses for each resource:

### Observed Status

This indicates the current status of the resource, as reported by the automation agent. See “Observed Status” on page 36.

### Automation Status

The automation status of a resource tells the automation manager what automation status a resource is currently in. If a resource’s automation status is currently *idle*, this means that the automation agent does nothing for automation, or conversely, if the automation status is *denied*, this means that automation is currently not allowed, possibly because the automation flag has been set off. See “Automation Status” on page 37.

### Desired Status

This status reflects the goal of a resource. The automation manager will be continuously trying to place the resource into this status unless the goal is revoked or changed. This status can be either *Available* or *Unavailable*.

If it is *Available*, the automation manager will try and start the resource. If it is *Unavailable*, the automation manager will try and stop the resource. The default is to try and make the resource available. See “Desired Status” on page 38.

### Startability Status

This is a calculation as to whether it is possible to start the resource if this is requested. It includes the startability of all inactive start dependent resources. This status is mainly used as input for Compound Status calculations and by groups to prevent them from selecting members that cannot be started. See “Startability Status” on page 38.

### Compound Status

This status is the composition of the other statuses to give the operator a chance to check from the automation manager if a resource is currently OK concerning the automation. See “Compound Status” on page 39.

### Health Status

The health status can be used by the automation manager to make decisions and, if necessary, trigger automation for the application if there is a change in the health state. See “Health Status” on page 39.

The operator can retrieve the status using two commands:

- INGLIST provides some main attributes of a resource, for example a resource’s different statuses.
- INGINFO provides information about a resource from the point of view of the automation manager.

## Statuses Supplied by the Automation Manager

Three of these statuses, the observed status, automation status, and health status are supplied by the automation agent that is responsible for running the resource.

The desired status is derived through the request propagation mechanism. The startability status is derived through status propagation.

The final status, compound, is a summary of all of the other statuses and a number of other values. The compound status is made visible through INGLIST or INGINFO and, eventually, placed into RODM and made visible on the NMC workstation. It is also displayed on the Tivoli Enterprise Portal (TEP).

### Observed Status

This represents the current status of the resource and is provided by the related automation agent.

Possible values are:

#### **Unknown (Assumed unavailable)**

The automation manager has no observed status information about the resource. To be on the safe side, the automation manager will assume that the resource is unavailable and will try to make it available if this is the desired status. On the other hand, this assumption allows the automation manager to process shutdown requests for supporting resources that can only be made unavailable if this resource is unavailable.

#### **SysGone (unavailable)**

The system that the resource is defined on is no longer a member of the sysplex.

#### **SoftDown (unavailable)**

The resource is unavailable and automation may restart it.

#### **HardDown (unavailable)**

The resource is unavailable and automation will not restart it.

#### **Starting (available)**

The automation agent is either in the process of making the resource available or has detected actions from some outside command to make the resource available.

#### **Available**

The resource is ready for use.

#### **Degraded (available)**

The resource is available but not all required members in the server group are available.

#### **Standby (unavailable)**

The resource has a primary/secondary system association defined. The automation agent posts this status to the automation manager when setting the automation agent status to MOVED or FALLBACK.

**Note:** The automation manager treats STANDBY like a HARDDOWN status except that it is not considered that this indicates an error condition.

#### **Problem (available)**

The resource is available but has a serious problem, probably making it unusable.

## States Supplied by the Automation Manager

### Stopping (available)

Either the automation agent is in the process of stopping the resource, or it has detected that some outside command is doing so.

### WasAvailable (Assumed available)

The automation manager has lost contact with the automation agent responsible for the resource, however the resource was available when contact was lost and the system it is on still appears to be running. The resource will be treated as being available. This is primarily to prevent inappropriate recovery actions from being initiated whenever an SA z/OS NetView is recycled.

## Observed Status Transitions

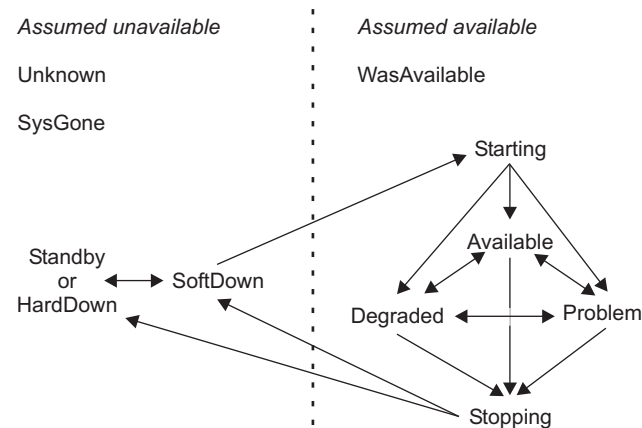


Figure 9. Observed Status Transitions

## Automation Status

This indicates the status of the automation for the resource. It is provided by the related automation agent.

Possible values are:

### Unknown

No connection with the automation agent.

### Idle

No orders have been sent; the automation agent is not doing anything. The manager may send the automation agent new orders.

### Ordered

Orders have been sent. This status is only posted within the automation manager.

### Busy

The automation agent is processing the orders, or observing/assisting a start or stop process initiated by some outside command. This status is entered when a resources status becomes Starting or Stopping and the previous Automation Status was Idle or Ordered. This status is changed to Idle if the resource achieves an observed status other than Starting or Stopping. This status may be posted directly by the automation agent.

### Denied

The automation agent was not allowed to process the last order it received. This status is changed to Idle if the resource achieves an observed status of HardDown, SoftDown or Available. This status is posted directly by the automation agent.

## Statuses Supplied by the Automation Manager

### Problem

The automation agent encountered a problem while processing for the resource. This status is changed to Idle if the resource achieves an observed status of HardDown, SoftDown or Available. This status is posted directly by the automation agent.

### Internal

This means that the automation of the resource is being handled internally.

## Automation Status Transition

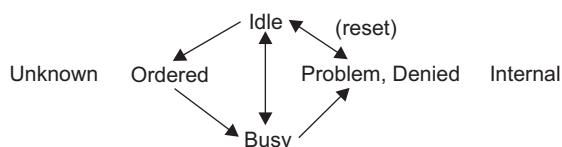


Figure 10. Automations Status Transitions

The automation manager may only send orders to an idle automation agent.

Transition from Idle to Busy occurs when the automation agent detects a third party (such as an operator or ARM) starting or stopping a resource.

## Desired Status

You can set the desired status of a resource either:

- With the INGREQ command
- With a service period
- With a schedule or resource override (INGSCHEM)
- Or indirectly through a vote that stems from the request propagation

Possible values are:

### Available

The resource should be started (made available).

### Unavailable

The resource should be stopped (made unavailable).

## Startability Status

The startability status is calculated from the resource's current status, its automation status, its automation and hold flags and the startability of its supporting resources.

Possible values are:

**Yes** The resource is either startable or already started.

**No** The resource cannot be started due to a problem with the resource itself.

### Inhibited

The resource cannot be started because of either a problem with one of its supporting resources or because automation has been prohibited.

### Denied

The resource cannot be started because the automation status of this resource is *Denied*.



### Compound Status

This status is a summary of all the statuses of a resource. It considers elements of the four other statuses as well as a number of other values.

Possible values are:

#### **Problem**

There is a problem with this resource that automation cannot solve. Operator intervention is required.

#### **Denied**

The resource is not in its desired state and automation is unable to proceed because either the automation flag has been turned off or the automation manager hold flag for the resource is set to YES.

#### **Inhibited**

The resource is not in its desired state and automation is unable to proceed because of a problem with a supporting resource. Operators should try to fix supporting resources with compound status *Problem* and *Denied*.

#### **Awaiting [Automation]**

The resource is not in its desired status and SA z/OS is waiting for its supporting resources to reach the appropriate state.

#### **InAuto[mation]**

SA z/OS is in the process of starting or stopping the resource.

#### **Degraded**

For a group it means that it is partially running, but not at full capacity. For normal resources, it can mean that the resource is Starting or Stopping, or that the application is suffering from a performance or throughput problem (corresponding to automation agent status HALTED).

#### **Satisfactory**

The resource's desired and observed statuses are corresponding; no further automation or operator activity is required.

### Health Status

With application-specific performance and health monitoring, a separate status shows up to inform you about the application's health.

Possible values are:

#### **Normal**

The observed resource is behaving normally.

#### **Warning**

The resource's health state has crossed the warning threshold.

**Minor** The resource's health state has crossed the minor threshold.

#### **Critical**

The resource's health state has crossed the critical threshold.

**Fatal** The resource's health state has crossed the fatal threshold.

#### **Unknown**

The health state of the resource is not known.

---

### Statuses Supplied by Automation Agents

There are different types of statuses that may be applied to an application. They are all retrieved on the side of the automation agent. These are:

- “Automation Agent Statuses”
- “Application Monitor Status” on page 46
- “Automatic Restart Manager Statuses” on page 47

Furthermore, this chapter discusses the following related topics:

- “Mapping the Automation Agent Statuses to the Automation Manager Observed Status” on page 48
- “SA z/OS Processes Involved in Status Determination” on page 50

### Automation Agent Statuses

SA z/OS defines different automation agent statuses that it applies to automated resources. The automation agent status of a resource is determined from a combination of its application monitor status, its desired status, recent history, and intended action.

These statuses are used on the automation agents to control the automation. They are translated to *observed statuses* (see “Mapping the Automation Agent Statuses to the Automation Manager Observed Status” on page 48) and then sent to the automation manager.

On the automation agent you can use the DISPSTAT and DISPINFO commands to see the automation statuses.

Table 4 gives a brief description of each automation agent status.

Table 4. Automation Agent Statuses

Value	Meaning
ABENDING	<p>The application is undergoing a recoverable abnormal end. This status is entered when a TERMINATION message with the ABEND=YES attribute is received. It remains in ABENDING until its final termination message is received. If the application is defined to Automatic Restart Manager it may be posted to EXTSTART when Automatic Restart Manager attempts to recover it. If the application has not exceeded its critical threshold, SA z/OS posts the application to RESTART and attempts to restart it. If it has exceeded its critical threshold it is put into BROKEN status when it has been cleaned up, and SA z/OS does not attempt to start it. If it is slow to leave the system after its FINAL message is received it may go to ZOMBIE status.</p> <p>If the application is undergoing a normal shutdown by SA z/OS no further shutdown commands are issued. The shutdown may be resumed by using the INGREQ command with a stronger type.</p>

Table 4. Automation Agent Statuses (continued)

Value	Meaning
ACTIVE	<p>The application is running, but is not yet ready for work. An application can be put into ACTIVE status in response to the following conditions:</p> <ul style="list-style-type: none"> <li>• SA z/OS has received an ACTIVE message for the application.</li> <li>• The SA z/OS startup checker has run for the application when its automation status was STARTED. The startup checker found the application monitor status for the application to be ACTIVE.</li> <li>• During routine status checking the application monitor status for the application was found to be ACTIVE when its automation status indicated it should be otherwise.</li> <li>• SA z/OS attempted to start the application, but found that its application monitor status was ACTIVE.</li> <li>• SA z/OS found that the Automatic Restart Manager status for the application is STARTING, but has not received any messages concerning the application status.</li> <li>• SA z/OS checked an attempt by Automatic Restart Manager to restart the application and found that its Automatic Restart Manager status is UNREGISTERED, but the application monitor status for the application is ACTIVE.</li> </ul> <p>An application remains in ACTIVE status until its UP message is received. If the application is starting or restarting, an SA z/OS monitoring check for the application is done after its start delay time. If the application is not found it is put into BROKEN status. If it is found, but is not yet UP, it is put into STARTED2 status.</p>
AUTODOWN	<p>The application is shut down. SA z/OS may restart it to comply with an operator request. If the shutdown specified that the application was to be restarted, it is put into RESTART status when the shutdown is complete.</p> <ol style="list-style-type: none"> <li>1. SA z/OS has shut the application down at the request of an operator.</li> <li>2. When SA z/OS initialized, the operator replied NOSTART to the AOF603D WTOR. Any applications that would have been put into the DOWN status during initial status determination have instead been put into the AUTODOWN automation status.</li> </ol> <p>You can use the SETSTATE command to change the application status to either RESTART or CTLDOWN.</p> <p>SA z/OS may attempt to restart an application in AUTODOWN status when SA z/OS is reloaded, or when an operator requests SA z/OS to start one of the application descendents. In both cases, the application goes to RESTART status.</p>
AUTOTERM	<p>SA z/OS is in the process of shutting the application down. The shutdown is in response to a INGREQ REQ=STOP command. This status persists until SA z/OS is sure that the application has been cleaned up.</p> <p>Many things may happen to an application that is being shut down. If the shutdown is successful, the application is placed in either AUTODOWN or CTLDOWN status. If the shutdown specified that the application should be restarted it goes through AUTODOWN to RESTART status.</p> <p>If the application abnormally ends while it is being shut down it may go into either ABENDING or BREAKING. A normal shutdown will stop processing an application that abends, but other shutdowns will continue. If the shutdown runs out of commands to issue, the application is placed into STUCK status. If it has problems shutting down, the application may be placed into ZOMBIE status.</p>
BREAKING	<p>The application is undergoing a nonrecoverable abend; that is, it has received a termination message specifying BREAK=YES. If the application is undergoing a normal shutdown by SA z/OS no further shutdown commands are issued. The shutdown may be resumed by using the INGREQ REQ=STOP command. This status persists until SA z/OS receives its final termination message and is sure that the application has been cleaned up. If the termination experiences difficulties, the application may be posted to ZOMBIE status.</p>

## Statuses Supplied by Automation Agents

Table 4. Automation Agent Statuses (continued)

Value	Meaning
BROKEN	<p>The application has suffered a nonrecoverable abend. SA z/OS will not restart it. An application can be put into BROKEN status in response to the following conditions:</p> <ol style="list-style-type: none"> <li>1. The application has suffered a nonrecoverable abend, indicated by the reception of a TERMINATION message with the BREAK=YES attribute.</li> <li>2. The application has suffered sufficient recoverable abends to exceed its critical threshold.</li> <li>3. The application has suffered a prestart or startup failure.</li> </ol> <p>This status is preserved across a recycle of SA z/OS or a re-IPL of the processor, unless the application has its Restart after IPL Option set to START or NOSTART.</p>
CTLDDOWN	<p>The application is shut down and SA z/OS is not allowed to restart it.</p> <ol style="list-style-type: none"> <li>1. An operator asked SA z/OS to shut the application down and not to restart it until authorized to do so by an operator.</li> <li>2. An operator used a SETSTATE command to tell SA z/OS that an application should not be restarted until an operator authorizes SA z/OS to do so.</li> <li>3. SA z/OS was started for the first time, the application monitor status is INACTIVE and its Restart after IPL Option is NOSTART.</li> </ol> <p>You can use the SETSTATE command to change an application status from CTLDDOWN to RESTART or AUTODOWN, in which case SA z/OS will attempt to restart it.</p>
DOWN	<p>The application has not been started during the lifetime of this SA z/OS.</p> <p>The DOWN status is set only during initial status determination and is possible only if the application monitor status is INACTIVE. The automation status of the application when SA z/OS was last shut down on this system is used in the following manner to determine if it is to be placed into the DOWN status.</p> <ol style="list-style-type: none"> <li>1. The previous automation status was not one of STOPPED, CTLDDOWN or BROKEN.</li> <li>2. The application has Start On IPL YES or NO specified, the previous state was BROKEN, CTLDDOWN or STOPPED and this is the first time the agent has been started since the system was IPLed.</li> </ol>
ENDED	<p>This status is used for transient applications only, and indicates that the job for the application has finished and left the system without any errors. Any start-dependent resources for the application will be started as though it were a normal z/OS subsystem that was UP.</p> <p>If the transient application can be rerun, you can use the SETSTATE command to restart it. If the transient application cannot be rerun, it will remain in ENDED status.</p>
ENDING	<p>A transient application is in the process of terminating. A transient application goes to ENDING status when a termination message is received for it, and it is not being shut down by SA z/OS. This status shows that the application is terminating, but that this is expected. A transient application may also go to ENDING if an operator is shutting it down outside SA z/OS control, or if it has abnormally ended, but the abend messages are being treated as normal termination messages.</p> <p>The application remains in ENDING status until either:</p> <ul style="list-style-type: none"> <li>• An abend message is received that will put it into either ABENDING or BREAKING status. If the application abends then either SA z/OS or Automatic Restart Manager can restart it.</li> <li>• The application final termination message is received, at which point the RESTARTOPT for the application is checked. If it is ALWAYS then the application is put into RESTART status and SA z/OS will attempt to restart it. If it is anything else, the application goes to ENDED status. It is assumed that if a transient application ends normally then it will deregister from Automatic Restart Manager. If it is slow to clear the system after its FINAL message is received it may go to ZOMBIE status.</li> </ul> <p>If an ACTIVE or UP message is received for the application, its automation status is changed to either ACTIVE or UP, as appropriate.</p>

Table 4. Automation Agent Statuses (continued)

Value	Meaning
EXTSTART	<p>SA z/OS has determined that the application is being started or restarted by an agent external to SA z/OS. In situations where SA z/OS is able to identify the external agent (such as Automatic Restart Manager), it takes appropriate steps to monitor that agent's actions and, if necessary, step in to assist it. An application can be put into EXTSTART status in response to the following conditions:</p> <ul style="list-style-type: none"> <li>• SA z/OS is unable to identify the external agent.</li> <li>• Automatic Restart Manager is in the process of restarting the application.</li> </ul>
FALLBACK	<p>The application is not running on the primary system where it should run, but this status has been encountered for this application on one of its secondary systems. It should be active on another system. If the other system fails, the application can fall back to this system, where it could possibly be restarted. However, SA z/OS, will not perform the restart on the fallback system, but this may be done by an operator request. This is implemented to leave the decision of restarting the application on the fallback system to the operator.</p> <p>An application can be put into FALLBACK status in response to the following conditions:</p> <ul style="list-style-type: none"> <li>• It is defined with a secondary association on this system.</li> <li>• An operator has used the SETSTATE command to put the application into the MOVED status. If this is one of the secondary systems for the application it will go to FALLBACK instead.</li> </ul>
HALFDOWN	<p>SA z/OS was in the process of shutting the application down, but the stop request was canceled while it was in progress. The application shutdown did not complete (for example, ASCBs may still be active). You may sometimes find that some, but not all, of the shutdown commands have been issued. To recover an application from HALFDOWN status you must determine where it is in its shutdown process and complete the shutdown manually. Applications go into HALFDOWN only when you cancel a stop request. Alternatively, you can use SETSTATE to put the application back into UP or RUNNING status.</p>
HALTED	<p>The application is still running, but something has happened that may have severely impacted its capabilities.</p> <ol style="list-style-type: none"> <li>1. SA z/OS has received a HALT message for the application.</li> <li>2. SA z/OS has detected that the application represents a JES2 application that is running short of spool space.</li> <li>3. The Automatic Restart Manager status for the application is ELSEWHERE, but SA z/OS found its application monitor status to be either STARTING or ACTIVE.</li> </ol> <p>An application is taken out of HALTED status if its UP message is received. Also, operators may use the SETSTATE command to put the application into UP or RUNNING status.</p>
MOVED	<p>The application is not running: this is one of its primary systems: it should be active on another system. An application can be put into MOVED status in response to the following conditions:</p> <ul style="list-style-type: none"> <li>• An operator has used the SETSTATE command to put the application into the MOVED status. This is possible only on a primary system.</li> <li>• The startup detected that Automatic Restart Manager has started the subsystem on another system.</li> </ul> <p>A subsystem will remain in the MOVED status until it is restarted on the primary system by an external agent, such as an operator.</p>

## Statuses Supplied by Automation Agents

Table 4. Automation Agent Statuses (continued)

Value	Meaning
RESTART	<p>The application is ready to be started. It has been previously active in the system. An application can be put into RESTART status in response to the following conditions:</p> <ul style="list-style-type: none"> <li>• The application abended and, after checking thresholds, SA z/OS is allowed to restart it.</li> <li>• SA z/OS has shut the application down in response to an operator request and is now preparing to restart it.</li> <li>• An operator has used the <code>INGREQ REQ=START</code> command to ask SA z/OS to restart the application.</li> <li>• SA z/OS checked an attempt by Automatic Restart Manager to restart the application and found that its Automatic Restart Manager status is UNREGISTERED and the application monitor status for the application is INACTIVE. This implies that the attempt by Automatic Restart Manager to restart the application timed out while the application was in RESTARTING status. SA z/OS changes the automation status of the application to RESTART and attempts to start the application itself.</li> </ul> <p>During restart processing, the application RESTART automation flag is checked. If it is turned on, the application start commands are issued and the application is put into STARTED status. If the RESTART automation flag is off, the application remains in RESTART status and the startup monitor cycle initiates the startup process each time it runs.</p>
RUNNING	<p>This status is equivalent to UP, but is used for <i>transient</i> applications. It indicates that the UP message has been received for the transient application, or an operator has used the SETSTATE command to change the status of a transient application to RUNNING. A transient application is one that SA z/OS expects to terminate on its own. When the job finishes the application goes through ENDING status to ENDED, at which point its descendants are started. Unlike the UP status, the descendants of a transient application are not started until it has terminated.</p> <p>A transient application should leave RUNNING status on its own. If it gets stuck, you should investigate it. You can use the <code>INGREQ REQ=STOP</code> command to put it into AUTODOWN status.</p>
STARTED	<p>The commands to start the application have been issued, but it has yet to start running. An application can be put into STARTED status in response to the following conditions:</p> <ul style="list-style-type: none"> <li>• SA z/OS has issued, or will soon issue, the commands.</li> <li>• When SA z/OS attempted to start the application it found that the application monitor status for the application was STARTING.</li> <li>• During initial status determination, SA z/OS found that the application monitor status for the application was STARTING.</li> <li>• SA z/OS checked an attempt by Automatic Restart Manager to restart the application and found that its Automatic Restart Manager status is UNREGISTERED, but the application monitor status for the application is STARTING.</li> </ul> <p>Note that the relevant automation flag, <code>Initstart</code> or <code>Restart</code>, must be on. The application startup commands as defined in the automation control file are issued after the application is placed in STARTING status.</p> <p>An application remains in STARTING status until either its ACTIVE or its UP message arrives. After the application start delay time, an SA z/OS monitoring check is issued for it. If it is not found, it is put into BROKEN status. If it is found, but is not yet UP, it is put into ACTIVE status.</p>

Table 4. Automation Agent Statuses (continued)

Value	Meaning
STARTED2	<p>The application has become active, but has not indicated that it is ready to accept work within its <i>start delay</i>. An application can be put into STARTED2 status in response to the following conditions:</p> <ul style="list-style-type: none"> <li>• The startup checker found that the application monitor status was still STARTING.</li> <li>• The startup checker was called for an application whose automation status was ACTIVE.</li> <li>• SA z/OS found the Automatic Restart Manager status for the application to be AVAILABLE-TO.</li> <li>• SA z/OS checked an attempt by Automatic Restart Manager to restart the application and found that its Automatic Restart Manager status is one of AVAILABLE-TO, RESTARTING or RECOVERING.</li> <li>• SA z/OS checked an attempt by Automatic Restart Manager to restart the application and could not find a better status to put it into. Its application monitor status is ACTIVE. This covers situations where the restart checker is unable to determine the Automatic Restart Manager status for the application, or its status is FAILED.</li> </ul> <p>An application remains in STARTED2 status until either its UP message arrives or an operator uses the SETSTATE command to change the application status to UP or RUNNING.</p>
STOPPED	<p>The application has been shut down by an external agent, such as an operator cancel. SA z/OS is not permitted to restart it and will not allow the Automatic Restart Manager to restart it. This status is preserved across a recycle of SA z/OS or a re-IPL of the processor unless the application has its Restart after IPL Option set to START or NOSTART.</p> <p>An application remains in STOPPED status until an operator uses the SETSTATE command to change its status to either RESTART or CTLDOWN.</p> <p>An application may also leave STOPPED status if it is restarted outside the control of SA z/OS. In this case, it goes to either the ACTIVE or UP status, depending on which message is received.</p>
STOPPING	<p>The application is being shut down by an external agent. This status is entered when a TERMINATION message is received while SA z/OS is not in the process of shutting the application down. It may also indicate that the application is abending, but the abend messages issued are being treated as normal termination messages by automation.</p> <p>An application that is STOPPING remains in that status until either an abend message is received that puts it into ABENDING or BREAKING status, or the final termination message for the application is received, at which point the application RESTART option is checked. If RESTART is ALWAYS, the application is put into RESTART status and SA z/OS attempts to restart it. Otherwise, the application goes to STOPPED status. If it is slow to clear the system after the final message is received, it may be placed into ZOMBIE status.</p> <p>If an ACTIVE or UP message is received for the application, its automation status is changed to either ACTIVE or UP, as appropriate.</p> <p>Automatic Restart Manager interaction with this status depends on a number of factors. If the application is deregistered then there is no interaction. If the application is registered then Automatic Restart Manager will attempt to restart it. If the application goes to STOPPED status before the SA z/OS or Automatic Restart Manager exit is invoked then SA z/OS will tell Automatic Restart Manager not to restart the application. If it does not, SA z/OS will tell Automatic Restart Manager that it can restart the application.</p>
STUCK	<p>An application can get STUCK when it is being shut down. This is because SA z/OS has run out of NORM, IMMED, or FORCE shutdown commands.</p>

## Statuses Supplied by Automation Agents

Table 4. Automation Agent Statuses (continued)

Value	Meaning
UP	<p>The application has finished initializing and is ready for work. An application can be put into UP status in response to the following conditions:</p> <ul style="list-style-type: none"> <li>SA z/OS has received an UP message for the application.</li> <li>An operator has used the SETSTATE command to change the application automation status. In this case, SA z/OS assumes that the operator has ensured that the application is actually UP.</li> <li>During initial status determination the application monitor status was found to be ACTIVE.</li> <li>SA z/OS found the Automatic Restart Manager status for the application to be AVAILABLE.</li> </ul> <p>There are a number of ways for an application to leave the UP status, if:</p> <ul style="list-style-type: none"> <li>It is shut down with the INGREQ REQ=STOP command, it goes to AUTOTERM status</li> <li>It is shut down outside SA z/OS, it goes to STOPPING status</li> <li>It abends, it might go to STOPPING, ABENDING or BREAKING status</li> <li>It has problems, it may go to HALTED status</li> <li>The regular monitor cannot find it, it will call the TERMMSG generic routine</li> <li>The application abends, SA z/OS does not pick up the abend messages, and Automatic Restart Manager detects that the address space has ended, the application may go to EXTSTART.</li> </ul>
ZOMBIE	<p>When an application is leaving the system it can enter a ZOMBIE status. This indicates that the final termination message for the application has been received but that SA z/OS monitoring in TERMMSG still finds the application. The application is put into ZOMBIE status if this situation persists for more than twice the application termination delay time.</p> <p>There are three ways that an application can enter a ZOMBIE status:</p> <ul style="list-style-type: none"> <li>If MVS is slow in clearing the application and the termination delay time is short.</li> <li>If there are two jobs with the same name in the system, one of which is the application. When either of them terminates and SA z/OS does not know the address space id of the application or does not get the address space id with the termination message, SA z/OS assumes that the application has stopped, but SA z/OS monitoring will find the other job. To change the status to UP, either manually shut down the other job, or use the SETSTATE command to change the application status back to UP.</li> <li>The job may have become stuck in the system after issuing its final message.</li> </ul> <p>From ZOMBIE status, if the application suffers an unrecoverable abend it will go into BREAKING status.</p> <p><b>Note:</b> The Restart after IPL option of the customization dialog may override these resource statuses at SA z/OS IPL or recycle, resulting in SA z/OS starting the subsystem.</p>

## Application Monitor Status

SA z/OS determines the application monitor status of an application by running a routine that you specified in the customization dialog. You can specify the routine for an individual application, and you can specify a default application monitor routine for applications on an entire system. For further information on application monitor routines, refer to *IBM Tivoli System Automation for z/OS Defining Automation Policy*.

Table 5. Regular Monitoring: Automation Status Determination

Application Monitor Status	Automation Status Change
ACTIVE	The application automation status is changed to ACTIVE.
STARTING	The application automation status is changed to STARTED and a startup checker is scheduled for it.



Table 5. Regular Monitoring: Automation Status Determination (continued)

Application Monitor Status	Automation Status Change
INACTIVE	If the application is defined with a RESTART option of ALWAYS, the automation status is changed to RESTART and the restart process is invoked. If its RESTART Option is not ALWAYS, its automation status is changed to INACTIVE.

### Automatic Restart Manager Statuses

The z/OS Automatic Restart Manager statuses are determined by SA z/OS and are used to help determine automation statuses. These statuses are included here as background information so that you can better understand the SA z/OS automation statuses. They are a superset of the statuses that Automatic Restart Manager uses, the added statuses being ELSEWHERE and UNREGISTERED. Possible values for the Automatic Restart Manager status are:

Value	Meaning
AVAILABLE	The element name is registered with Automatic Restart Manager, the job is running on this system and the Automatic Restart Manager status for it is AVAILABLE.
AVAILABLE-TO	The element name is registered with Automatic Restart Manager and the job is running on this system. The job has been restarted by Automatic Restart Manager, has issued its IXCARM REGISTER call, but did not issue its IXCARM READY call within the <b>ready timeout</b> defined to z/OS for it.
STARTING	The element name is registered to Automatic Restart Manager and the job is running on this system. The job was started by an agent external to MVS, has issued its IXCARM REGISTER call, but has not yet issued its IXCARM READY call. Unlike the RECOVERING status, this status does not time out to the AVAILABLE-TO status.
RECOVERING	The element name is registered to Automatic Restart Manager and the job is running on this system. Automatic Restart Manager has issued the restart command for the application, it has issued its IXCARM REGISTER call, but has not yet issued its IXCARM READY call. If it takes too long to do this, it is placed in the AVAILABLE-TO status.
FAILED	The element name is registered with Automatic Restart Manager, the job has failed, but was last active on this system. It is unknown at this point if Automatic Restart Manager will attempt to restart the job. SA z/OS will continue to report an elements status as FAILED until MVS either deregisters it, or is committed to restarting it. MVS query commands issued during this time will probably return a RESTARTING status, however MVS Automatic Restart Manager is still consulting its policies and exits to determine whether it should restart the element.

## Statuses Supplied by Automation Agents

Value	Meaning
RESTARTING	The element name is registered with Automatic Restart Manager, the job has failed and Automatic Restart Manager is attempting to restart it. This status is entered very early in the restart process, before Automatic Restart Manager is committed to restarting the element. SA z/OS continues to report the status as FAILED until Automatic Restart Manager <i>is</i> committed to issuing the restart commands. This means that MVS query commands may report a status of RESTARTING while SA z/OS reports a status of FAILED. The RESTARTING status is exited when the job issues its IXCARM REGISTER call. If it takes too long to do this, its element name is deregistered.
ELSEWHERE	The element name is registered with Automatic Restart Manager, but is currently active on a different system. Unless that system fails, Automatic Restart Manager will not move the application to this system.
UNREGISTERED	The element name is not, currently, registered with Automatic Restart Manager. This may mean that the application has never been active within the sysplex, or it may mean that it failed and Automatic Restart Manager could not recover it. If the SA z/OS routine to determine an Automatic Restart Manager status finds that Automatic Restart Manager recovery is inactive, the UNREGISTERED Automatic Restart Manager status is returned for all enquiries. This means that SA z/OS effectively ignores Automatic Restart Manager.

## Mapping the Automation Agent Statuses to the Automation Manager Observed Status

Table 6 shows how the current resource statuses that are retrieved by the related automation agents are mapped to the observed status that is used by the automation manager.

*Table 6. Automation Agent to Automation Manager Status Mapping*

Automation Agent	Observed	Automation	Notes
DOWN	SoftDown	Idle	The automation agent is ready to perform an initial start of the resource.
RESTART	SoftDown	Idle	The automation agent is ready to perform a startup of the resource.
STARTED	Starting	Busy	The automation agent has issued the start commands for the resource.
EXTSTART	Starting	Busy	This means that the automation agent has detected that an external agency (probably ARM) is in the process of starting the resource.
ACTIVE	Starting	Busy	The automation agent has received some indication that the start is successful, although it is not yet complete.
STARTED2	Problem	Problem	Although the resource is still active, it does not appear to have started properly within the time period that the automation agent expected it to start within.
UP	Available	Idle	The resource has entered a stable state and is available for usage.

## States Supplied by Automation Agents

Table 6. Automation Agent to Automation Manager Status Mapping (continued)

Automation Agent	Observed	Automation	Notes
HALTED	Degraded	Busy	The resource has encountered a problem, but is still active. The automation agent may be attempting to return it to an UP state.
RUNNING	Starting	Busy	The resource is a transient subsystem that is currently executing.
ENDING	Starting	Busy	The resource is a transient subsystems that is terminating.
AUTOTERM	Stopping	Busy	The automation agent is in the process of stopping the resource.
STOPPING	Stopping	Busy	The automation agent has detected that the application is terminating normally outside of its control.
ABENDING	Stopping	Busy	The automation agent has determined that the application is terminating abnormally but can be recovered.
BREAKING	Stopping	Busy	The automation agent has determined that the resource is terminating abnormally in a nonrecoverable fashion.
ENDED	Available	Idle	The resource is a transient subsystem that has completed successfully.
AUTODOWN	SoftDown	Idle	The resource is down and may be restarted as needed by automation.
CTLDOWN	HardDown	Idle	The resource is down and may not be restarted by automation.
STOPPED	HardDown	Idle	The automation agent has determined that the resource has been stopped normally outside the control of automation and its policy indicates that an automated restart is not allowed.
BROKEN	HardDown	Idle	The automation agent has determined that the application has failed in a nonrecoverable fashion.
HALFDOWN	Degraded	Problem	A shutdown for the resource by the automation agent was cancelled at the users request. Although the resource is still active, its status is indeterminate.
STUCK	Stopping	Problem	While trying to stop or start the resource, the automation agent ran out of instructions.
ZOMBIE	Problem	Problem	The automation agent believes that the resource should be inactive, but it is not.
MOVED	Standby	Idle	The automation agent has determined that the resource is active elsewhere within the sysplex. The manager may reactivate it on this system.

## Statuses Supplied by Automation Agents

Table 6. Automation Agent to Automation Manager Status Mapping (continued)

Automation Agent	Observed	Automation	Notes
FALLBACK	Standby	Idle	The resource is a backup for a resource that normally runs on a different system within the sysplex. The manager may start it if it needs to.

## SA z/OS Processes Involved in Status Determination

This section describes a number of SA z/OS processes that are involved in determining the various statuses of resources. A resource's status is determined by the automation agent and reported to the automation manager.

### Initial Status Determination

This process occurs at the end of SA z/OS initialization and after a configuration refresh. It attempts to determine a suitable automation status for each application that is defined to SA z/OS.

### Subsystem Startup

This process is invoked whenever SA z/OS is about to start an application. It checks that conditions are right for the application to be started and that SA z/OS is allowed to start the application. If appropriate, the subsystem startup process then issues the commands to start the application and changes the application automation status to STARTED.

### Startup Checker

This process is run for each application after SA z/OS has issued the startup commands, after the amount of time specified by the Start Timeout defined for the application. It changes the automation status from STARTED to STARTED2 if the application fails to start within its expected start delay.

### Regular Monitoring

This process runs at an interval defined by the automation programmer in the customization dialog. Normally the time interval is between 30 minutes and 12 hours. The purpose of regular monitoring is to carry out a reality check to ensure that the current SA z/OS automation status for an application is consistent with the current application status as determined by the application monitor routine.

To help determine consistency, SA z/OS statuses are divided into four groups:

#### UP STATUSES

These statuses mean that the application monitor status should be ACTIVE. They include:

- UP
- RUNNING
- ACTIVE
- HALTED

#### DOWN STATUSES

These statuses indicate that the application monitor status should be INACTIVE. They include:

- DOWN
- RESTART
- INACTIVE

## Statuses Supplied by Automation Agents

- AUTODOWN
- CTLDOWN
- BROKEN
- STOPPED
- ENDED
- FALLBACK
- MOVED

### STARTING STATUSES

These statuses indicate that the application monitor status may be STARTING or ACTIVE. They include:

- STARTED
- STARTED2
- EXTSTART

### TRANSITIONAL STATUSES

These statuses are either part of the shutdown process, or indicate that the application may or may not be present in the machine. *All* application monitor statuses are possible with any transitional status. They include:

- AUTOTERM
- STUCK
- ZOMBIE
- BREAKING
- ABENDING
- HALFDOWN
- STOPPING
- ENDING

If regular monitoring finds a status mismatch, it changes the application automation status according to Table 5 on page 46.

It should be noted that several other routines have similar status mismatch detection capabilities built in.

### Secondary Monitor

This is invoked, after a small delay, from the application monitor routine. This process performs a recheck for a specific application that the application monitor routine found to be in an unusual (generally a transient) state. It is responsible for making some of the transitions outlined under the Routine Monitor process.

### Shutdown

This process is normally invoked by an operator to shut down one or more applications. The shutdown process is responsible for the sequencing of the various shutdown commands. Depending on the parameters it is invoked with, the application that is shut down may be put into an AUTODOWN or CTLDOWN automation status, or it may be restarted.

### Resynchronization Process (Environmental Analysis Process)

It may be invoked manually by an operator using the RESYNC command. This process assesses the current status of the various parts of the z/OS system that SA z/OS automates. It issues display commands to obtain information on WTORs and sysplex resources. It also invokes the Initial Status Determination process. When it runs, all previous status information is discarded.

### Using SA z/OS for Monitoring

As an operator, you can monitor and display information about the resources using two types of interfaces:

- The NMC workstation provides monitoring capability for system operations and processor operations.
- The SA z/OS 3270-type interface provides monitoring capability for:
  - Processor operations
  - System operations via Status Display Facility (SDF), displaying status information (INGLIST) and detailed resource information (INGINFO and DISPINFO for the automation agent view)
  - The Tivoli Enterprise Portal (TEP) of the IBM Tivoli Monitoring infrastructure. It displays information about the status of automation on z/OS systems and z/OS sysplexes that has been collected by the SA z/OS monitoring agent.

You can also monitor messages and alerts. More information about this topic is given in “How to Monitor Alerts on a NetView Console” on page 114.

If SA z/OS has been configured properly, status mismatches between the actual and the monitored status should not occur. In the customization dialog, the automation programmer can specify a monitoring interval for every resource individually. If such an individual interval is not defined for a resource, every half hour (which is the default, but the interval is user specifiable) SA z/OS will run a monitor routine to check the resources' status as known to SA z/OS versus its actual status. If the result indicates an error, its automation status will be updated.

In the customization dialog, the automation programmer can define either an individual monitor routine per application or a default monitor routine for a system that will monitor all applications running on that system. You can manually invoke an application's related monitor routine through the MONITOR command. You can use a wildcard for the applications (subsystems) that you want to be monitored, for example, use \*abc\* for all subsystems containing the string 'abc' in their names.

### Resynchronization

If you believe that a lot of the status information provided by SA z/OS is badly out of sync with reality, you can request that SA z/OS reruns its complete subsystem resynchronization process.

This is achieved with the RESYNC command, but it should be used with care. The resynchronization can take a few minutes and during this time a lot of SA z/OS's commands are not available.

---

## Chapter 3. Starting and Stopping SA z/OS

This chapter contains the following sections:

- “Starting, Stopping, and Maintaining the Automation Manager”
- “Starting and Stopping I/O Operations” on page 57
- “Starting NetView (System Operations)” on page 57
- “Starting and Stopping Processor Operations” on page 58

---

### Starting, Stopping, and Maintaining the Automation Manager

Normally, user interaction with the automation manager is limited to using INGXXXX commands to, for example, send the automation manager requests to start or stop resources. In special cases, however, there may be a need to interact with the automation manager during the initial start process and when it is necessary to stop an automation manager instance.

#### Starting the Automation Manager: START (S)

The automation manager runs in a z/OS address space of its own. Normally, you would add this start command to the COMMNDxx PARMLIB member so that the automation manager is automatically started at IPL time as described in *IBM Tivoli System Automation for z/OS Planning and Installation*. Alternatively, you can start it with an MVS start command that calls a module that initializes the automation manager.

In certain cases, you can also use the START (S) command.

Because the automation manager runs as a started task, the start command format may be modified by the installation. However a sample procedure is provided and can be used in most installations without modification. In addition to the various data sets identified by this procedure, there are several parameter values that may be specified. None of these parameters are required due to default values or additional sources of the information that are provided. You can find a sample startup procedure called INGEAMSA in the SINGSAMP sample library, see also *IBM Tivoli System Automation for z/OS Planning and Installation*.

In most cases, the parameters specified via the start command, if provided, override information obtained from the automation manager’s PARMLIB member or other default value. The only exception to this is that START=HOT will be ignored when another automation manager instance has claimed primary mode.

**Note:**

Do not start two instances of the automation manager with the same job name on the same system. They would come up as the primary and secondary automation manager. However, if you issue a stop command with the same job name, it would be accepted by both instances.

Also you cannot start more than nine instances of an automation manager per z/OS system within the sysplex or per single system.

To allow your installation to specify a preference as to which automation manager instance should assume PAM responsibility at sysplex IPL, each automation manager enters a DELAY state. The duration of this state is set with the DELAY parameter of the automation manager initialization member of the PARMLIB (see also the description of the HASPRM00 sample member in *IBM Tivoli System Automation for z/OS Planning and Installation*). This allows other instances with a lower delay value to assume PAM mode if started within the difference (in seconds) between the respective delay values. This makes sense when you would prefer that the primary automation manager runs on a special powerful processor.

### Stopping the Automation Manager: STOP (P)

To allow an automation manager to be stopped or otherwise influenced by an operator, the automation manager "listens" for MVS MODIFY and STOP commands (see also "Making an Automation Manager the Primary.") When an MVS STOP (P *jobname*) command is received, the automation manager will be terminated. If the automation manager was the primary one, all work in process will be completed prior to actual termination.

If you try to stop the last primary automation manager, you will receive the reply HSAM1390E that you must confirm the automation manager stop request to.

### Making an Automation Manager the Primary

You can use the GO subcommand of the MVS MODIFY (F) command to cancel the initial DELAY invoked when an automation manager instance is started thereby making it eligible to become the primary instance without having to wait for the delay time to complete. This is useful when it is known that a preferred primary system will not become available during a sysplex wide IPL:

```
F jobname,GO
```

### Stopping the Automation Manager with Different Options

The STOP subcommand of the MVS MODIFY (F) command may be used to stop an automation manager instance in different ways:

```
F jobname ,STOP,[ NORMAL | IMMED | FORCE | DEFER ],[ NORESTART | RESTART ]
```

The available options are:

**NORMAL**

The instance will be stopped after all in progress work has been completed



## Starting, Stopping and Maintaining the Automation Manager

and committed. No new work will be accepted from the automation agents, however any internally queued work will be processed. This is the default option.

### IMMED

The instance will be stopped after the current work items have been completed and committed. Internally queued work may be lost.

### FORCE

The instance will be stopped without waiting for in progress work to complete.

### DEFER

The instance will be conditionally and normally stopped after the STOPDELAY interval has expired. Termination will occur only if another (secondary) instance is available, or if no automation agent is active.

On a secondary automation manager instance, the above variations are effectively treated as NORMAL as there is no work to be completed by SA z/OS. The second option specifies whether ARM restart processing should be attempted following the completion of the stop request.

### NORESTART

The instance will be stopped as requested and no restart attempt will be made by ARM. This is the default option.

### RESTART

The instance will be stopped in such a way as to cause ARM to attempt an automatic restart. Note that this will only occur if the instance is properly defined to ARM.

## Obtaining Automation Manager Diagnostic Information

There are two methods of obtaining diagnostic information from the automation manager activities if problems occur:

- You can get a snapshot of the state image queue written into a specified data set using the command:

```
F jobname, DIAGINFO,SNAP,data_set_name
```

- You can get a log of the automation manager's activities for problem reporting (maybe the ones to reproduce the problem). You issue a command to start the log that is written to a specified data set:

```
F jobname , DIAGINFO,RECORD,data_set_name
```

With a second command, you terminate the log:

```
F jobname , DIAGINFO,ENDREC
```

**Note:** The DIAGINFO command accepts every data set name without checking whether the data set exists or is being accessed by some other user. If there are conflicts, the automation manager does not write any diagnostic information, but will not issue a message to the user.

## Takeover File Repair

The takeover file is a shared VSAM file. The takeover file is allocated to the primary automation manager (PAM) and all secondary managers (SAM) via a DD statement in the startup procedure with a disposition of SHR.

## Starting, Stopping and Maintaining the Automation Manager

If a VSAM open error occurs for example due to locking or any other I/O error while performing a hot takeover, the automation manager continues with a warm start rather than a hot start without giving the operator the ability to correct the error situation.

If an error occurs while writing to the takeover file, the automation manager continues to run using the in-storage version of the resources objects but disables updating of the takeover file. The in-storage state information of the resources is not written to the takeover file when the automation manager terminates. The user has lost the ability to continue with a hot start when switching the primary automation manager.

An explicit request to enable the takeover file will retry the "old broken" VSAM file only.

If an error occurs while reading the takeover file, the automation manager disables the takeover file and performs a warm start.

The automation manager allows you to repair the takeover file while the PAM and the SAMs are running. Furthermore you are prompted before a hot start is degraded to a warm start due to a takeover file open/read error.

### Dynamic Takeover File

The takeover file is dynamically allocated during the initialization of the primary automation manager (PAM). The old DD-statement HSATKOVVR in the startup procedure is no longer used and must be removed.

An automation manager started as SAM does not allocate the takeover file.

During initialization of the automation manager, the name of the takeover file is read from HSAPRMxx and the takeover file is allocated using the same DD statement as used before (HSATKOVVR). The name of the takeover file cannot be changed when one automation manager (PAM or SAM) is running. All automation managers (in the same XCF group) must use the same takeover file name.

In case of a hot start or takeover, and the takeover file allocation or a VSAM open/read error occurs, the operator will be prompted to make a decision:

1. Retry reading the takeover file
2. Continue with a warm start

If the operator continues with a warm start and this fails, the automation manager terminates and causes a takeover.

If the operator continues with a hot start and the previous allocation or a VSAM open/read error reoccurs the automation manager prompts the user again.

In case of cold or warm start the takeover file will be allocated during automation manager initialization. If it cannot be allocated the automation manager terminates. No WTOR is issued because this is not necessary for cold or warm start.

### Notes:

1. SA z/OS must be running on all automation managers in the same XCF group before the takeover file can be removed from the primary automation manager in-flight for repair. This is because the downlevel automation manager(s), including SAMs, have the takeover file allocated statically in shared mode.
2. If the Automatic Restart Manager (ARM) is active, it triggers the automation manager restart. This then uses the same start parameters as the original start and thus the automation manager performs a *hot* start again.

---

## Starting NetView (System Operations)

NetView must be started on the host before you can use the system operations and processor operations of SA z/OS. NetView can be automatically initialized when you start z/OS. For general information on initializing NetView, see *Tivoli NetView for z/OS Installation: Configuring Additional Components*. You may have a dual NetView environment: a Networking NetView (which runs NetView Graphic Monitor Facility and other networking facilities) and an Automation NetView (which runs SA z/OS). This topic assumes that both instances of NetView are initialized at the focal point system and that you are logged on to a NetView user ID at the SA z/OS workstation.

The system programmer may have set up your system so that SA z/OS is started automatically when the Automation NetView initializes. (Directions for performing this setup are given in *IBM Tivoli System Automation for z/OS Planning and Installation*). As SA z/OS initializes, a multiline write-to-operator (MLWTO) message, AOF767I, is issued describing the options available and system default actions. See Appendix A, "Automation Initialization Options," on page 219 for the options that can be defined and for details about the MLWTO. To continue automation, enter

R *number*

Where *number* is the message reply number of the accompanying AOF603D message. This is the default. If you do not issue a reply within two minutes, automation continues and SA z/OS loads automation policy from an automation control file.

**Note:** Use the AOFINITREPLY variable to suppress the issuing of the AOF603D message and have SA z/OS initialize with the default settings (see also *IBM Tivoli System Automation for z/OS Customizing and Programming*).

---

## Starting and Stopping I/O Operations

It is likely that the system programmer will have customized SA z/OS to initialize all of its components (system operations, processor operations and I/O operations) and related applications automatically. If this is not the case, you need to start the SA z/OS I/O operations manually.

1. I/O operations is started as an application running on the Automation NetView at the focal point system and at the target systems. To start I/O operations, issue the following command from the z/OS operator console, specifying the I/O operations procedure name that has been created as a member in your

## Starting and Stopping I/O Operations

SYS1.PROCLIB or USER.PROCLIB (the location of the I/O operations procedure depends on your installation). In the following example, INGEIO is the procedure name. Optionally, you can append an MVS identifier, which is IO in the example.

```
/S INGEIO.IO,SUB=MSTR[,PARM='TIMEOUT=n']
```

You can omit SUB=MSTR, but this is not recommended. If SUB=MSTR is omitted and the JES subsystem is not running, I/O operations will neither start nor run.

The PARM parameter TIMEOUT lets you override the default timeout value of 60 seconds. Note that this parameter is effective only on the first system where I/O operations is started. Otherwise this value is received from any other I/O operations system that communication can be established with. For details refer to the RESET TIMEOUT command in *IBM Tivoli System Automation for z/OS Operator's Commands*.

2. You stop I/O operations by issuing one of the commands /P INGEIO or /P IO.

---

## Starting and Stopping Processor Operations

1. Processor operations is started as an application running on the automation NetView at the focal point system. To start processor operations, issue the following command from a NetView operator console:

```
ISQSTART data_set_name
```

where *data\_set\_name* is the name of the data set containing the processor operations control file built by the customization dialog. For details on the ISQSTART command, refer to *IBM Tivoli System Automation for z/OS Operator's Commands*.

2. You stop processor operations by issuing **ISQSTOP** from the NetView operator console.

---

## Starting and Stopping Applications

Figure 11 on page 59 shows the startup processing of an application under the control of SA z/OS.

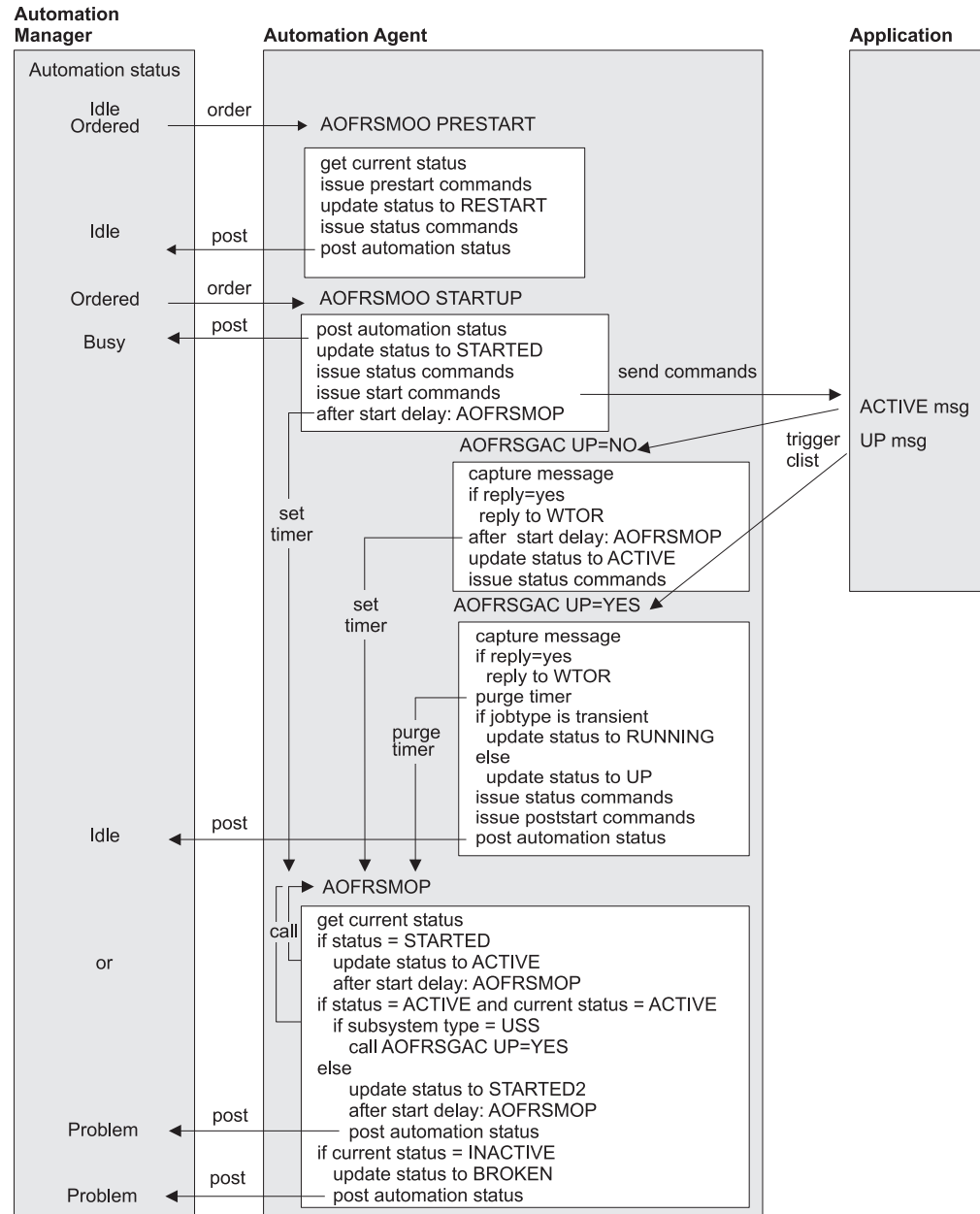


Figure 11. Application Startup Under SA z/OS's Control

Actions such as issuing prestart, startup, poststart, or status commands are only processed if the appropriate automation flag is ON.

The REXX procedure AOFRSGAC is also known by its synonym ACTIVMSG.

Figure 12 on page 60 shows the stop processing of an application under the control of SA z/OS.

# Starting and Stopping Applications

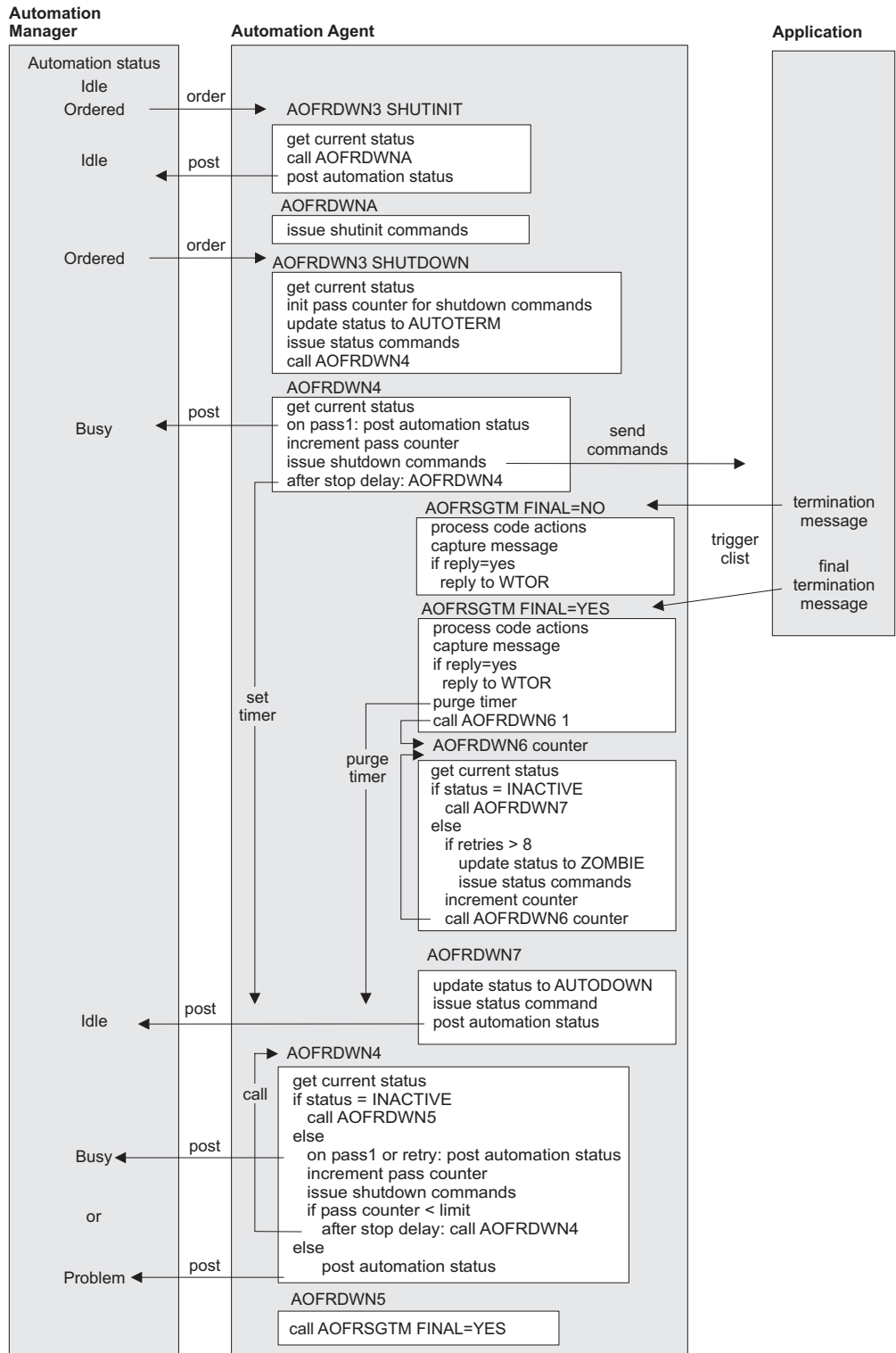


Figure 12. Application Stop Under SA z/OS's Control

Actions such as shutinit, shutdown, or status commands are only processed if the appropriate automation flag is ON.

The REXX procedure AOFRSGTM is also known by its synonym TERMMSG.

---

## Chapter 4. Refreshing Automation Policy

The primary method to refresh the automation policy is to issue the command:

```
INGAMS REFRESH
```

This command operates sysplexwide. All systems in the sysplex are refreshed automatically. It supports incremental updates on a system basis, that is, only the data that is affected by changes in the policy database is refreshed.

Besides the method of using the INGAMS command with the REFRESH option, there are other ways to perform the task of refreshing the automation policy. These are described in this chapter.

To help you dynamically change policy after you have initialized SA z/OS on your system, SA z/OS lets you:

- Load a different automation configuration file; this includes the automation control file and the NetView automation tables
- Reload an automation configuration file that you have changed
- Restore an automation configuration file that has been previously saved to a warm start cache

The automation control file load is normally performed at system operations initialization (NetView start). It is automatically triggered, if the automation manager configuration file (AMC file) is changed. It can also be triggered from an operator by entering an ACF LOAD/REFRESH command. You do normally not have to know the correct automation configuration file name. It is automatically determined by SA z/OS.

At an ACF load the automation manager is contacted and queried for the 'Configuration Token' to make sure that the ACF that will be loaded is compatible with the AMC file.

If you issue an ACF command and there is no automation configuration file that is compatible with the AMC file, the load is not performed and an error message is issued.

The ACF REFRESH is able to do incremental updates of an ACF that is either in storage or in the save/restore cache. An ACF load will normally not be necessary, in most cases an ACF REFRESH is sufficient. The initialization processes the different data on several autotasks in parallel. Therefore there is no need to disable automation during a refresh.

**Note:** For a description of the complete ACF syntax, refer to *IBM Tivoli System Automation for z/OS Operator's Commands*.

To reload the NetView automation tables that are specified in the System Info policy, use the following command:

```
ACF ATLOAD
```

You can check the consistency of ACF data with the automation manager configuration and test the automation tables that are specified in the System Info policy with the following command:

```
ACF CHECK
```

See *Tivoli NetView for z/OS Command Reference* for more information on the AUTOTBL command.

---

## Refreshing an Automation Manager Configuration

The REFRESH subcommand of the MVS MODIFY (F) command may be used to initiate a configuration refresh operation, and is effective only when directed to the primary automation manager.

```
F jobname ,REFRESH,new_data_set_name
```

The *new\_data\_set\_name* may be specified in several forms, as follows:

*dsname*

This form specifies the fully qualified data set name to be used.

*gdgname(generation)*

This form specifies a generation data group name and the generation group member to be used. The generation number is relative to the most recently created member, and may be 0 (for the most recent) or a negative number (-1,-2, ..., -*n*) where *n* is the number of generations specified when defining the GDG, minus 1.

\* This specifies that the refresh operation should use the same data set (and GDG member) as was used on the last successful refresh operation or during automation manager initialization.

\*(*generation*)

This specification form specifies that a member of the same GDG used for the previous refresh operation is to be used. The generation number is relative to the most recently created GDG member at the time of command entry.

Use this command when no automation agent is running.

### Example:

```
REFRESH,*(0) - use most recent generation of the specified GDG
```

---

## Identifying Current Configuration Files

The automation configuration file is the file that controls system-level automation. It contains the SA z/OS automation policy, which specifies what, when, and how to automate. The NetView automation table specifies which automation procedure to run when various messages are received.

An automation configuration file is made available during SA z/OS initialization. The NetView automation table is loaded into NetView, and the automation control file can be either loaded into SA z/OS or restored from a warm start cache.

You can use the DISPSYS command to identify the automation configuration file that is currently active. See Figure 13 on page 63 for an example of the information that is displayed.



```

Configuration
Data set      : OPER1.USER.KEYPLEX.ACF(ACFZ995)
Built by     : OPER1 04/11/04 10:20:36
Activated    : 04/11/04 10:21:19
CFG Token    : 200404051055068A0345169672

Message automation table(s)
DSITBL01 INMSG01
    
```

Figure 13. DISPSYS Command Dialog

To identify the current automation manager configuration file, you can use the diagnostic facilities of the INGAMS command (see also “How to Use the Diagnostic Option of INGAMS” on page 237). From the INGAMS command dialog, you can select option **B Show Details** for an automation manager. A panel similar to the following is displayed, where relevant information concerning the AMC file is shown.

```

Config dataset name : OPER1.USER.KEYPLEX.ACF
Config member       : HSAZ999
                   Z999CLGC  STRUCTURE  20040405105506
                   Z999CRES  STRUCTURE  20040405105506
                   Z999CSCH  SCHEDULE   20040405105506
Config token        : 200404051055068A0345169672
Config version      : 01
    
```

Figure 14. INGAMS Command Dialog

## Identifying Current Configuration Files

---

## Part 2. Using SA z/OS on the Host

This part describes how to use SA z/OS in a host session. It contains the following:

- Chapter 5, "Getting Started," on page 67
- Chapter 6, "How to Get Resource Information Sysplexwide," on page 69
- Chapter 7, "How to Monitor and Control Resources," on page 79
- Chapter 8, "Scenario: Finding Out Why a Resource Is Not Available," on page 147
- Chapter 9, "Using I/O Operations," on page 157
- Chapter 10, "SA z/OS Discovery Library Adapter," on page 163
- Chapter 11, "VM Second Level Systems Support," on page 165
- Chapter 12, "LPAR Management Functions," on page 173



---

## Chapter 5. Getting Started

Different sets of commands are available from host sessions:

- Automation manager commands
- System operations commands (automation agent commands)
- Processor operations commands
- I/O operations commands

With system operations commands, you can control and maintain all of the resources sysplexwide from a single point of control. They operate in two modes:

- Fullscreen mode: If it is a command to display information and you just enter the command name, a panel is displayed showing all available resources. On this panel, you can specify further actions/commands for a special resource. If you enter a command to maintain a resource without further parameters, a fullscreen panel prompts you for more information.
- Line mode: From a command line, you can enter the complete syntax of a command to receive either the desired output directly or to manipulate the resource you wanted in the way you wanted.

All system operations commands (automation agent commands) also operate in fullscreen mode and line mode unless stated in the reference documentation that the OUTMODE parameter is not supported.

The tasks that you can manage with the available commands are described in more detail in the subsequent chapters and sections. If you need the complete syntax of these commands, refer to *IBM Tivoli System Automation for z/OS Operator's Commands*.

---

### Issuing Commands

With SA z/OS, there are several ways that you can issue a command:

- You can use command dialogs to enter command parameters and issue commands using 3270 panels. See "Using Command Dialogs" on page 68.
- You can use an object's popup menu on the NMC workstation.

SA z/OS adds a selection of predefined processor operations and system operations commands to the NetView Management Console. This enables you to use the NetView Management Console as a front end for managing your z/OS host environment.

The sets of available commands are provided in corresponding response files. If you want to add or remove commands from the delivered sets, you can use the *Command Profile Editor (CPE)*. For information on how to use this editor, refer to *NetView Management Console User's Guide*.

For information about how to use this graphical command interface, refer to Chapter 13, "Using the NetView Management Console for SA z/OS," on page 179.

- You can enter I/O operations commands at the z/OS operator console. See "Using the z/OS Operator Console for I/O Operations Commands" on page 160.

## Issuing Commands

- You can issue processor operations, system operations, and automation manager commands from the same interface via NetView terminals as described in “Using Command Dialogs.”
- I/O operations commands can be issued from ISPF dialog panels as described in “Using the I/O Operations ISPF Dialog” on page 160.

For each method, instructions on the window or panel tell you what to do. The commands are described in greater detail in *IBM Tivoli System Automation for z/OS Operator's Commands*.

---

## Using Command Dialogs

Enter A0C from the NetView command line in your NetView session to reach the System Operations MAIN MENU.

Enter 2 or CD on the command line to invoke the Command Dialogs Main Menu. Starting with this menu, you can interact with 3270-type panels to issue SA z/OS system operations commands.

You can enter a command with two methods:

1. Type the number or letter corresponding to the desired command at the command line,
2. Type the name of the command. If you choose to type a command, you can add options to it.

For example, if you enter number 2 at the command line of the Command Dialogs Main Menu, you reach the AOCTRACE command dialog that prompts you for further options.

Some of the commands operate sysplexwide; especially those sent to the automation manager. For other commands you can specify a target system that they should operate on, for example, if you want to display the automation flags on a certain target system.

The commands present a snapshot of the data, taken at the time the command is executed. Some data may be refreshed when other commands are issued from the same panel. To force a refresh of all data you must press PF9.

---

## Chapter 6. How to Get Resource Information Sysplexwide

This chapter describes commands that you can use to retrieve information from all resources defined within a sysplex from a single point of control. The commands are presented here in an overview. Each command is documented with sample screens and the complete syntax in *IBM Tivoli System Automation for z/OS Operator's Commands*.

SA z/OS provides commands that let you look at resources from two different point of views:

- From the point of view of the automation manager. This lets you see what state a resource is in or what votes are currently active against this resource. These commands are described in “Commands That Are Processed by the Automation Manager”
- From the point of view of the automation agents. This lets you see thresholds or startup and shutdown commands that have been defined for the resources. These commands are described in “Commands That Are Processed by Automation Agents” on page 73.

---

### Commands That Are Processed by the Automation Manager

The automation manager decides how automation is performed for the known resources in the sysplex. To decide how to handle the requests that it receives through the various channels (for example, the defined automation policy or interactive operator requests), the automation manager uses an internal hierarchy of priorities that is given in Table 2 on page 15.

SA z/OS offers a series of commands that you can use to find out what information the automation manager currently holds about resources, for example, relationship information or information about the status of resources and whether schedules exist for them. These commands are introduced in the following sections.

#### Using INGLIST to View Resources

This command provides the details about the resources in the sysplex from the point of view of the automation manager. The following information is displayed for a resource:

- Name and type of the resource
- Compound status
- Current status (observed)
- Desired status
- Automation status
- Startability status
- Health status
- Automation flag
- Hold flag
- Description
- Schedule name, if any
- Trigger name, if any

## Commands that Are Processed by the Automation Manager

- Start type
- Stop type
- Resource category and nature

You can see more information about the resources by scrolling horizontally using PF11. Some of this information, especially the automation status, changes in real time. By pressing PF9 (Refresh), you can obtain a new set of data for the displayed subsystems.

On this panel, you can use PF5 to invoke a filter dialog (which is equivalent to using the INGFLT command available from any command line). With this filter, you can reduce the amount of information shown in the display. It enables you to setup a specific operator view for an operator. For example, you can set up a filter that shows only certain resources in specified observed and desired states.

You can also use the INGLIST command dialog to launch other actions that are related to resources or groups, that means, you can also this command to maintain and control resources.

The CMD column on the left hand side of the screen allows you to issue any of the commands listed across the top of the screen against any of the resources that are displayed. Enter the letter corresponding to the command beside the resource you want it issued against and press enter. When you return from the command there may be a slight pause as the display refreshes itself.

The available commands are indicated at the top of the INGLIST panel:

A Update	B Start	C Stop	D INGRELS	E INGVOTE	F INGINFO
G Members	H DISPTRG	I INGSCHED	J INGGROUP	M DISPMTR	/ scro11

**Note:** If you have defined exit routine AOFEXC04, there is an additional action code U. See *IBM Tivoli System Automation for z/OS Customizing and Programming* for details.

On the INGLIST panel, you can use PF4 to toggle to the DISPSTAT command dialog, which gives you the automation agent view of the resources in the sysplex (see “Displaying the Automation Agent View with DISPSTAT” on page 73). 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 but this might not be the same as the original INGLIST display since the DISPSTAT display might have been modified due to filtering or suppressing of resources that the automation agent does not handle.

You can use PF5 to define a filter as described in “Specifying Filters.”

### Specifying Filters

You can use the INGFLT command dialog to specify or revise filter criteria. The filter settings are saved in task global variables, so that when you next run INGLIST they will be used again. Press PF9 to save the currently displayed filters in these task global variables.

You can specify one or more of the following filters:



## Commands that Are Processed by the Automation Manager

### Resources

The list of resources in name/type/system format. You can also specify a wildcard.

### Observed status

The list of observed statuses. You can abbreviate the status, for example "av" for Available. Any resource whose observed status is in the list is eligible for display.

### Desired status

The list of desired statuses. You can abbreviate the status. Any resource whose desired status is in the list is eligible for display.

### Automation status

The list of automation statuses. You can abbreviate the status. Any resource whose automation status is in the list is eligible for display.

### Compound status

The list of compound statuses. You can abbreviate the status. Any resource whose compound status is in the list is eligible for display.

### Health status

The list of health statuses. You can abbreviate the status. Any resource whose health status is in the list is eligible for display.

### Automation flag

The automation flag for the resource. Specify YES or NO.

### Resource category

The list of categories the resource belongs to. It is one of the following: CICS, IMS, DB2, OPC, JES2, JES3, USS, or IMAGE.

### Resource subtype

This specifies the region subtype for CICS, DB2 and IMS or the type for TWS applications, for example, AOR for CICS or IRLM for DB2.

### Group type

The type (nature) of the resource group. You can specify more than one group type.

You can abbreviate the status in the status lists, for example, av for Available. In order to be eligible for the display, a resource must match all filter criteria. An asterisk means that the filter is not set. A '^' or '\' in front of the status displays all resources whose status is not what you specified.

Use the PF4 key to clear the currently established filter settings. The filter will then contain an asterisk, meaning that the filter is not set, or a blank. Use the PF5 key to revert to the currently established filter settings.

**Note:** If INGFLT was called from INGLIST, the filters that are displayed will take effect when you press Enter. If you press PF9 first, the filters will be saved.

## Using INGINFO to View a Selected Resource

The INGINFO command displays details about a specified application or application group from the viewpoint of the automation manager. If you enter this command without parameters, a dialog prompts you for more information. For a detailed description of the command syntax refer to *IBM Tivoli System Automation for z/OS Operator's Commands*.

## Commands that Are Processed by the Automation Manager

You can also issue this command against any resources that are displayed in the result panel of the INGLIST command. In the resulting panel you see information about the resource's different statuses, automation flag settings, whether dependencies are satisfied and what votes are currently active against the resource.

On the INGINFO panel, you can use PF4 to toggle to the DISPINFO command dialog, which gives you the automation agent point of view of the current resource (see "Using DISPINFO to Display Detailed Information" on page 76).

### Displaying Application Group Information

The DISPAPG command displays detailed information about a specified application group. If you do not specify a application group, a menu is displayed where you can specify the application group name. Use PF9 to refresh the information about the current application group and PF4 to invoke the INGINFO command to get details about the application group from the automation manager's point of view.

### Displaying Monitor Information

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

- 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** Lets you reset the health status of the resource being monitored to NORMAL.
  - B** Lets you start the monitor resource (make it available).
  - C** Lets you stop the monitor (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** field shows the name of the monitor.
- The **System** field shows the name of the system where the monitor runs.
- The **Status** field 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** field shows when the resource was last monitored.

## Commands that Are Processed by the Automation Manager

- The **Status message** field shows message that is associated with the status. If the message has been truncated it will end with "..." -- use command code D to view full details about the message.
- The **Description** field shows descriptive information about the monitor resource.

Use the PF10 and PF11 keys to scroll horizontally between the screens of information on the panel.

## Displaying Relationship Information with INGRELS

The INGRELS command displays a panel that shows the relationships that are defined for the specified resource and the status of the current conditions (whether they are satisfied or unsatisfied).

The INGRELS command dialog shows the relationships that have been defined for the current resource in both directions, so in the **Dir** column from the INGRELS command dialog there is:

- A **B** for backward relationships, that is, the relationship exists from the partner resource to the specified resource
- An **F** for forward relationships, that is, the relationship exists from the specified resource to the partner resource

Relationship conditions are also shown (with signature *Cond:*) and whether they are unsatisfied.

The level of detail that is displayed can be changed using the following keys:

- + Shows the condition details that exist for the selected relationship. This is only applicable if the relationship conditions were previously collapsed.
- Collapse the condition details. Only the relationship will be shown.

---

## Commands That Are Processed by Automation Agents

An automation agent is installed on each system with resources that need to be automated by SA z/OS. For example, automation agents are responsible for:

- Having the appropriate commands processed for subsystem startup and shutdown
- Reporting status updates to the automation manager
- Message automation

Automation agents access information for their tasks from the automation control file.

SA z/OS offers some commands that you can acquire information with about resources running on the system where the automation agent resides.

## Displaying the Automation Agent View with DISPSTAT

The DISPSTAT command displays fullscreen panels showing the following information for resources defined to the current automation agent:

- Subsystem name
- Automation status (not dynamically refreshed)
- The system the instance of the subsystem is running on
- Job name
- Automation flag setting overrides

## Commands that are Processed by Automation Agents

- The type (MVS, Transient, One Shot, or Non-MVS) of the subsystem
- The date and time its automation status last changed
- The date and time its automation status was last confirmed by active monitoring
- A user supplied description of the resource

Status conditions are color-coded to make it easier to identify problems.

**Note:** If you have defined exit routine AOFEXC04, there is an additional action code U. See *IBM Tivoli System Automation for z/OS Customizing and Programming* for details.

You can cycle through further information using the PF10 and PF11 keys.

In cases where a wide (132 column) display is used, all of the data can be displayed on two screens. On a smaller 80 column display there is a third screen that displays Description data.

Some of this information, especially the automation status, changes in real time. By pressing PF9 (Refresh), you can obtain a new set of data for the displayed subsystems.

You can use PF4 to toggle to the INGLIST panel to see the same resources as shown in the DISPSTAT panel but from the automation manager point of view. To return to the DISPSTAT panel press the PF3 key on the INGLIST panel. Note, pressing PF4 on the INGLIST panel displays the DISPSTAT panel for all resources shown in the INGLIST panel but this might not be the same as the original DISPSTAT display since the INGLIST display might have been modified due to filtering.

From the DISPSTAT display you can initiate a number of other commands:

### **INGAUTO**

This lets you change the subsystems automation flags

### **SETSTATE**

This lets you change the subsystems automation status if it is inactive.

### **INGREQ - STOP**

This lets you shut down the subsystem.

### **thresholds - INGTHRES**

This lets you review the subsystems threshold settings.

### **EXPLAIN**

This presents an explanation of the subsystems current automation status.

### **DISPINFO**

This presents a detailed display of the subsystem.

### **DISPTREE**

This presents a view of the target systems parent/child graph, focused on the subsystem.

### **DISPTRG**

This shows details of what trigger, if any, is associated with the subsystem.

### **INGSCHED**

This shows details of the service period that has been defined for the subsystem.

## Commands that are Processed by Automation Agents

All of these commands work in the same way as DISPSTAT - fetching it from the target system and presenting it to you on a full screen panel.

On the DISPSTAT panel, you can use PF5 to invoke a filter dialog, which is equivalent to using the DISPSFLT command available from any command line. With this filter, you can reduce the amount of information shown in the display. It enables you to setup a specific operator view for an operator. For example, you can set up a filter that shows only resources of a certain name or that are in a specific state. For more information refer to *IBM Tivoli System Automation for z/OS Operator's Commands*.

The panel displays the filters that will be used for the DISPSTAT display. The filters are stored in task global variables, and you can update them by pressing PF9. When the panel is invoked from within DISPSTAT, pressing Enter will take you to the DISPSTAT display, using the filters that are currently displayed. The filters are not saved unless you have previously pressed PF9 in the DISPSFLT command dialog.

The filters you can set are:

- **Resources:**

You can specify a list of one or more subsystem names. They can have leading or trailing wildcards (for example, \*VIEW, NET\* and \*TVIE\* are okay, but NE\*EW will not work).

You can also specify a list of job names instead of subsystem names.

- **Statuses:**

You can specify a list of automation statuses to be included or excluded from the display. This filter is used whenever the display is refreshed.

SA z/OS provides the following sets of pre-grouped statuses, which can be abbreviated by their numbers:

1. All resources that are currently UP or ENDED
2. All resources that are in a normal status: DOWN, STARTED, UP, AUTOTERM, AUTODOWN, ACTIVE, ENDED, ENDING, RESTART, EXTSTART, RUNNING
3. All resources that are in a down status: DOWN, INACTIVE, RESTART, AUTODOWN, CTLDOWN, STOPPED, BROKEN, MOVED, FALLBACK
4. All resources that are in a start transition state: RESTART, STARTED, STARTED2, ACTIVE, RUNNING, EXTSTART
5. All resources that are in a stop transition state: AUTOTERM, ENDING, STOPPING, ABENDING, BREAKING

- **Target Systems:**

This lets you specify the systems within the sysplex that you want data gathered from. By default it shows just your local system.

You can specify a list of one or more system, domain, or sysplex names (in any combination), and the query is sent to those systems. Alternatively you can specify '\*ALL', and the query is sent to all active systems within the sysplex.

In general the more specific your query and the fewer systems you send it to, the faster it will execute.

When you have updated the filters (the defaults will do for just exploring), press Enter to go into the DISPSTAT display.

### Using DISPINFO to Display Detailed Information

To see detailed information on a subsystem from the automation agent's point of view, use the DISPINFO command. This will only work on one subsystem at a time. If the subsystem you specify happens to be on more than one system within the sysplex, you will be prompted to select the one you would like to see information about.

The DISPINFO command displays information about the selected resource, in the following example, the APPC resource. As you can see, it displays a lot of the definitions that have been made to SA z/OS about the subsystem. You can use PF8 to scroll forward to see more information about this resource. You can use PF4 to toggle to the INGINFO command display for this resource, which shows the automation manager view of it.

You can use PF8 to scroll forward through the pages of information, and PF7 to scroll back.

Use PF8 until you reach a display line like the following that indicates the end of the display:

```
*** Data produced at 17:58:17 on 07/26/05.
```

If you attempt to scroll further down than the last page, an error message is displayed:

```
AOF169I THE LAST LINE IS ALREADY DISPLAYED ON THE SCREEN
```

**Note:** If you have defined the exit routine AOFEXC03, the DISPINFO panel also shows user supplied information about the subsystem. Refer to *IBM Tivoli System Automation for z/OS Customizing and Programming* for more information about AOFEXC03.

If you use DISPINFO for a CICS-controlled or IMS-controlled resource, PF10 is active and with this you can view additional CICS or IMS related information as if you issued the INGCICS REQ=INFO or INGIMS REQ=INFO commands.

For a CICS resource it displays the version (both CICS and Transaction Server), the Indoubt Unit of Work status, the last start type, the RLS status, and the CICSplex/SM CMAS region of the CICS system.

For an IMS resource it shows the Version of IMS that is running, the subsystem name of the shared queues control task, the subsystem name of the Fast Database Recovery region, the command prefix for commands, the status of the availability manager and data communication, and also IMS checkpoint information.

DISPINFO shows further information for resources that have been defined with INGVTAM startup and shutdown commands. It displays the specific APPLID and optionally the ACB (if it is different from the APPLID) that the resource is using. It also shows the generic APPLID and USERVAR if any have been defined. Finally a list of major nodes that have been registered as being used by the resource is displayed.

### Using DISPTREE to Display Dependency Information

Generally if a resource has not been started, it will be because of a problem with one of its supporting resources. You may also wish to know which other subsystems will be impacted by problems with a particular subsystem. To find out what the problem is, use the DISPTREE command.

## Commands that are Processed by Automation Agents

The DISPTREE command presents a view of the start dependencies, stop dependencies or the group membership of the specified subsystem (resource). Supporting resources are shown above and to the left of the specified subsystem, its dependent resources are shown below and to the right.

The panel uses reverse video to improve the highlighting. Non-subsystem resources are indicated with |.

PF4 will refocus the display onto the subsystem under the cursor. PF5 will invoke DISPINFO against the subsystem under the cursor. However, you cannot use DISPINFO on non-subsystem resources.

### Displaying Statistical Information

The INGRPT command displays a panel providing statistical information about the automation agent and some basic information about the automation manager of a system in a sysplex.

The summary report lists information about, for example:

- The number of resources defined to NetView and managed by the automation agent
- The total number of messages automated
- The total number of resulting commands
- The total number of START and STOP commands issued
- The number of systems in the sysplex
- The total number of resources managed by the automation manager

If you want to see detailed statistical information for a particular resource, issue the command INGRPT STATS=DETAIL. A panel is displayed that shows statistical information about all resources that are automated by the automation agent.

For each resource the following information is shown:

- The number of messages automated
- The number of commands resulting from the message traps
- The number of START and STOP commands that were issued
- The number of critical threshold conditions that have occurred

You can select a resource to display details by entering S in the Cmd column.

### Displaying Application Group Information

The DISPAPG command displays detailed information about a specified application group. If you do not specify a application group, a menu is displayed where you can specify the application group name. Use PF9 to refresh the information about the current application group.

### Displaying Monitor Information

The DISPMTR command displays and allows you to manage monitors that you have defined using the customization dialogs for your system, as described in "Displaying Monitor Information" on page 72.

Use the PF10 and PF11 keys to scroll horizontally between the screens of information on the panel.

## Commands that are Processed by Automation Agents



---

## Chapter 7. How to Monitor and Control Resources

This chapter introduces commands that you can use to control resources. Controlling resources in this context means making them available or unavailable.

Besides the principal methods for controlling resources as introduced in Chapter 2, “Concepts,” on page 13, SA z/OS offers more sophisticated techniques to the operator.

With the commands introduced in this chapter, you can, for example:

- Start or stop a resource or application group
- View resource or group details
- Turn on or off the automation flag for a resource or application group
- View and modify service periods, referred to as schedules
- Manage sysplex resources
- Display resource statistics
- View trigger definitions

Table 7 shows the various tasks that are discussed in this chapter along with the main commands that you can use to perform them. For more detailed information about the commands refer to *IBM Tivoli System Automation for z/OS Operator's Commands*.

Table 7. Operator Tasks and Related Commands

Task	Commands
“Starting and Stopping Resources” on page 80	INGREQ, INGVOTE
Displaying history information, “Using INGHIST” on page 83	INGHIST
“Using Schedules” on page 84	INGSCHED
“Displaying and Setting Trigger Information” on page 85	DISPTRG, INGTRIG
“Displaying and Setting Events” on page 85	DISPEVT, DISPEVTS, INGEVENT
“Determining Automation Agent Flag Values” on page 86	DISPFLGS, DISPSCHD, DISPSTAT, INGAUTO
“Working with Application Groups” on page 88	INGGROUP, DISPAPG
“Moving Sysplex Application Groups” on page 92	INGMOVE
“Setting Timers” on page 94	SETTIMER
“Interacting with the Automation Manager” on page 94	INGAMS
“Changing the Automation Agent Status” on page 95	SETSTATE
“Monitoring Messages” on page 97	INGNTFY, ISQCMON, ISQXMON
“Monitoring Hardware” on page 100	ISQXDST
“Monitoring Health Performance” on page 102	DISPMTR, INGLIST
“Monitoring OMEGAMON Sessions” on page 103	INGSESS
“Monitoring Processor Status” on page 107	ISQXDST

Table 7. Operator Tasks and Related Commands (continued)

Task	Commands
“Monitoring with the Status Display Facility” on page 114	SDF
“Monitoring and Controlling a Sysplex” on page 122	INGPLEX, INGSTR
“Displaying Threshold Settings and Occurrences” on page 139	DISPASF, DISPERRS, INGTHRES
“Handling Jobs That Are Not Controlled by SA z/OS” on page 140	INGLKUP
“Communicating with Other Systems” on page 140	INGSEND, ISQSEND

## Starting and Stopping Resources

Besides goal driven automation that you exploit by defining automation policy, SA z/OS offers a series of commands that let the operator react to special circumstances beyond regular automation. The commands for such purposes are introduced in the subsequent sections. Each command is documented with the complete syntax and some further sample output screens in *IBM Tivoli System Automation for z/OS Operator’s Commands*.

### Using INGREQ

As described in “Automation Concepts” on page 13, commands issued with INGREQ and a high priority are the most powerful automation commands to make applications available or unavailable. They supersede all other requests that exist against a resource at that time.

You can use this command to:

- Initiate the start process of one or more specified application resources or application group resources
- Initiate the shutdown process of one or more specified application resources or application group resources

A request is persistent until:

- You cancel it using the INGVOTE command
- It expires according to the specified EXPIRE parameter
- It is automatically removed when a specified condition comes true
- You want it to be cancelled after a specified timeout period, if the request has not been satisfied after that time
- It is overwritten by a request from the same source

Even though requests are persistent, it is not guaranteed that they will be satisfied by the automation manager.

With a request you specify two categories of parameters:

- How should the request be executed (for example, with low or high priority)?
- What should happen with the request during its lifetime, for example, should it expire after a certain time anyway, or after a timeout if it has not been successful?

If you specify the command name without any parameters, SA z/OS displays a panel where you can specify, for example, startup or shutdown parameters, a priority, and a timeout for the request.

Use PF11 to see additional parameters.

After pressing Enter to submit the request, you will see a verification panel that lists all resources that are affected either when posting an INGREQ command to the automation manager or when cancelling a previously made request. When making a request, a resource is affected if it is within the dependency graph that the request is propagated along. This will also display resources that are already in the desired state. In this case nothing will happen to the resource.

When cancelling a request, a resource is affected when it holds a vote for the request to be removed. Only resources that are highlighted will be processed by the command. All other resources are already in the state satisfying the appropriate dependencies.

### Starting Subsystems

To start a subsystem it must be put into either the DOWN or RESTART automation status that is seen by the automation agent. The DOWN status occurs just after SA z/OS initialization and indicates that the subsystem has not been active since SA z/OS was last started.

To put a subsystem into the RESTART status, you can use the SETSTATE command.

### If Subsystems Do Not Start

If you have asked for a subsystem to start and SA z/OS refuses to do it, there are some things you can check.

At first examine the AOF313I message. This should clearly state why the startup request was rejected. If there isn't one, search your netlog for more information.

- **Supporting Resources Not Up**

If supporting resources are not up, SA z/OS tries to start them. If you wait a while they should come up and then the target subsystem will be started.

You can check on the state of the parents through the DISPTREE display, including status, or through messages issued to the NCCF console. If for example, it turns out that a parent is in a status of BROKEN, it is not possible to start the desired subsystem until this problem is fixed.

- **Automation Flags Turned Off**

Automation flags may be turned off for the subsystem or for just the INITSTART or RESTART actions. If the INITSTART flag is turned off, the subsystem cannot be started from a DOWN status. If the RESTART flag is turned off, the subsystem cannot be started from a RESTART status.

To see the flag settings for a subsystem you can use either DISPFLGS or DISPINFO. DISPFLGS will show you both the current settings in force for the subsystem and, perhaps more importantly, the settings that were made that resulted in those values.

- **Trigger Not Satisfied**

Triggers tell SA z/OS when it is allowed to stop and start subsystems. If a trigger has been linked to a subsystem, SA z/OS will not stop or start that subsystem unless the trigger says it is OK to do so. Normally this function is fine, but there may be times when you need to override it.

## Using INGREQ

To find out about the trigger linked to a subsystem, use the DISPTRG command. For example, the following command shows the triggers and events that are associated with the resource TEST0:

```
DISPTRG TEST0,TARGET=KEY3
```

If you want to start something and override the event settings you can either issue an INGREQ REQ=START OVERRIDE=TRG command or with the INGREQ command, you can specify that trigger conditions should not be considered by specifying option *Override ==> TRG* in the INGREQ command dialog panel . If you want to ensure that the trigger is satisfied before you start the resource, you can set it with INGEVENT. INGEVENT can be used to set an event (but may affect more than just the subsystem you are dealing with).

- **The Automation Manager Automation Flag Is Off:**

Use the INGINFO command or the INGLIST command to find out what the current setting is. If the automation flag is off, the automation manager will not send the order to start the subsystem to the automation agent.

For a detailed scenario on how to use SA z/OS diagnosis methods, refer to Chapter 8, “Scenario: Finding Out Why a Resource Is Not Available,” on page 147.

### If Subsystems Do Not Stop

If you have asked for a subsystem to stop and SA z/OS refuses to do so, here are some things you can check:

- **Supporting resource cannot stop:** the supporting resources cannot be brought in the condition fulfilling the stop dependency, so that the dependent resource that you want to stop, also cannot stop. This is the case if the compound status indicates: AWAITING.
- **Request with higher priority to have the subsystem UP:**  
Use the INGVOTE command to determine if a higher priority request exists for the subsystem. If so, you can either reissue the stop request with a higher priority thus overruling the other request or cancel the other request.
- **The shutdown trigger is not satisfied:**  
Use the DISPTRG command to find out the trigger conditions defined for the subsystem. Watch the shutdown conditions that are defined for the subsystem. You can either set one or more events bringing the shutdown condition into a satisfied state or use the INGREQ OVERRIDE=TRG parameter to bypass the trigger check.
- **Automation flags turned off:**  
The automation flag may be turned off for the subsystem or just the terminate action. To see the flag settings, use DISPSTAT, DISPFLGS, or DISPINFO.
- **The automation manager automation flag is off:**  
Use the INGINFO command or the INGLIST command to find out what the current setting is. If the automation flag is off, the automation manager will not send the order to stop the subsystem to the automation agent.
- **Check history:**  
Use the INGINFO command to analyze the history of the resource.

## Using INGVOTE

Whenever you (or a program) want the automation manager to do something, you must send it a request. Requests are remembered within the automation manager. They are made against individual resources and then propagated across the relationship graph to other resources within the automation manager. If, for example, your request is to start resource A, and A has a relationship to B such

that B needs to be started before A can run, the automation manager generates a vote to resource B to start it. Thus a request can be viewed as a black box of related votes that is generated by the automation manager.

If you enter the INGVOTE command without any parameters, it displays all pending requests that are persistent at the automation manager.

If you enter the INGVOTE command specifying a resource, or launch it from the INGLIST command dialog, SA z/OS displays all pending votes for that resource with the following information:

- The name of resource the request was made against
- The request action
- The source of request
- The request priority
- The request vote

From the INGVOTE command dialog, you can enter the following commands against a selected vote:

- C** Cancels the request. This command displays the verification panel that lists all resources that are affected if you want to cancel the selected vote/request (see also “Using INGREQ” on page 80).
- K** Cancels the request without showing the verification panel.
- S** Shows details about the request.
- V** Displays all the requests or votes that have been entered for the selected resource.

## Using INGHIST

The INGHIST command is a sysplexwide command issued against the sysplex’s automation manager. It displays history information about the work items processed by the automation manager. You can specify a certain resource that you want SA z/OS to display the work history for, or you can specify a time interval that you want to see all activities for, or you can combine both criteria.

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.

If you enter the INGHIST command without parameters, you receive the output for the complete sysplex for the period starting from the last hour before you entered this command.

In the **Cmd** column on the left of the INGHIST panel, you can expand each work item with the + or e command to see information about the detail actions that have been performed by the automation manager and what resources were involved.

You can collapse the expanded work items again with - or c.

---

### Using Schedules

Using service periods (also called schedules) is one of the methods to set goals for resources. Service periods are defined as permanent automation policy in the customization dialog. To display all the service periods that are defined in the sysplex, use the INGSCHED command dialog .

On the INGSCHED panel, you can enter the following command codes for a selected schedule:

- A** Using this action code to view the From-To time windows of the selected service period for the week. No updates are possible on this panel.
- B** Displays the actual From-To time windows for the selected schedule including the overrides, starting from the requested day (initially it is the current day). Use this option if you want to override a schedule for all resources that use this schedule (that is, for all resources that this schedule is linked to in the customization dialog). Such an override is called a *schedule override* and is described in "Schedule Override."
- C** Displays the resources that are using the selected schedule. You can select a resource to view the schedule that is used for this resource. You can override the schedule just for the selected resource. Such an override is called a *resource override* and is described in "Resource Override."

The column with the **UP** and **DN** entries indicates whether the schedule is an UP or DOWN time window.

### Schedule Override

You can interactively override a schedule for all resources that use this schedule. You can interactively override the schedule starting from a specified starting date for selected dates in the near or distant future.

To reach the schedule override panel, use action code **B** against a schedule on the INGSCHED panel, or enter the INGSCHED command with the schedule name as a parameter. Schedules that have been overwritten are shown in yellow. From this panel you can overtype or complete the displayed schedule.

### Resource Override

You can interactively override a schedule for a particular resource that uses this schedule. You can interactively override the schedule starting from a specified starting date for selected dates in the near or distant future.

There are three ways to reach the resource override panel:

- Use action code **C** against a schedule on the INGSCHED panel and then, from the resource list that is displayed, select the resource that you want to override the schedule for.
- Issue the INGSCHED command with the resource name as a parameter, for example:  

```
INGSCHED AGFG2/APL/AOCA
```
- From the INGLIST command dialog (see "Using INGLIST to View Resources" on page 69), issue action code **I** against a resource.

Service windows that are overrides for the resource are shown in pink. You can edit the schedule by overtyping new values.

From the INGSCHED panel, which is specific for the current resource, you can:

- Enter new service times. Pressing Enter displays the schedule override in pink.
- Overtyping or blanking out existing service times. Pressing Enter displays the schedule override in pink.
- Delete a complete schedule override on a particular date (displayed in pink) by entering action code **D** in the command field. The schedule override is removed and the general schedule that has been defined for the current resource is restored and displayed in green.

---

## Displaying and Setting Trigger Information

Triggers are a very powerful tool of goal driven automation. Because they influence the behavior of applications and application groups, the automation programmer needs to be very careful when connecting triggers to resources. For information on event and trigger support refer to *IBM Tivoli System Automation for z/OS Defining Automation Policy*.

In SA z/OS two commands are available to display trigger information:

### INGTRIG

This command shows all triggers that are defined and used by resources within the automation manager scope. You can view all resources that use a trigger by entering the action code **S** in its command field. Entering the action code **S** against a resource on this panel shows the trigger details for the resource (that is, it invokes the DISPTRG command for the resource).

### DISPTRG

This command displays the trigger conditions for a resource. To display the DISPTRG command dialog for a resource, enter, for example:

```
DISPTRG BATB000P100
```

If you enter this command without parameters or SA z/OS cannot find the resource that you specified, SA z/OS prompts you for more information to help locate the resource. If there is more than one resource that matches the resource specification, you are presented with a selection list.

Use command code **S** against a trigger to view details about the events that must be set to activate it. This displays the Trigger Condition List. From this panel you can use the following command codes against an event:

- **S** to set or **U** to unset the events that have been defined for the resource
- **D** to view a list of the resources that use the event as a trigger

---

## Displaying and Setting Events

You can display and set event information at different levels in your enterprise:

- DISPEVT shows resource information for a particular event.
- DISPEVTS shows all events in a sysplex.
- INGEVENT allows you to set or unset an event for a particular resource or all resources on one system, or all systems in the sysplex.

### DISPEVT

You can reach the DISPEVT command dialog by issuing the DISPEVT *event\_name* command or using the command code **D** from the DISPTRG trigger condition list.

## Displaying and Setting Events

This panel shows all resources that have the specified event in their trigger definition. The following information is displayed for each resource:

- The name of the resource
- The resource type
- The name of the system where the resource resides
- The name of the trigger that the resource is linked to
- The status of the event (SET or UNSET). If the **Status** column is blank this means that the status is unknown. This is treated as UNSET.

### DISPEVTS

You can reach the DISPEVTS command dialog by issuing the DISPEVTS command from any command line. This displays the events that are defined for resources in the sysplex and the condition when the event is reset for the resource.

From this command dialog, you can issue the following action codes against an event:

- **D** to show related resources (that is, it invokes DISPEVT for the event)
- **S** to set or unset the event (that is, it invokes INGEVENT for the event)

### INGEVENT

With the INGEVENT command, you can set or unset an event for a specified resource or for all resources in the sysplex that it is defined for.

To reach the INGEVENT command dialog issue the action code **S** against an event in the DISPEVTS command dialog.

**Note:** With the line mode INGEVENT command, you can directly set or unset an event for all resources or a specified resource, for example:

```
INGEVENT BATCHSTP REQ=UNSET RESOURCE=BATB000P200/APL/AOC7
```

Successful completion produces the following message:

```
AOF442I UNSET EVENT BATCHSTP DONE FOR RESOURCE BATB000P200/APL/AOC7  
ON AOC7. - FROM=AOC7  
AOF099I FUNCTION COMPLETED
```

---

## Determining Automation Agent Flag Values

Automation flags can be predefined in the customization dialogs or can be set during run time. Their meanings are explained in detail in the chapter “Automation Concepts” in *IBM Tivoli System Automation for z/OS Defining Automation Policy*

The automation flags that are used by SA z/OS are:

<b>A</b>	Automation
<b>I</b>	Initial Start
<b>S</b>	Start
<b>R</b>	Recovery
<b>T</b>	Terminate
<b>RS</b>	Restart



## Determining Automation Agent Flag Values

Automation flags are predefined by the automation programmer in the customization dialogs. Their meanings are explained in detail in the chapter "Automation Concepts" in *IBM Tivoli System Automation for z/OS Defining Automation Policy*.

To determine the actual flag values that have been set in the customization dialog or during runtime, together with the effective flag values for a particular subsystem, use action code A to call DISPFLGS for this subsystem. It displays the actual and effective flag values for the selected subsystem and for related minor resources.

Each flag has one of the following values:

- There is no explicit setting.
- Y The flag is turned on.
- N The flag is turned off.
- E The value of the automation flag depends upon the values returned by one or more user exit and will be determined when the value is required.
- L Resulting commands or replies are logged only.

## Displaying Actual and Effective Automation Agent Flags

Enter DISPFLGS at any command line to display the actual flags that have been set in the customization dialog or during run time, and the effective flags for any resources that do not have explicit flags coded for them.

The DISPFLGS command dialog displays the following:

- The name of the system where the resource resides.
- The resource name. Two types of resources are displayed: major and minor. Major resources are subsystems or generic settings (for example, SUBSYSTEM), while minor resources are specific situations or resources within a major resource. Minor resources are indented one space to the right under their major resource.
- The flags that have been set for the resource.
- The flags that are in effect for the resource.

The following command codes are supported:

- A Allows you to define automation flags for a resource using the same flag settings as the selected resource. Another panel is displayed where you can specify the resource name and optionally overwrite the flag settings.
- C Allows you to modify the automation flags of the selected resource. Another panel is displayed showing the current flag settings where you can overwrite the flag values.
- R Causes to reset 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. You can specify scheduled overrides using the customization dialog. These are specific times when the automation is turned off for a particular flag and resource.

### Displaying Automation Agent Flag Override Settings

The customization dialog allows you to turn automation flags off for a scheduled time period. You can view these time period settings by issuing the DISPSCHD command, which leads to the DISPSCHD command dialog panel.

- The **System** field shows the name of the system where the resource is defined.
- The **Resource** field shows the name of the resource.
- The **FI (flag)** field shows an abbreviation of the automation flag name.
- The **Day** field shows the day of the week that automation is turned off.
- The **Start** and **End** fields show the time periods that automation is turned off.
- The **Flags** field shows the full name and the one character abbreviation for each automation flag.

### Setting Automation On or Off: INGAUTO

You can change the automation setting for a specific automated resource or a group of automated resources using the INGAUTO command dialog. You can turn all automation off or only particular automation flags.

**Note:** Be sure to specify the correct subcomponent name. Although the subcomponent does not exist, the message FUNCTION COMPLETED will appear.

Enter INGAUTO on any command line to reach the INGAUTO command dialog panel.

In the **Resource name** field, type the name of the resource that you want to determine automation flags for. If you specify the value defaults, you set automation flags for all resources that do not have more specific automation flags defined or that do not belong to a group that has flags defined.

You can use the **Interval** field to turn automation on or off for all or selected flags for a specific time period. In this field, type the amount of time you want the automation turned on or off. This time period begins immediately when you press the Enter key.

At the end of the interval, the flag (or flags) will be reset to the value defined in your automation control file. If you leave the **Interval** field blank, the flag (or flags) will remain changed until they are manually reset or updated.

---

## Working with Application Groups

An application group is a resource that both summarizes the status of other resources (its members) and that controls the activation of those members. A special feature is that the members can inherit the group's dependencies.

Each member is a resource, as is the group itself. This means that groups can contain other groups. You can issue commands to start or stop groups as an entity. Also, you can link a group to a service period and thus schedule it automatically.

Detailed information on how to organize applications into a group and how to nest groups into other groups and how to generate resources from groups and applications is provided in *IBM Tivoli System Automation for z/OS Defining Automation Policy*.

The concept of resources allows you to dynamically move applications and groups from one system to another system in the sysplex. This move is achieved by stopping an active set of resources and starting an alternate set in a coordinated fashion. With this concept, you can also start and stop equivalent resources to maintain a desired level of server capacity.

There are two types of application groups:

- *System Application Groups* - application groups that are associated with particular systems.
- *Sysplex Application Groups* - application groups that are associated with a certain sysplex. The application group members can be spread over multiple systems within this sysplex.

Furthermore, groups can have one of three different natures:

### **BASIC**

In a BASIC application group *all* of its components must be available before the group is considered to be available.

### **MOVE**

In a MOVE application group exactly one of its components must be available before the group is considered to be available. In MOVE groups you can specify alternative components that will be started if the primary component fails to start.

### **SERVER**

In a SERVER application group any number of components can be specified that must be available before the group is considered to be available. In SERVER groups you can specify what should happen if its components fail to start.

The automation programmer can set the numbers of members that should be available for a SERVER group by specifying the *Availability Target* when defining the group.

Preference values are used for SERVER and MOVE groups to determine which members should preferably be made available.

For detailed information on how to define application groups and what nature to use for different purposes, refer to *IBM Tivoli System Automation for z/OS Defining Automation Policy*.

You can use the DISPAPG command to display detailed information about an application group from the automation agent's point of view, such as automation agent or automation manager related information or automation policy information.

## **Using INGGROUP to Change Policy for Applications**

The functions provided by the INGGROUP command are:

- Automatic group management to maintain application availability. This includes the activation of backup/alternate resources in response to resource or system failure.
- A superset of the behavior provided by Automatic Restart Manager. While INGGROUP does not provide a true cross system restart, it is able to take recovery actions in a variety of situations where Automatic Restart Manager is

## Working with Application Groups

not. These include application failure, supporting application failure, and 'total active instance' based management (for SERVER groups).

- The ability to move resources away from a system, on either a group or subgroup level or a system level.
- The ability to query and dynamically change a group's policy to stop one instance of a resource and start another.
- The ability to view an estimate of the impact of those changes before committing them.
- The ability to perform actions resulting from the above changes in a delayed, non-disruptive fashion.
- the ability to return resources to their normal systems in a delayed, non-disruptive fashion.

Thus one of the most important functions of the INGGROUP command is moving resources around in a sysplex. Moving applications or application groups from one system to another, that is, stopping an active set of applications (dependent and supporting resources) on one system and starting them on another system without loss of information. This can be advisable for the following reasons:

- To ensure availability of resources throughout a sysplex despite single system failure
- To balance workload in a sysplex
- To enable complete applications to be moved away from a certain system if this is needed for other work or maintenance

Application move is implemented in SA z/OS in two flavors:

- A planned or scheduled move, which is defined by automation policy and where the availability of resources is automatically ensured due to defined availability targets and preference values for the single resources. For detailed information, refer to *IBM Tivoli System Automation for z/OS Defining Automation Policy*.
- An immediate move for emergency reasons; for this purpose, you can use the INGGROUP command that you issue against a group to move it.

When an application has been automatically moved to a backup system after a problem, the policy can be set to return it to its home system either as soon as the home becomes available (causing an application outage) or at the next (scheduled) application outage.

For more information about application move see "Controlling Application Move and Server Management" on page 23.

You can invoke the INGGROUP command dialog either by typing the command name on the command line or by selecting option J from the INGLIST command dialog.

This panel allows you to change the policy of application groups. You may specify:

- The target sysplex for the command via the **System** field. If omitted, this defaults to the local system/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 through the **Action** field.
- Next you must specify 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 matching those names. While applying a policy to a Basic group is not an error, it will not achieve anything either.

- For the Exclude-Avoid-Include option, you can specify a list of systems that the policy is to be set for. Specify a question mark (?) at the **System(s)** line to see a list of systems within the sysplex.

This option indicates that you want to remove movable members from a system (exclude), prohibit SA z/OS from activating movable members on a system (avoid) or that you wish to undo either of the other two actions (include).

**Note:** The AVOID/EXCLUDE options are only applied to the members of the group that are applications. They are not applied to members of a group that are also groups. In other words, these options are not recursively spread out to nested application groups.

- With the PACIFY option, you can make the specified group passive. This option will immediately revoke all votes from all of the group members. This may cause the desired state of some of its members to change, resulting in them being started or stopped.

With the ACTIVATE option, you can make a group active. When a group becomes active, SA z/OS will propagate *MakeAvailable* requests to the group members that are required to make the group available. Those members that are not required will receive a *MakeUnavailable* request. Note that if a member has a preference of 1 it will always be propagated a *MakeUnavailable* request and if it has a preference of 0 (a passive member) it will never be propagated any sort of request.

- With the RESET option you can set the preference value assigned to each member of the group to its initial value.
- With the DEFAULT option you can set the availability target, satisfactory target, group passive attribute and the member's preference value to the value defined in the policy.
- With the ADJUST option you can indicate that you want to temporarily adjust a group's policy.
- With the OVERRIDES option you can display any overrides that have been made with the INGGROUP command for the specified resource groups.

To make your changes active, press PF10. Pressing PF4 from this panel takes you to a panel where you can change the preference policy values for the members of a Server group.

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 via 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.

## Working with Application Groups

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.

---

## Moving Sysplex Application Groups

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.

If you enter the INGMOVE command, the INGMOVE command dialog is displayed. You can specify the name of one or more sysplex application groups, separated by a blank or a comma, and each name can contain a wildcard, for example, `ingmove movplex*`. The panel then displays all sysplex application groups of type MOVE (referred to as *move groups*) that match the specified filter criteria.

You can also specify a WAIT period, which is 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 and, if it is not specified, it defaults to 30 seconds.

On the panel, 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) that has the highest preference value is 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. You can designate the system by specifying:

- The name of the system in the input field.
- An asterisk (\*). This means that the group is moved back to its home system. If more than one home system exists, it is moved to the 1st one in the list.
- An equals sign (=). This is a shorthand 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. Refer to “Using INGREQ” on page 80 for further information.
- C** Stop. Calls the INGREQ command dialog to make the resource unavailable. Refer to “Using INGREQ” on page 80 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. This means that the group is effectively tied to that 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 a panel 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.

## Moving Sysplex Application Groups

**Note:** Pressing PF2 or PF3 has the same behavior as PF11.

---

## Setting Timers

You can set a timer to issue commands at a specific time or interval of time for your convenience. Therefore, you do not need to be present at a console to perform a repetitive task at a certain time. This NetView timer is actually the SETTIMER command that issues a command or list of commands at a specified time or specified interval.

Invoking the SETTIMER command displays a list of existing timers. From this panel, you can perform the following tasks.

### How to Create a Timer

Type add at the command line of the SETTIMER command dialog and press Enter. On the Add/Change Timer panel that appears fill in the fields as required.

**Note:** If you want to add a new timer but use most of the same values as an already-defined timer, type A in the CMD column and modify the SETTIMER panel.

### How to Modify a Timer

On the SETTIMER command dialog panel, enter C in the CMD column for the timer that you want to change. The Add/Change Timer panel is then displayed where you can modify the settings of your selected timer.

### How to Delete a Timer

On the SETTIMER command dialog panel, type D in the CMD column for the timer you want to delete and press Enter.

---

## Interacting with the Automation Manager

You can operate the automation manager itself with the INGAMS command to:

- Display information about all currently registered automation managers and automation agents.
- Change the operation mode of an automation manager from secondary to primary
- Refresh the configuration data used by the automation manager
- Perform diagnostic functions

The INGAMS command displays all currently registered automation managers and automation agents. The **Role** column describes what role the member has, as follows:

<b>AGENT</b>	An automation agent
<b>PAM</b>	A primary automation manager
<b>SAM</b>	A secondary automation manager

You can initiate various activities related to automation managers by entering the following command codes (displayed in the upper part of the INGAMS command dialog) in the **Cmd** field next to the system name:

**A** Allows you to:



- Modify the operation mode of the automation manager from secondary to primary
- 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.

- B** Displays additional information about the automation manager or the automation agents. For an automation agent, the DISPSYS command is issued to show details about the agent and the environment it runs on.
- C** Displays a panel that prompts you for the name of the data set that contains the automation manager configuration file members. This is then used to update the definition data for the automation manager in real time and to reload the corresponding automation control 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 associated automation control files.
- D** You can issue this command for a primary automation manager to view or produce diagnostic information, such as recording actions performed during automation or displaying work item statistics. For more information, see “How to Use the Diagnostic Option of INGAMS” on page 237.

---

## Changing the Automation Agent Status

You may want to change the status of an application in the following circumstances:

- When you want SA z/OS to resume handling of an application after a manual operator shutdown or after any other event that causes an application to go to a non-automatable status (for example, STOPPED, CTLDDOWN, BROKEN, ACTIVE).
- When you want to prevent automation from restarting an application by setting its state to CTLDDOWN.
- When you want to inform SA z/OS that an application that has been shut down is going to be moved to another system.

System operations defines 24 different statuses for automation, described in “Automation Agent Statuses” on page 40. For details about these statuses, see the help panels or *IBM Tivoli System Automation for z/OS Customizing and Programming*. Figure 15 on page 96 and Figure 16 on page 96 indicate the relationships between the automation statuses. You can change the states illustrated here with the SETSTATE command (described in the *IBM Tivoli System Automation for z/OS Operator’s Commands*).

## Changing the Automation Agent Status

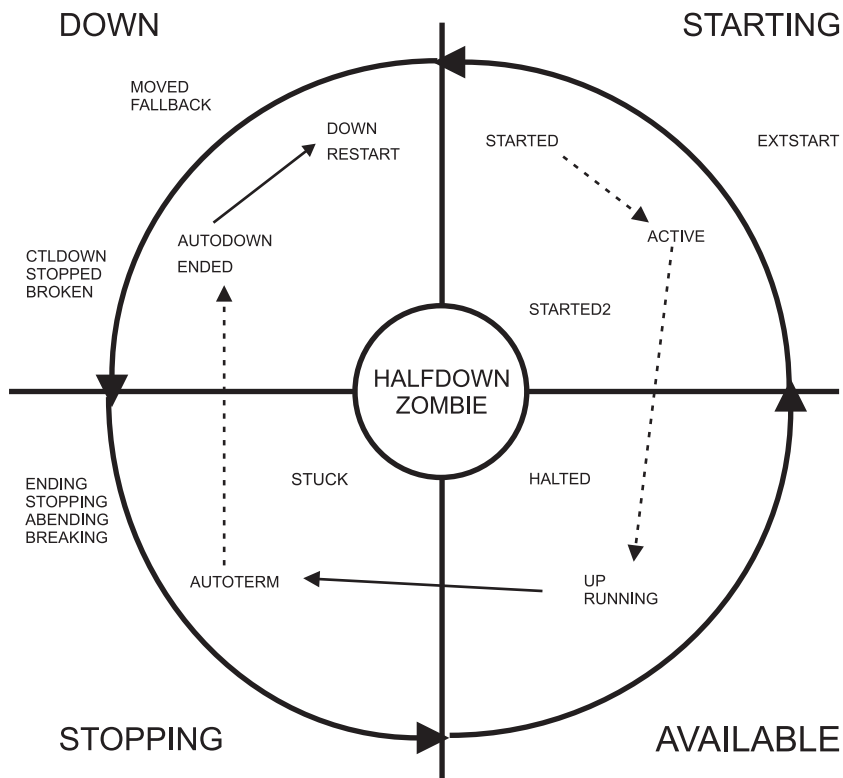


Figure 15. Transitions for a Normal Subsystem

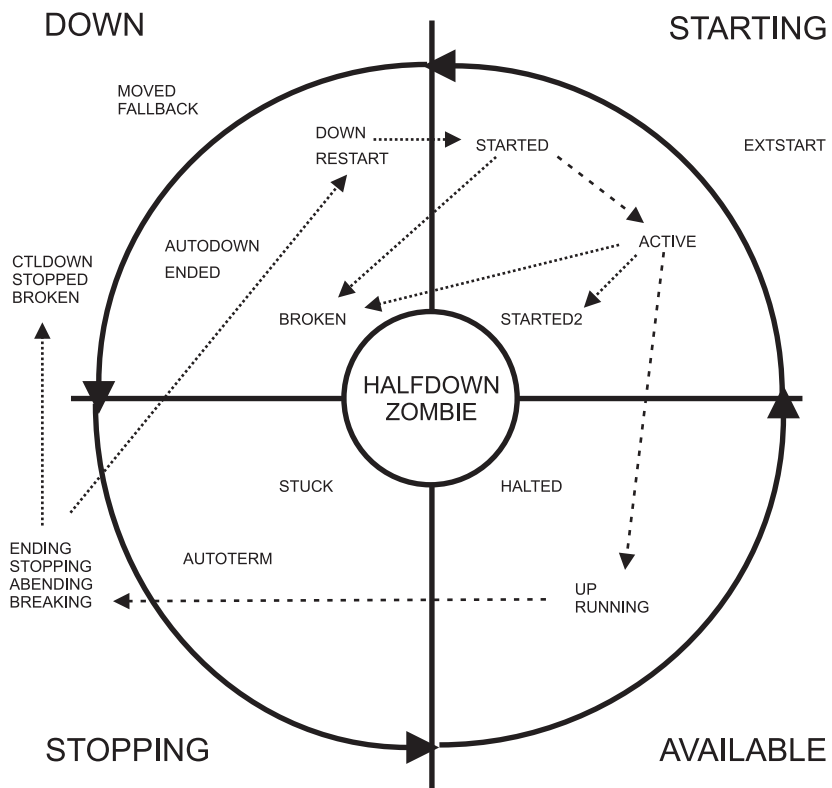


Figure 16. Transitions for an Abending Subsystem

To change the status of a subsystem, issue the SETSTATE command. The SETSTATE command is used to change the automation status of a subsystem or a group of subsystems (supporting and dependent resources). The status change you are able to make depends upon the subsystem's application monitor status. If the subsystem's application monitor status is ACTIVE, you may change its automation status to UP. If its application monitor status is INACTIVE, you can change it to RESTART or CTLDOWN. You may also be able to change it to MOVED or FALLBACK, depending upon its system associations.

With the SETSTATE command dialog, you can change the automation agent status of a subsystem or a group of subsystems by means of the **Scope** field.

**Note:** The **Override** field is valid only for systems that are automated with a previous release of SA z/OS.

---

## Monitoring Resources

With SA z/OS you can monitor various aspects of your resources, including:

- Messages, see "Monitoring Messages"
- Hardware, see "Monitoring Hardware" on page 100
- Health performance, see "Monitoring Health Performance" on page 102
- OMEGAMON sessions, see "Monitoring OMEGAMON Sessions" on page 103
- Processor status, see "Monitoring Processor Status" on page 107

## Monitoring Messages

Message suppression facilities at the target system determine the amount of message traffic sent from a target system sent to a console. Messages that do not normally require operator intervention are usually suppressed at the target system. Messages that do normally require operator intervention may also be suppressed at the target system as automation programming is developed to respond for you.

You can monitor messages at the SA z/OS workstation via several mechanisms and at several levels. You can view:

- No messages
- Messages that have been sent to a specific system console or operator console
- Messages that are related to the control of processor operations-managed resources
- SA z/OS messages (specified by message class)

### How to Display Outstanding WTOR Messages

WTOR (Write to Operator with Reply) messages are z/OS messages that require operator action. These messages can be viewed on the z/OS operator console.

### How to Display Messages for a Specific System or Operator Console

You can choose to receive all messages for a specific target system operator console using the Processor Operations Interested Operator List. These messages are displayed at your NetView console session.

Messages from the target console that are displayed include new message lines displayed on the Support Element console integration (CI) window, sent by z/OS, OS/390, VM, Linux for z/Series, VSE, or the Coupling Facility Control Code (CFCC). Multiline messages issued by z/OS look like several single-line messages

## Monitoring Messages

from the console. Similarly, if an z/OS message extends over two lines, it looks like multiple separate messages from the processor operations console.

When you are monitoring a CI, make sure it is enabled to receive regular messages. For z/OS, refer to the VARY CONSOLE command parameters ACTIVE, DEACTIVATE.

All processor operations messages from a target system begin with the processor operations identifier ISQ900I or ISQ901I plus the target system name and the console type. Messages prefixed by ISQ900I are not displayed but are used by the automation. User automation should key on the ISQ900I messages. Messages prefixed by ISQ901I are sent to interested operators and should not be used for user automation. Multiline messages appear as multiple messages.

**Joining or Leaving an Interested Operator List:** For target system operator console messages from CI use the ISQXMON command. For general processor operations resource control information messages use the ISQCMON command on a NetView console.

**Viewing the List of Interested Operators:** The processor operations status facility contains status panels showing the list of interested operators for a specific target system.

You cannot directly access the Processor Operations Interested Operator List panel with the ISQXDST command. You can access the Processor Operations Interested Operator List panel only from the following status panels:

- Target System Summary panel (Figure 20 on page 108).
- Target Hardware Summary panel. This is accessible only from other status panels.

The best way to access the Processor Operations Interested Operator List panel is to issue the following command to access the Target System Summary panel for a specific target system:

```
ISQXDST target_system_name
```

The resulting status panel shows the communication path status for the connection between the processor operations focal point and the target hardware (CPC) where the target system image is running (LPAR or Basic Mode). Place the cursor on the communication status line and press the PF7 key to see the associated Processor Operations Interested Operator List panel. If your operator identifier (the name you use to log on to NetView) is in this list, you should be receiving messages from that path. All other operators whose names are in the list also receive messages from that path.

You check the interested operator list for processor operations resource control messages (the ISQCMON list) by issuing the following NetView command:

```
LIST ASSIGN=GROUP
```

The interested operator list for processor operations is in group +TSCFMON.

### How to See SA z/OS Messages Specified by Message Class

Notification operators are defined during customization of SA z/OS using the SA z/OS customization dialog. A notification operator is a NetView console operator who is authorized to receive messages from SA z/OS. The classes of messages that a notification operator is authorized to receive are specified at the

time a notification operator is defined. See the tasks below to understand how to display information about a specific notification operator, how to add and delete notification operators, and how to change the messages that a notification operator receives.

Automation manager messages can be received automatically.

Also, you can receive messages from all systems within the sysplex.

### How to Get Notify Messages: INGNTFY

This section provides information about notification operators and those messages that notification operators should be notified of.

**Display Settings for a Notification Operator:** If you specify the INGNTFY command without a parameter, the main INGNTFY panel is displayed, showing a list of all the notification operators that have been defined. From this panel you can:

- Add (A) a notify operator using the settings of the selected operator
- Show or change settings (C) for the selected operator
- Delete (D) the notify operator
- Turn off (O) the sending of notify messages to the selected operator

For further information on the INGNTFY command refer to the *IBM Tivoli System Automation for z/OS Operator's Commands*.

**Query a Specific Notification Operator:** To display information about a specific notification operator, type INGNTFY on any command line and type query into the **Status/Action** field. Also, type the ID of the operator you want information about

**Turn On or Off all Notification Messages:** Use INGNTFY OFF on a NetView or operator console to temporarily turn off messages when you do not want to see them. Use INGNTFY ON on a NetView or operator console to turn the messages back on when you want to see them again.

**Note:** Each Notify Operator is issued a copy of the originating message. Consequently, if this message is being automated (it has an entry in the NetView automation table) this automation is replicated for each Notify Operator.

**Add a Notification Operator:** You can add more notification operators through the INGNTFY command dialog. Remember that this operator will only be able to receive messages until the next time the automation control file is loaded, when all changes made using the operator interface are lost.

To add a notification operator, enter INGNTFY on any command line.. Type add in the **Status/Action** field. Also, type the ID of the operator you want to add.

In the **Classes** field, type the classes of messages you want this operator to see. The default classes are 40 and 80.

If you want to hold certain types of messages on this operator's screen, move the cursor beside the desired Message type (Information, Eventual Action, Immediate Decision, System Wait, or Immediate Action), and type any character. For more information on message types, see *IBM Tivoli System Automation for z/OS Messages and Codes*.

## Monitoring Messages

**Change the Messages a Notification Operator Receives:** To change the messages you or another operator receives, you access the INGNTFY command dialog and you fill in the panel exactly as if you were adding a new operator. (You do not have to delete the definition first because the new definition overwrites the old.)

Because the new definition overwrites the old, you need to be careful when you are changing the messages an operator receives. For example, if the operator was originally assigned to receive classes 41 and 43, and you typed 44 in the Classes field, 44 would be the only class that the operator would now receive. If you want to add a class, you must also type all the classes that were previously assigned.

You can perform a query to find out what messages the operator currently has assigned. See “Query a Specific Notification Operator” on page 99 for more information.

**Delete a Notification Operator:** You can also use the INGNTFY command dialog to delete notification operators. Remember that if you use this panel to delete an operator, the deletion will be in effect only until the automation control file is reloaded.

To delete a notification operator, type `delete` in the **Status/Action** field. Also, type the ID of the operator you want to delete. This field can contain any valid NetView operator or the keyword `SYSOP`. If `SYSOP` is defined as a notification operator its function is to receive messages to the master console when no other operator is logged on, or to receive specific messages depending on the message classes assigned.

## Monitoring Hardware

You can monitor all hardware, from processors to devices, using processor operations and I/O operations.

You can also monitor hardware resources and you can issue system operations and processor operations commands from the NetView Management Console as described in Chapter 13, “Using the NetView Management Console for SA z/OS,” on page 179.

**Some General Information About Panels:** Pressing the Enter key toggles between a dynamically updated panel and a static panel. If any processor operations screen is updated frequently, it appears to flicker. You may need to place the panel into a static condition to select another function. The Updates field in the top right of the panel shows the current mode of the panel (Dynamic or Static).

### Processor Operations Status Information

Processor operations allows you to monitor target processors, target systems, and focal point communication path by using the ISQXDST status panels of your NetView operator console. These panels are available only on a NetView operator console attached to the SA z/OS processor operations focal point system.

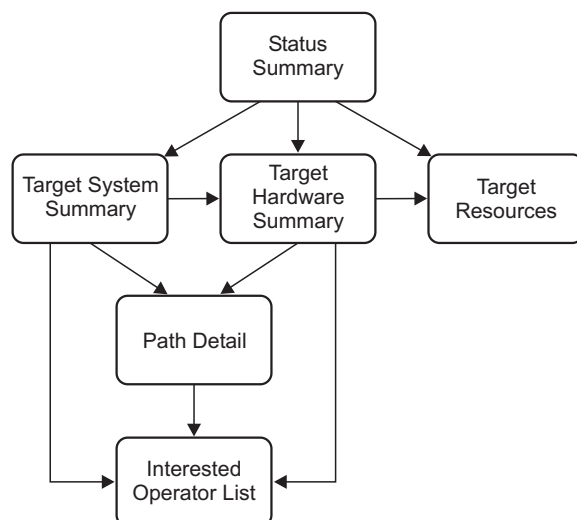


Figure 17. ISQXDST Command on NetView Console

## Displaying Processor Resource Information

To view target hardware summary of a processor, select PF9 from the Target System Summary panel (Figure 20 on page 108) or from the Status Summary panel (Figure 19 on page 107) on a NetView console.

```

ISQEHARP          SA z/OS Proc-Ops THW Summary  Updates: Dynamic
Target Hardware Name: YORAMA  Type: 9672      Mode: LPAR
                          Model: RX4        Serial Number: 51-0064523

Target Hardware Description: Web Services Backbone Bldg. 14

Initialized Target Systems:
    KEY6      CF61      CF62      KEY7

Communication Path Error Status:

Last Significant Message:

Enter=Static PF1=Help  PF3=Exit  PF6=Roll  PF7=Oper List  PF11=PATH Dtl's  PF12=Quit
  
```

Figure 18. Target Hardware Summary Panel for OCF-Based Processors

This panel defines the PF keys as follows:

- The PF7 key displays the Processor Operations Interested Operator List panel (the one accessed with the ISQXMON command).
- PF11 displays the Path Detail panel (Figure 21 on page 113). 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:

## Monitoring Hardware

### 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 zSeries 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:

#### **NVC or SNMP SESSION BUSY**

(red) Processor operations cannot presently communicate with the target support element, the problem appears recoverable.

#### **NVC or 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

## Monitoring Health Performance

You can monitor the health performance of your resources using monitor resources with the NCCF panels or the NMC.

### Using the NCCF Panels

Once you have loaded your configuration you can use the NCCF panels to check the status of your resources. You might want to start with INGLIST. If you scroll to the right, you will see the health status field.

Assume that you are monitoring CLIENT/APL with INGLIST and you find a compound status of DEGRADED and a health status of WARNING. In this case you can enter the line command M in front of CLIENT/APL. This will take you to the DISPMTR panel showing all MTRs that are connected to CLIENT/APL via HasMonitor relationships. To find out the reason for the health status WARNING you can scroll to the right to find more information or enter the line command D in front of the MTR that has a health status of WARNING, which takes you to the DISPMTR Details panel.

### Using the NMC

To use the MTRs in your NMC you can define a view that will contain one or more MTRs by entering the appropriate filter criteria. If you want to see your APL together with the connected MTRs you must work out a naming convention that



allows you to specify wildcards. Another (less flexible) possibility is to name every resource (APL and MTR) explicitly in the view.

If you have a view with both APLs and connected MTRs you will see the HasMonitor relationships as lines between them. You can right-click on the icons and issue the associated commands such as DISPMTR or INGMON.

### Using the Tivoli Enterprise Portal

The Tivoli Enterprise Portal (TEP) is a Java-based interface to the data monitoring and management resources of the IBM Tivoli Monitoring infrastructure. It displays information about the status of automation on z/OS systems and z/OS sysplexes that has been collected by the SA z/OS monitoring agent.

The SA z/OS monitoring agent uses the TEP to provide a view of your enterprise from which you can drill down to examine details about each system being monitored. The TEP application window consists of a Navigator that shows all the systems in your enterprise where monitoring agents are installed, and a workspace that includes table and chart views of system and application conditions. Each workspace is designed to help monitor a specific component of your system. A table of attributes is provided for each workspace.

Each attribute represents a particular kind of data about system resources being monitored and reported. Attributes can also be used to define situations to test for specific conditions. When the conditions for a situation are met, situation event indicators are displayed in the Navigator.

The SA z/OS monitoring agent provides predefined workspaces, which you can access from the Navigator in the Tivoli Enterprise Portal. The product workspaces enable you to monitor the status of automation of your z/OS systems and sysplex resources. You access these workspaces from nodes at two levels of the Navigator tree: the SA z/OS level and the SA z/OS Agent level.

Attributes are characteristics or properties of the objects monitored by the SA z/OS monitoring agent. Related attributes are organized into attribute groups (also called attribute tables). The attributes are used to define the queries that collect the information displayed in tables and charts in the SA z/OS monitoring agent workspaces and to create situations that trigger alerts in response to specified conditions.

The SA z/OS monitoring agent provides a set of predefined situations that monitor the status and health of automation resources within the SA z/OS subplex. These situations check for specified conditions and can trigger Critical, Warning, or Informational situation event indicators (also called alerts) in the Navigator. When a situation triggers an alert, you can investigate the situation event by opening its workspace.

For more details see *IBM Tivoli System Automation for z/OS Monitoring Agent Configuration and User's Guide*.

## Monitoring OMEGAMON Sessions

SA z/OS allows you to use Monitor Resources to connect to classic OMEGAMON monitors to send commands and receive responses.

You can use the INGSESS command either from the command line or with the fullscreen command dialog to:

## Monitoring OMEGAMON Sessions

- Display OMEGAMON sessions
- Start sessions manually to test connection and authorization
- Stop sessions to perform maintenance
- Show additional session attributes, for example, logon data, timeout, statistics, etc.

### Displaying OMEGAMON Sessions

When you enter the INGSESS command at the NetView command line the INGSESS command dialog is displayed.

The panel displays the following session 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 one of the following:

INACTIVE	The session does not exist
ACTIVE	The session exists and is ready to receive requests
AUTHFAIL	The session could not be created because the logon to OMEGAMON failed; all requests will be denied
SESSFAIL	SA z/OS received unexpected data or some NetView communication error occurred; all requests will be denied
MAINT	The session was terminated for maintenance purposes; all requests will be denied
- 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.

### Starting OMEGAMON Sessions

You can start sessions manually to test connection and authorization using either:

- The command line, for example  
INGSESS ims742cr req=start outmode=line
- The command dialog, as follows:

CMD: B Start session C Stop session D Details

CMD	Session	System	Type	Status	Appl-id	User id	SessOper
—	CICSKY41	OMIICICS	AOC9	ACTIVE	IPSP0C0	SAOM	AOFSES01
—	DB2SGG4	OMIIDB2	AOC9	INACTIVE	IPSPD2C	SAOM	AOFSES02
—	DB2SG14	OMIIDB2	AOC9	MAINT	IPSPD2C	SAOM	AOFSES03
<b>B</b>	IMS742CR	OMIIMS	AOC9	INACTIVE	IPSP0I0	SAOM	AOFSES01
—	OMSY4MVS	OMIIMVS	AOC9	AUTHFAIL	IPSPM2RC	SAOM	AOFSES02

This 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.

If the session could not be started, the status will become either AUTHFAIL or SESSFAIL.

### Stopping OMEGAMON Sessions

You can stop sessions to perform maintenance using either:

- The command line, for example:  
INGSESS cicsky41 req=stop outmode=line
- The command dialog, as follows:

CMD: B Start session C Stop session D Details

CMD	Session	System	Type	Status	Appl-id	User id	SessOper
<b>C</b>	CICSKY41	OMIICICS	AOC9	ACTIVE	IPSP0C0	SAOM	AOFSES01
—	DB2SGG4	OMIIDB2	AOC9	INACTIVE	IPSPD2C	SAOM	AOFSES02
—	DB2SG14	OMIIDB2	AOC9	MAINT	IPSPD2C	SAOM	AOFSES03
—	IMS742CR	OMIIMS	AOC9	INACTIVE	IPSP0I0	SAOM	AOFSES01
—	OMSY4MVS	OMIIMVS	AOC9	AUTHFAIL	IPSPM2RC	SAOM	AOFSES02

This destroys the session with the appropriate OMEGAMON. The session status will be changed to MAINT. The session is disabled to prevent it from being created implicitly again.

After you have fixed the problem you can restart the session.

### Recovering from Session Failure

A session may not be created because either the logon to OMEGAMON failed, or SA z/OS received unexpected data or some NetView communication error occurred. In these cases all requests to the session will be denied.

To recover a failed session:

1. Put session in maintenance mode, for example:
  - From the command line:  
INGSESS omsy4mvs req=stop outmode=line
  - Using the INGSESS command dialog:

## Monitoring OMEGAMON Sessions

CMD: B Start session C Stop session D Details

CMD	Session	System	Type	Status	Appl-id	User id	SessOper
—	CICSKY41	OMIICICS	AOC9	ACTIVE	IPSP0C0	SAOM	AOFSES01
—	DB2SGG4	OMIIDB2	AOC9	INACTIVE	IPSPD2C	SAOM	AOFSES02
—	DB2SG14	OMIIDB2	AOC9	MAINT	IPSPD2C	SAOM	AOFSES03
—	IMS742CR	OMIIMS	AOC9	INACTIVE	IPSP0I0	SAOM	AOFSES01
<u>C</u>	OMSY4MVS	OMIIMVS	AOC9	<b>AUTHFAIL</b>	IPSPM2RC	SAOM	AOFSES02

- Analyze and fix the problem
- Restart the session, for example:

- From the command line:

```
INGSESS omsy4mvs req=start outmode=line
```

- Using the INGSESS command dialog:

CMD: B Start session C Stop session D Details

CMD	Session	System	Type	Status	Appl-id	User id	SessOper
—	CICSKY41	OMIICICS	AOC9	ACTIVE	IPSP0C0	SAOM	AOFSES01
—	DB2SGG4	OMIIDB2	AOC9	INACTIVE	IPSPD2C	SAOM	AOFSES02
—	DB2SG14	OMIIDB2	AOC9	MAINT	IPSPD2C	SAOM	AOFSES03
—	IMS742CR	OMIIMS	AOC9	INACTIVE	IPSP0I0	SAOM	AOFSES01
<u>B</u>	OMSY4MVS	OMIIMVS	AOC9	<b>MAINT</b>	IPSPM2RC	SAOM	AOFSES02

### Displaying Additional Information

You can display additional session attributes (for example, logon data, timeout, statistics, etc.) using either:

- The command line, for example:

```
INGSESS omsy4mvs req=detail outmode=line
```

- The command dialog by entering the action code D for a session, as follows:

CMD: B Start session C Stop session D Details

CMD	Session	System	Type	Status	Appl-id	User id	SessOper
—	CICSKY41	OMIICICS	AOC9	ACTIVE	IPSP0C0	SAOM	AOFSES01
—	DB2SGG4	OMIIDB2	AOC9	INACTIVE	IPSPD2C	SAOM	AOFSES02
—	DB2SG14	OMIIDB2	AOC9	MAINT	IPSPD2C	SAOM	AOFSES03
—	IMS742CR	OMIIMS	AOC9	INACTIVE	IPSP0I0	SAOM	AOFSES01
<u>D</u>	OMSY4MVS	OMIIMVS	AOC9	AUTHFAIL	IPSPM2RC	SAOM	AOFSES02

This produces output where the additional information includes:

- The **Description** field shows descriptive information for this session as specified in the automation policy
- 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 **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.

## Monitoring Processor Status

Processor status can be monitored by viewing the status of a system icon on the NMC workstation or processor status can be seen in more detail using the processor operations 3270-type panels. On the NMC workstation, the status of a system icon represents the aggregation of status of the target's operating system software and its underlying processor hardware. You can see at a glance whether the systems that you are responsible for have satisfactory status.

Refer to Chapter 13, "Using the NetView Management Console for SA z/OS," on page 179 for information on how to enter commands against an unsatisfactory processor operations object.

This section contains the following subsections that describe certain aspects of monitoring processor status:

- "View Status Summary of all Target Systems"
- "View One Specific Target System" on page 108

### View Status Summary of all Target Systems

On a 3270 console, issue `isqxdst` (with no parameters) on a NetView console. The Status Summary panel is displayed.

```
ISQESUM SA z/OS Proc-Ops Target Status Summary      Updates: Dynamic 1
Control file in use: SATEST.SYSTEM.V31.IPSFO.ISQCNTL NMC Bridge: INACTIVE
I isqxiii C isqxc1s 0 isqxopt  A,B view netlogs E events Debug Mode: OFF

Cmd   Target System   Status 3 Focal Points - Primary: IPSFO Backup: IPUFM

      CFA          CLOSED
      CFB          CLOSED
      CFF          CLOSED
      CF1          UNKNOWN
      CF61         INITIALIZED
      CF62         INITIALIZED
      KEYA         STAGE-1 ACTIVATE COMPLETE
      KEYB         IPL COMPLETE
      KEYC         IPL COMPLETE
      LNXOLI1      NOT ACTIVE
      TEL01FE     LOAD FAILED

Enter=Static PF1=Help PF3=Exit PF4=Tgt Sys Summary PF5=Debug On/Off 2
PF6=Roll    PF7=Up    PF8=Down PF9=Tgt HW Summary PF11=PATH Details
```

Figure 19. ProcOps Target Status Summary Panel

## Monitoring Processor Status

- 1** This indicates whether updates are static or dynamic (it changes automatically whenever the status changes). You can toggle to a static state with the Enter key (for example, when you want to enter a choice).
- 2** The PF keys allow you to invoke other status panels.
- 3** The status that you see for a target system represents the most severe condition that exists for that system. These values are extracted from the *status*, *target hardware*, *attention*, and *console summary* status fields of the Target System Summary panel (see Figure 20).

### View One Specific Target System

On a 3270 console, use ISQXDST followed by the name of the target system on a NetView console. The Target System Summary panel is displayed.

```
ISQETARG          SA z/OS Proc-Ops Target System Summary  Updates: Dynamic

Target System
Name              : KEY3
Description       :
O.S.              : MVS
Load Profile      :
Status            : UNKNOWN
Attention         :

Target Hardware
Name              : FREEWAY      Mode : LPAR
Attention         :
Path Status       : DORMANT

Target Hardware LPAR
Name              : KEY3          Mode : ESA
Image Profile     : KEY3

Last Significant Message:

Enter=Static  PF1=Help  PF3=Exit  PF6=Roll  PF7=Oper List
PF9=Target Hardware  PF11=Path Detail  PF12=Quit
```

Figure 20. Target System Summary Panel

This panel defines the PF keys as follows:

- 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 (Figure 18 on page 101). 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, LOAD FAILED, or IPL FAILED to IPL COMPLETE 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 its status becomes INITIALIZED, then you change the value of *tstat* to IPL COMPLETE. You may also want to change the status manually if the cause of the LOAD FAILED 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 IPL COMPLETE to IPL FAILED. Processor operations sets a status of IPL COMPLETE 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.

## Monitoring Processor Status

### 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

(Yellow) SNMP Path Only: Target system attention status indicating an alert has been detected.

**DCCF** (Red) z/OS only: Processor operations has detected a DCCF message.

### DISABLED WAIT

(Red) Processor operations has detected a Disabled Wait condition.

### ENV ALERT

(Yellow) SNMP Path Only: Target system attention status indicating an environment alert has been detected.

### SERIOUS ALERT

(Red) SNMP Path Only: Target system attention status indicating a serious alert has been detected.

### Target Hardware

The name of the target hardware that this target system is running on.

**O.S.** MVS, VM, VSE, LINUX, or CF

**Mode** LPAR, ESA, 370

### LPAR name

The name of the logical partition that this target system is running in.

### Connection Path Information

Identifies the connection type and current status of the connection between the processor operations focal point and the CPC support element or HMC.

**Type** NVC (SNA-based NetView connection), SNMP (IP-based Simple Network Management Protocol connection).

**Status** (Updated dynamically) The current value of the processor operations internal variable *scstat*.

## Monitoring Processor Status

**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 Figure 21 on page 113).

### **ACTIVE**

(Green) The path is ready for communications.

### **NVC SESSION BUSY**

(Red) Processor operations cannot presently communicate with the target Support Element, the problem appears to be recoverable.

### **NVC SESSION PROBLEM**

(Red) Processor operations cannot communicate with the target Support Element.

### **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.

### **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.

## **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 21 on page 113.

ISQEPNVC	SA z/OS PATH Detail	Updates: Dynamic	
Target Hardware Name: FREEWAY		Connection Type: SNMP	
Network ID	: DEIBMD1	LU Name	: X7F1E30A
Support Element IP Address or Hostname: <b>9.164.172.200</b>			
Hardware Management Console IP Address or Hostname: <b>SAHMC</b>			
Connection Authorization: <b>POKLICKO</b>		Lock Holder	: <b>IGORDON</b>
PATH Status : <b>ACTIVE</b>			
Poll Frequency : 30			
Poll Retries : 2			
Last Significant Message:			
Enter=Static	PF1=Help	PF3=Exit	PF6=Roll
PF7=Oper List	PF12=Quit		

Figure 21. Path Detail Panel

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

## Monitoring Processor Status

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.

### 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.

### Path 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.

### 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.

## How to Monitor Alerts on a NetView Console

Alerts are high priority events that require operator action.

Issue NPDA from the command line of your NetView operator's console. The NPDA (Network Problem Determination Application) menu is displayed. You can request a dynamic display of alerts. From that display you can select an individual alert and see event detail and recommended action. For more details on the use of the NPDA panels, refer to *Tivoli NetView for z/OS User's Guide*.

---

## Monitoring with the Status Display Facility

The Status Display Facility (SDF) displays the status of various resources of a z/OS system, and of other z/OS systems that specify this system as an automation focal point. The resources that can be displayed by SDF include:

- Applications and subsystems
- WTORs
- Gateways
- Application groups

**Note:** A *gateway* is a combination of a NetView-NetView task session and two automation operators (one on each of two systems) that allows communication of messages, commands, and responses between the two systems.

SDF also shows spool usage problems from z/OS subcomponents. SDF consists of a hierarchy of dynamically updated panels showing color-coded status conditions. SDF is set up during the customization of SA z/OS.

Use the status display facility (SDF) to monitor the status of application software, WTORs and gateways on your systems from a NetView or operator console.

All the resources you need to monitor are set up by hierarchical arrangement during customization. At this time, the system programmer:

1. Defines the SDF hierarchy.  
The SDF hierarchy indicates which automated resources are affected by status changes in other automated resources (see “How SDF Works” on page 118). The SDF hierarchy can be different from that specified in the automation control file for system startup and shutdown. Use whatever makes sense for your system.  
If SDF displays more than one system, programmers create an SDF hierarchy for each system.
2. Creates the panels that show the systems and subsystems. These panels can also be customized to display gateway sessions and WTORs.
3. Determines the status colors and other screen defaults for the enterprise or takes the defaults for these attributes.

For information on setting up SDF, see *IBM Tivoli System Automation for z/OS Programmer's Reference* and *IBM Tivoli System Automation for z/OS User's Guide*. Because SDF can be customized, your panels may not look like those shown in this topic.

### SDF Focal Point Monitoring

At first you need to construct an SDF screen that contains all systems within your sysplex. This will give you an *at a glance* panel that will be dynamically updated with the overall status of each system in your sysplex. By selecting an individual system, you can view a more detailed display of the state of the resources on that system.

On the focal point, you can issue a command that specifies a target system. The command interacts with the target system (via XCF or RMTCMD) and gives you a full screen display as if you were logged on to the target system.

This means that from SDF you can issue DISPINFO on a target subsystem by simply pressing a key. This allows you to see detailed information about a subsystem that has a problem.

Other commands that support this kind of usage include:

- DISPERRS
- DISPFLGS
- DISPSTAT
- DISPTREE
- INGINFO

## Monitoring with the Status Display Facility

- INGREQ
- INGTHRES
- INGVOTE
- SETSTATE

All of these commands can be used from the focal point to interrogate the target system without having to log on to it.

### Note

If the target system is not in the same sysplex as your focal point system, RMTCMD will be used for intersystem communication. This means that there is a dependency upon VTAM being active on both the focal point and the target system for communication to be possible.

If the target system is in the same sysplex, communication occur via XCF and you do not need to have VTAM active on the target system to talk to it.

## Interpreting Colors in SDF and DISPSTAT

In SDF and on the DISPSTAT command dialog, subsystems appear in different colors that indicate their status. In SDF, the condition of WTORs and gateways is also indicated by color.

### Subsystem Colors and Priorities

The color of a subsystem on a DISPSTAT command dialog panel and the color and highlighting of a subsystem in SDF indicate the current status of a subsystem (the possible status conditions are defined in "Automation Agent Statuses" on page 40). Table 8 shows the default color (and highlighting) assigned to each status by SA z/OS. White is also used by default to identify status components without a tree structure. These colors may have been changed for your SDF display. Blinking as a means of highlighting is not used by every type of display, in particular when you are using 3270 terminal emulation on a PC. You may see another type of highlighting, for example, black text on a white background.

Table 8. Table of Default Subsystem Status Colors and Priorities

Color	Highlight	Status	Priority
Red	Reverse	BROKEN	120
	Blinking	BREAKING	130
	Normal	STOPPED	150
Pink	Normal	HALFDOWN	220
	Blinking	STARTED2	230
	Underline	STUCK	240
	Reverse	ZOMBIE	250
White	Reverse	ABENDING	320
	Underline	HALTED	330
Yellow	Reverse	STOPPING	420
	Normal	AUTOTERM	430
	Underline	ENDING	440

## Monitoring with the Status Display Facility

Table 8. Table of Default Subsystem Status Colors and Priorities (continued)

Color	Highlight	Status	Priority
Turquoise	Blinking	RUNNING	520
	Reverse	ACTIVE	530
	Underline	STARTED	540
	Normal	RESTART	550
	Blinking	EXTSTART	550
Green	Normal	UP	640
	Underline	ENDED	650
Blue	Underline	DOWN	730
	Normal	AUTODOWN	740
	Blinking	CTLDOWN	750
	Reverse	MOVED	760
	Normal	FALLBACK	770

Each automation status is assigned a priority number that SDF uses to determine what color to display if there are multiple status conditions present in a system. The status conditions are listed in the preceding table in order of descending priority. In other words, the first status listed is the status with the highest priority, and the one whose color will dominate. All of the colors and priorities listed above are defaults, and can be modified when a customer defines SA z/OS policy.

**Note:** Panels can be modified according to your needs.

### Gateway Colors

Besides displaying the status of subsystems, SDF uses colors to show the status of gateway sessions. The following list explains what the gateway colors indicate:

Color	Meaning
Reverse Turquoise	Gateway is active
Red	Gateway is inactive

### Spool Colors

SDF uses the following colors to show the status of spools.

Color	Meaning
Green	Spool is OK
Yellow	Spool is short
Red	Spool is full

### WTOR Colors

SDF also uses colors to show what type of WTOR each displayed WTOR is, based on classifications that have been specified using the customization dialog. See *IBM Tivoli System Automation for z/OS Defining Automation Policy* for more details.

Color	Meaning
Green	WTOR is normal

## Monitoring with the Status Display Facility

Color	Meaning
Yellow	WTOR is unusual
Red	WTOR is important

### Using SDF

You use SDF to monitor the status of subsystems, WTORs, and gateways on your system and other systems. To monitor status, you watch the highest level SDF panel (also known as the root panel or system summary panel). The following example shows a panel with the systems of a sysplex. This panel displays for example, systems, subsystems, WTORs or gateways within this sysplex.

```
SYSTEM                SA z/OS - SUPPORT SYSTEMS

  System  Subsystems  WTORs   Gateways  Products  System
  KEY1   HUGC2APL0  NETATST1  IPSFP0    C I D O   P V M B T U
  KEY2    IMS723C4    IMS721M1  IPSFN0    C I D O   P V M B T U
  KEY3    IMS733C4    NETATST3  IPSF00    C I D O   P V M B T U
  KEY4                                C I D O   P V M B T U
  XXXX                                C I D O   P V M B T U

                                05/12/04 08:45

===>
1=HELP 2=DETAIL 3=RETURN 6=ROLL 8=NEXT SCR 10=LEFT 11=RIGHT 12=TOP
```

Figure 22. Example SDF System Summary Panel

On this panel, each system is displayed in a color that reflects the highest priority status of the resources in that system. If a resource in a system changes status, the system changes color to reflect the new status. By watching this panel you will see status changes in any system.

Imagine that **KEY1 HUGC2APL0** is displayed in red reverse on your screen, which means that the subsystem HUGC2APL0 on system KEY1 is in status BROKEN. To view more information on this subsystem, move the cursor to HUGC2APL0 (or KEY1) and press PF2 to get a panel as shown in “How to Check Resource Status Conditions” on page 119).

### How SDF Works

SDF uses the hierarchy established during SDF customization to propagate colors up to the highest level panel. The following figure shows a graphic representation of an example SDF hierarchy for KEY1:



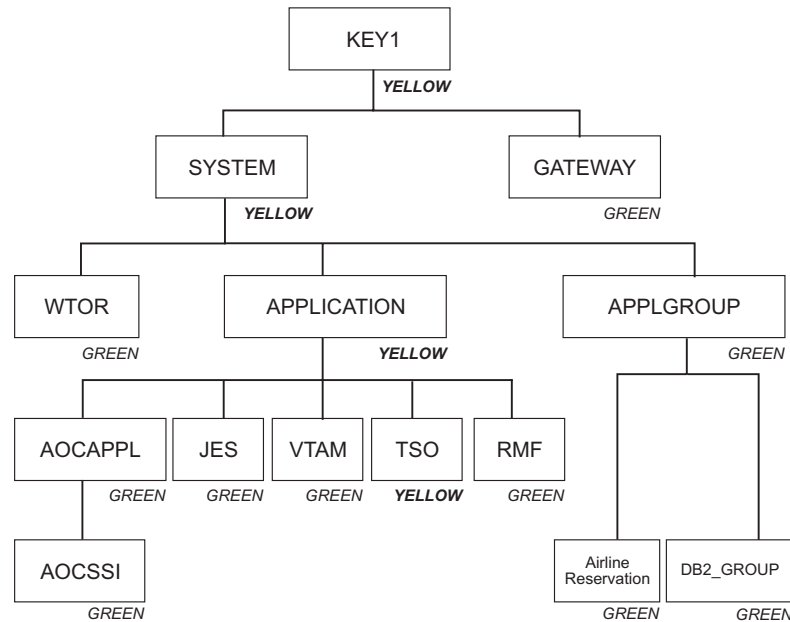


Figure 23. Example SDF Hierarchy for KEY1

Imagine that TSO appears in yellow on your screen because it is in AUTOTERM status. This color is reflected up the hierarchy to APPLICATION, SYSTEM, and KEY1, because AUTOTERM has a higher priority than the status UP (indicated by the color green). If various resources in a system have different status conditions, the one with the highest priority status controls the color of that system. This allows you to resolve issues concerning the highest priority status first, then deal with the lower priority status conditions.

### How to Access SDF

To access SDF, type `sdf` on a NetView command line or type 3 (SDF) on the command line of the SA z/OS System Operations menu (see “Using Command Dialogs” on page 68).

You see the highest level (or root) panel of SDF. Figure 22 on page 118 shows an example of the highest level panel for the SA z/OS supported systems. On this panel, **KEY1 HUGC2APL0** is highlighted to indicate that it appears in red on your screen, which means that subsystem HUGC2APL0 on KEY1 is BROKEN. All other systems may appear in green, which means UP.

### How to Check Resource Status Conditions

Normally when you use SDF to monitor resources, you watch the highest level panel until you see a system change color. This color change indicates a change in status in one of the resources in the system. To determine which resource, and to see more information on the status change:

1. Move the cursor to the system (or subsystem) that shows a status change.
2. Press PF2. This takes you to the Detail Status Display panel for the resource that caused the system to change color.

If more than one resource changes status at the same time, SDF shows the information for the system with the highest priority status first. See “Subsystem Colors and Priorities” on page 116 for information on status priorities. You can press PF8 to page through the Detail Status Display panels for other resources on KEY1.

## Monitoring with the Status Display Facility

For example, in the Best Goods Company, you move the cursor to MVSESA3 (the system in red). When you press PF2, you see the Detail Status Display panel for DB2, which is the resource with the highest priority, because it is BROKEN.

```
----- DETAIL STATUS DISPLAY -----                                1 OF 242

COMPONENT: HUGC2APL0          SYSTEM   : KEY1
COLOR   : RED                 PRIORITY :    120
DATE    : 11/08/06           TIME     : 19:21:26
REPORTER : GAT1PUFM          NODE     : IPUFM

REFERENCE VALUE: HUGC2APL0

AOF571I 19:21:26 : KEY1.HUGC2APL0 SUBSYSTEM STATUS FOR JOB HUGC2A0 IS
BROKEN

===>
1=HELP 3=RETURN 4=DELETE 6=ROLL 7=UP 8=DOWN 11=BOTTOM 12=TOP
```

Figure 24. Example SDF Detail Status Display Panel for HUGC2APL0

This panel also shows the automation or human operator that sent the notification and gives the priority number that corresponds to the automation status. Priority numbers determine which color the resource is displayed in and the order of the Detail Status Display panels (the lower the number the higher the priority).

In this example, the panel is one of four Detail Status Display panels (as indicated at the top right corner of the panel), one for each of the resources under MVSESA3 that has detail information. Press PF8 to see the Detail Status Displays for the other resources. PF8 pages through the displays in order of descending status priority.

To return to a particular panel, you can press PF7, which pages through the displays in order of ascending priority, or you can press PF12, which takes you back to the first Detail Status Display in the stack, the one with the highest priority status.

**PF3** Returns to the system panel.

**PF4** Deletes the panel you are currently on. (The record is physically deleted.)

**Note:** You should normally not use this PF key.

**PF11** Moves to the Detail Status Display panel for the resource with the lowest priority, the last panel in the Detail Status Display stack.

**PF12** Moves to the Detail Status Display panel for the resource with the highest priority, the first panel in the Detail Status Display stack.

**Note:** PF keys can be customized. How to achieve this is described in the *IBM Tivoli System Automation for z/OS Programmer's Reference*.

### How to View the Resource Hierarchy

To look at the resource hierarchy, select a resource with the cursor and press PF8 to go down a level or PF7 to go up a level.

```

Resource Status for System Boulder1

System   Subsystem   WTORs   Gateways   Spool   MVS Comps
FOC1     SDSF        NetView
FOC2     AOCAPPL3
TGT1     AOCSSI
TGT3     JES
SYS1     VTAM
SYS2     TSO
SYS4     RMF
    
```

### How to Monitor WTORs with SDF

To view a WTOR through SDF (if SDF is set up to display WTORs) you can enter `sdf` on a NetView console to use the status display facility. Position the cursor on a job under WTORs (for example, `INGESO`, in the following panel extract) and press PF2 for more details. A Detail Status Display panel will be shown, containing text and information about an outstanding WTOR message.

```

Resource Status for System Boulder1

System      Subsystem   WTORs      Gateways   Spool   MVS Comps
SYS_TSCF2   INGESO     INGESO
    
```

### How to Monitor Application Groups with SDF

To view application groups through SDF (if SDF is set up to display application groups, see also “How SDF Works” on page 118) you can enter `sdf` on a NetView console to use the status display facility. By using PF8 (ZOOM), navigate to the SUMMARY STATUS panel for the system that you want to monitor application groups for. In our customized example from Figure 25, we selected system `KEY3`.

```

KEY3

                SUMMARY-STATUS
                KEY3

                >WTOR      >GATEWAY
                >MESSAGES >MVSCOMP

                !-----!

                >MVS

                >SYSTEM

                >APPLS    >GROUPS

                >TEST

                                                    06/05/04 14:02

===>
1=HELP 2=DETAIL 3=RETURN 6=ROLL 7=UP      8=ZOOM 12=TOP
    
```

Figure 25. SDF Summary Status Panel

Position the cursor under `GROUPS` and press PF8. In our example, we will see a customized panel containing information about application groups on `KEY3`.

## Monitoring with the Status Display Facility

```
KEY3                      KEY3:  APPLGROUP-STATUS

QFETCH

SYBNET_G

SYBSYS_G

                                06/05/04 14:13
====>                          13=EXPLAIN      17=START 18=STOP
1=HELP 2=DETAIL 3=RETURN 6=ROLL 12=TOP  20=APPLGR 21=ACF 22=ASF 23=LIST 24=INFO
```

Positioning the cursor on an application group of your choice and pressing PF2 (DETAIL) will display the SDF Detail Status Display panel for the selected application group, in our example, QFETCH (see Figure 26).

```
----- DETAIL STATUS DISPLAY -----                                1 OF 3

COMPONENT: QFETCH          SYSTEM   : KEY3
COLOR   : PINK            PRIORITY : 490
DATE    : 11/09/06        TIME    : 20:16:32
REPORTER : AUTMSG         NODE    : IPSNO
REFERENCE VALUE: QFETCH

QFETCH GROUP STATUS FOR SYSTEM APPLGROUP IS CTLDOWN

====>
1=HELP 3=RETURN 6=ROLL 7=UP 8=DOWN 10=QUIT 11=BOTTOM 12=TOP
```

Figure 26. SDF Detail Status Display Panel for an Application Group

## Monitoring and Controlling a Sysplex

SA z/OS provides a command interface to monitor and operate sysplexes. It helps operators to manage critical sysplex resources like coupling facilities and CF structures, couple data sets, and more. You can benefit from reduced sysplex operations complexity leading to higher sysplex availability and greater productivity.

The following section describes the panels of the system operations commands that support you in working with sysplex specific automation tasks. For details about the syntax and parameters of these commands, refer to *IBM Tivoli System Automation for z/OS Operator's Commands*.

### How to Work with INGPLEX

SA z/OS provides the command INGPLEX to monitor and control the couple data sets and coupling facilities of your sysplex.

If you enter INGPLEX in the command line without any parameters, the INGPLEX Selection Panel is displayed.

To launch other command dialogs that show sysplex-related information, specify the number or the function and press Enter.

The subcommands support the following functions:

<b>SYStem</b>	Displays the target sysplex name, its GRS mode and its systems. For each system, details about the system name and status are shown.
<b>CONsole</b>	Displays details about the target sysplex. For example, the master console name, WTO and WTOR buffer utilization, and the number of queued messages.
<b>CF</b>	Displays the coupling facilities in the sysplex. For each coupling facility the structures and paths can be displayed, and the coupling facility can be drained and enabled.
<b>CDS</b>	Displays the couple data sets in the sysplex. For each couple data set you can obtain detailed information, and you can switch the couple data sets or allocate new alternate couple data sets.
<b>IPL</b>	Collects IPL information.
<b>DUMP</b>	Displays the DUMP submenu.
<b>INGSTR</b>	Displays the INGSTR panel. See “How to Work with INGSTR” on page 138

These functions are described in more detail in the following sections.

### INGPLEX SYStem

INGPLEX SYStem displays the target sysplex name, its GRS mode and its member systems.

- 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** column allows you to specify one of the command codes to display more information. 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 lets 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.

## INGPLEX SYStem

- The SSUM WEIGHT field shows the SSUM weight specified in the current SFM policy. This weight is used in sysplex configurations after a signalling connectivity failure.

Use the PF11 key to view WLM-related information to the right.

- The **Timestamp** field shows the last time stamp when capacity data was queried from WLM.
- The **SUs Total** field shows the number of free SUs in the last 10 minutes.
- 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.

To display more detailed information, enter the following command codes:

- |          |   |
|----------|---|
| <b>C</b> | Displays the online or offline status of one or more processors and any vector facilities, or ICRFs attached to those processors  |
| <b>E</b> | Displays the timer synchronization mode and ETR ports   |
| <b>I</b> | Displays IPL information  |
| <b>O</b> | Displays IOS-related configuration information  |
| <b>S</b> | Displays the number of megabytes of central and expanded storage assigned and available to the system   |
| <b>D</b> | 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 |
| <b>T</b> | Displays detailed signalling path information for all coupling facility structures  |

## INGPLEX CONsole

INGPLEX CONsole 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)
- 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 of messages awaiting replies.
- The **Immediate Action** field shows a decimal 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 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 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 of outstanding mount requests.
- The **Operator Requests** field shows a decimal number of outstanding requests for operator intervention.
- The **CMD** column lets you specify the command codes shown on the panel. Type the appropriate letter next to the resource name, and press Enter.
- The **Console** column 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 indicated when 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 will be indicated only when 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 will be indicated only when 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** column shows which commands can be entered from this console. The following values can occur:

**ALL** Any INFO SYS, IQ or CONS command can be entered from this console.

**CONS** INFO commands and any commands from the console command group can be entered from this console.

**INFO** Any command from the informational command group can be entered from this console.

**IO** INFO commands and any commands from the I/O Control command group can be entered from this console.

**MASTER** The specified console is authorized to enter any operator command.

**NONE** This console has no command authority.

**SYS** INFO commands and any commands from the system control command group can be entered from this console.

- The **NBUF** column shows the number of WTO message buffers currently queued to this console. If *mnn* is greater than 9999, asterisks (\*\*\*\*) appear in this field.

- The **UD** column shows whether this console is receiving messages with the UD attribute.
- The **Device** column shows the name of the console as specified in the CONSOLxx parmlib member.
- The **System** column shows the system name of the active console.
- The **ALTGRP** column shows the alternate group defined for this console.
- The **MSCOPE** column 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 CF

With INGPLEX CF you can:

- Drain or enable a selected coupling facility
- Display the paths and structures to a selected coupling facility

This panel displays all coupling facilities in a sysplex including particular storage information of each coupling facility. On this panel you can select one of the following actions by entering the command code next to the coupling facility:

#### **D (Drain CF)**

Releases a coupling facility from its structures and connections in order to remove it from the sysplex for maintenance.

#### **E (Enable CF)**

Integrates a coupling facility into a sysplex.

#### **P (Display Sender Paths)**

Displays the paths from the systems in the sysplex having a connection defined to the specified coupling facility. For each path the physical and the logical status are displayed.

#### **S (Display Structures)**

Shows all structures allocated in the specified coupling facility.

These actions are described in detail in the following sections.

**Drain CF:** With this function, you can remove a CF from the sysplex, for example, for maintenance purposes. You can perform the following sequence of tasks:

1. Display information for all allocated structures of the CF.
2. Rebuild all rebuildable structures on another CF and delete instances of structures on the target CF that are being duplexed on another CF.
3. Force the deletion of structures that have no active connectors and cannot be rebuilt.

Note that there are structures that you can neither rebuild nor delete with the force action. These include 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.

4. Disconnect the CF from the systems that it is connected with.

The DRAIN option ensures that these actions are performed in the correct order, as specified above.

Depending on the status of the CF and its allocated structures, you can execute one of four actions with the program function keys:



**REBUILD (PF10)**

Starts the rebuild of structures that can be rebuilt on another CF. Therefore, a rebuild is only initiated for structures whose preference list contains more than one CF.

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 z/OS 2.8 or above; it must 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 CF that are being duplexed on another CF.

**FORCE (F5)**

Forces the deallocation of structures with 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.

This action is only available after all structures that can be rebuilt have been rebuilt.

**DRAIN (F4)**

Disconnects the coupling facility from its connected systems by setting the sender path(s) OFFLINE.

This action is only enabled after all structures of the target CF have been removed to another CF or deallocated. Note that structures that have active connectors but do not support rebuild cannot be removed with PF10 or F5. They must be deallocated manually before execution of this step is enabled.

*Example of Draining a CF:* The status of the CF (NORMAL) and the authorization type of the operator (ALL) are displayed on the right of the INGCF DRAIN panel. The main part of the panel consists of a list of the structures allocated in CF1 and their conditions. The conditions are classified by color and an asterisk. The asterisk signifies that a structure cannot be rebuilt.

The only action that is enabled is REBUILD with PF10. Pressing PF10 leads to a confirmation panel. After PF10 has been pressed, the structures are in the process of being rebuilt.

After the rebuild has been completed, a panel is displayed that shows that one structure could not be rebuilt because no alternate CF is specified in its preference list. The REBUILD status is no longer available. Instead, the FORCE action (F5) is offered because the structure that could not be rebuilt has a condition that allows forcing the deallocation of the structure. Pressing F5 displays a confirmation panel similar to that for REBUILD.

Pressing PF10 on the confirmation panel and refreshing the command dialog after the action has been completed results in a panel that shows that no more structures are allocated in the coupling facility, so 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 the status of the coupling facility changes to INACTIVE.

**Enable CF:** The ENABLE function of the INGCF command is intended to support the integration *and* reintegration 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. Switch to another CFRM policy if the target CF is not defined in the active policy and a suitable policy is available.  
A suitable CFRM policy must contain:
  - A definition of the target CF
  - Appropriate definitions for every active CF and every allocated structure
4. Rebuild all structures on the target CF whose preference list starts with this CF, unless this is excluded by other requirements.

The ENABLE CF option ensures that these actions are performed in the correct order, as specified above.

The following function keys support these actions:

**\*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 CF is active.

**\*SWITCH (F5)**

Switches to another CFRM policy when the target CF 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 CF
- Definitions for every active CF and every allocated structure

This action is only available when the target CF is active, but not defined in the current CFRM policy.

**\*POPULATE (PF10)**

Starts a rebuild process where all structures that have the target CF at the beginning of their preference list, but are currently allocated on another CF, are allocated on the target CF.

This action requires that the CF is 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 CF although it is the preferred CF 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.

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 option ENABLE, 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.

After selection of CF02, INGCF displays a panel that shows that the selected CF has already been activated manually, therefore its status, as shown on the right of the panel, is DRAINED. The authorization type of the operator (ALL) is also displayed on the right of the panel. The main part of the panel is empty because no structures are allocated in CF02. The only action that is activated is ENABLE with PF4. If you press PF4 a confirmation panel is displayed.

After pressing PF10 on the confirmation panel, the command dialog shows that the status has changed to NORMAL, and you can populate the CF with PF10. This implies that the target CF is defined in the active CFRM policy.

The structure list contains three entries with the condition 'Structure is currently allocated in CF01.'. These are the structures that are currently allocated in CF01, but have CF02 at the first position in their preference list.

Pressing PF10 populates the CF, and the refreshed panel shows that the POPULATE action is no longer available because all structures whose preference list starts with CF02 are now allocated in CF02.

**Display Sender Paths:** With this function, the paths from the connected systems to the specified coupling facility are displayed.

The last sender path of each system can only be set to OFFLINE when no more structures are allocated.

The following command codes are available:

**F** Sets the sender path OFFLINE.

**N** Sets the sender path ONLINE.

- 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.

- The **Permission** field shows your authorization level.
- The **System** column contains the names of the systems that are connected to the target CF.
- The **CHPID** column shows the IDs of the sender channel paths.
- The **Physical** column shows the status of the sender channel paths.
- The **Logical** column shows the logical status of the paths to that CF.
- The **Type** column shows the type of the sender channel paths.

**Display Structures:** 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.

- If you have specified INGCF with the STR parameter, the **Coupling Facility** field is an input field. 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 entering Yes or No in this field you determine whether the conditions of the structures are displayed in the **Structure** column.
- The **Permission** field shows your authorization level. There are two possible values, ALL and DISPLAY. DISPLAY indicates that you can only use the display functions. ALL indicates 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.
- The letter P in the **P** column indicates that policy changes are pending for the structure.

A structure has policy changes pending when it was allocated at the time of a CFRM policy switch, and 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 **Condition** column shows the status of the structures. You can switch the display of the conditions on and off with the **Include Condition** field.
- The D field indicates the type of duplexing that is possible. The following values are possible:

<b>U</b>	User-managed duplexing
<b>S</b>	System-managed duplexing
<b>B</b>	User-managed and system-managed duplexing

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 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 z/OS 2.8 or above. It must 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 the section "Structure Conditions" in the description of the INGC command in *IBM Tivoli System Automation for z/OS Operator's Commands*.

The rebuild function works differently depending on the status of the structure (PENDING calls LOCATION=NORMAL, otherwise LOCATION=OTHER).

**\*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 all systems in the Parallel Sysplex have been upgraded to z/OS 1.2 or later with APAR OW41617, and appropriate APARs listed in the CFDUPLEX PSP bucket (for more information, see *System-Managed CF Structure Duplexing*, GM13-0103-03). System-managed duplexing must have 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.

### INGPLEX CDS

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)

This 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 CFlevel** 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 CFlevel** 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 **SMREBLD** field shows whether (value 1) or not (value 0) system-managed rebuild has been activated in the CFRM couple data set.
- The **Max SMlevel** field shows the maximum system-managed process level supported by this system.

For each couple data set you can enter one of the following command codes to initiate an action:

#### \*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. For details see *IBM Tivoli System Automation for z/OS Defining Automation Policy*.

- 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 detailed 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, namely:

##### **display policy (D)**

Displays details about the selected policy.

##### **start policy (S)**

Makes the selected policy the active policy. The policy switch must be confirmed before it is executed.

#### **\*P - switch alternate CDS to primary CDS**

This action makes the alternate CDS the primary. Since an alternate CDS is no longer available after the switch, SA z/OS displays a confirmation panel before the action is performed. On this 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).

**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 initial INGPLEX CDS command dialog panel and press Enter. INGPLEX CDS displays a confirmation panel.

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 PF10 causes SA z/OS to generate the new alternate CDS. After returning to the CDS command dialog, refreshing the panel, and scrolling down with PF8, the panel shows that the previous alternate LOGR CDS has become the primary, and there is a new alternate that 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 initial INGPLEX CDS command dialog panel and press Enter. The CFRM Couple Data Set Information panel 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** field, 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. INGPLEX CDS displays a confirmation panel.

**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 initial INGPLEX CDS command dialog panel and press Enter. The CFRM Channel Path Information panel is displayed.

- The **System** column shows the name of the sysplex members.
- The **T** column (for 'type') indicates whether the CDS is the primary (value 'P') or alternate (value 'A').
- The **DEVN** displays the number of the device that the CDS resides on.
- The **CHPIDs** column shows the status of the paths to the devices in the format *chpid=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 IPL

With the INGPLEX IPL function you can record, view and compare the IPL information of the operating system. 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

Use PF10 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** column shows the name of the system in the sysplex.
  - The **IPL Timestamp** column shows the date and time of the IPL. The format is YYYY-MM-DD HH:MM converted to local time zone.
  - The **Dev** column shows the IPL device number.
  - The **Volume** column shows the volume serial of the IPL device.
  - The **OpSys** column shows the name of the operating system, for example, z/OS or OS/390.
  - The **Release** column shows the release level of the operating system.
  - The **FMID** column shows the FMID of the operating system.

For further information about the panel fields refer to the online help.

## INGPLEX Dump Submenu

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.

## INGPLEX Dump

**INGPLEX SDUMP:** If you select option 1, the INGPLEX SDUMP panel 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:

**Q=** Shows whether 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 modification panel is displayed, which 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 on. 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:** With the INGPLEX SVCDUMP command you can display the dump status information of each system in the sysplex. 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.

On the INGPLEX SVCDUMP target system selection panel:

- The **Sel** field lets you select a system that a dump should be taken for.
- 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 INGPLEX SVCDUMP Address Space Selection panel is displayed.

If you select the VTAM address space and the WATS address space, which is a user, press Enter and then PF5, the INGPLEX SVCDUMP Address Space Detail panel is displayed.

The VTAM address space 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 PF5, the dump option selection panel is displayed. It 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.

**INGPLEX SLIP:** 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.

The following command codes are available on this panel:

- + 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.

### How to Work with INGSTR

If you enter the INGSTR command without any parameters, a panel with all the coupling facilities of the sysplex is displayed.

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.

**B** 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.

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. 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.

## How SA z/OS Uses Error Thresholds

Error thresholds influence whether SA z/OS recovers from an error situation. For applications, you can define a critical threshold for restarting these applications. This is a number of error conditions within a certain time interval, for example, five error conditions requiring restart within one hour. During a condition requiring restart, SA z/OS checks whether the number of occurrences of the condition reaches the critical threshold. If it is reached, SA z/OS does not attempt to restart a resource.

For z/OS components, such as dump data sets or log data sets, you can define thresholds to limit the frequency of how often they may be deleted after they have filled up without an action being taken or a notification being sent to the operator.

Error thresholds also determine when you should be alerted to problems. The primary use of error thresholds is to track subsystem abends and ensure that the abend and restart cycle does not become an infinite loop, but they may also be customized for other uses.

Refer to *IBM Tivoli System Automation for z/OS Defining Automation Policy* for information on how to define error thresholds. The following sections describe how to obtain information about them.

### Displaying Threshold Settings and Occurrences

SA z/OS counts the number of times that a subsystem abends. Each time, an abending subsystem is restarted, a check is made to see if the restart is done too often. SA z/OS allows you to specify thresholds in terms of error rates. If the measured error rate reaches the *critical* threshold, SA z/OS stops to recover the resource and posts it to a broken status.

You have the following possibilities to both obtain information about threshold settings and to view when thresholds have been reached:

#### INGTHRES

If you want to display all defined thresholds or add, change, or delete threshold settings for a particular resource, use the INGTHRES command. This displays the related Command Dialogs panel.

#### DISPERRS

If you want to check the error counts for all resources on a specific system, you should issue a DISPERRS command to this target system. SA z/OS will display the DISPERRS Command Dialogs panel with a list of all the errors that have been recorded for all resources on the target system (see “How to Display Occurred Errors: DISPERRS” on page 140).

#### DISPASF

If you want to check the error counts for a specific resource on a specific target system, issue the DISPASF command.

### How to Display Threshold Settings for a Selected Resource: INGTHRES

Enter INGTHRES on any command line to display a panel that shows the settings for the critical, frequent, and infrequent thresholds for the specified resource.

## How SA z/OS Uses Error Thresholds

### How to Display Occurred Errors: DISPERRS

If you specify the DISPERRS command, you get the DISPERRS command dialog, which displays the number of errors for all the applications on the target system and also when an application has reached its threshold. Scroll through the panel to find the resource that has failed.

If a subsystem is in the BROKEN status and it has not exceeded its critical threshold, it has probably encountered an unrecoverable error. If you look into either SDF or DISPINFO for it, you should find the text of the AOF57II status change notification message, which will explain why it was posted to BROKEN.

### How to Display Occurred Errors for a Specific Resource: DISPASF

While DISPERRS displays occurred errors for all a system's resources, DISPASF lets you see detailed information about errors for a resource on a specific system. For example, if you want to see more information about the error with resource MVSDUMP on system AOC7 from the DISPERRS command dialog panel, issue the following command:

```
DISPASF MVSDUMP TARGET=AOC7
```

## How to Set or Change Error Thresholds

You can set or change thresholds for a resource, the resource group defaults, or the system defaults using the INGTHRES command dialog.

1. Follow the instructions in "How to Display Threshold Settings for a Selected Resource: INGTHRES" on page 139 to specify the resource that you want to set thresholds for and to access the INGTHRES command dialog.
2. To change the Critical, Frequent, or Infrequent thresholds, type the number of errors and specify the time period.

---

## Handling Jobs That Are Not Controlled by SA z/OS

The INGLKUP command dialog displays all active jobs, started tasks (STC), and APPC/MVS transaction programs that are not controlled by SA z/OS. You can either stop, cancel, or force these jobs.

For each job the following information is displayed:

- Job name
- Step name
- Procedure name
- Job type
- Address space ID (this is only shown when the job name is not unique)

---

## Communicating with Other Systems

This chapter discusses the following topics:

- "Sending Commands within a Sysplex"
- "Terminal Access Facility Full-Screen Sessions" on page 143
- "Gateway Sessions" on page 144

### Sending Commands within a Sysplex

This section informs you about the prerequisites for communicating with systems in the following environments:

- Within a sysplex; this scenario is called *single system image*.
- Within your complete enterprise; this scenario is called *single point of control for the enterprise*.

There is also a series of commands that is directly processed by the automation manager of a sysplex. For these commands, you need no target parameter. You can retrieve sysplexwide information, such as the statuses of resources within the sysplex. For a complete list of sysplexwide commands, refer to the *IBM Tivoli System Automation for z/OS Operator's Commands*.

You can use a system within the sysplex to monitor and manipulate all other systems in the sysplex. Then each system in the sysplex looks like the one you are currently working on (single system image).

If your system is the focal point system, you can use it to manipulate all resources of your enterprise from this focal point system issuing NetView commands. The target systems need not be members of your sysplex (single point of control).

To exploit these features, the following prerequisites and conditions need to be fulfilled:

- The target system is within the local sysplex (this is valid for single system image) or the target system has a direct gateway connection with the local system (this is valid for single point of control).

The following prerequisites need to be fulfilled for both cases:

- The target is active.
- SA z/OS on the target has been fully initialized.
- The target can be reached either by XCF or RMTCMD.

SA z/OS will search your environment for the target system in the following order:

- System name within the local sysplex
- Domain ID within the local sysplex
- Local sysplex name
- Domain ID within the enterprise
- System name within the enterprise
- Sysplex name within the enterprise

You have several choices as to how to route commands to other systems within the sysplex. These choices are described in the following subsections.

### **INGSEND**

This will assist you in sending a command over a semi-shared RMTCMD session (this requires some configuration).

1. Enter **INGSEND** without parameters on the command line of any NetView or system operations panel to display the **INGSEND** panel.
2. Fill out the other fields and press Enter. (For more information on the individual fields, refer to *IBM Tivoli System Automation for z/OS Operator's Commands*).

## Communicating with Other Systems

### Single System Image and Single Point of Control

If the command you want to send is an SA z/OS command and is supported by single system image (most of them are), you can just issue it, specifying the destination system as the target. This only requires that NetView is active on both systems: VTAM is not used.

The following example shows the usage of single system image to determine the status of APPL1 on every system within the sysplex (AOCA, AOCC, AOCD) and then to restart it on AOCC where it is BROKEN. All operations are carried out from the IPUFA system.

```
DISPSTAT APPL1,TARGET=*ALL
```

CMD	RESOURCE	STATUS	SYSTEM	JOB NAME	A	I	S	R	T	RS	TYPE	Activity
---	---	---	---	---	---	---	---	---	---	---	---	---
-	APPL1	AUTODOWN	AOCA	APPL1	Y	Y	Y	Y	Y	Y	TRANS	--none--
-	APPL1	DOWN	AOCC	APPL1	Y	Y	Y	Y	Y	Y	TRANS	--none--
B	APPL1	BROKEN	AOCC	APPL1	Y	Y	Y	Y	Y	Y	TRANS	--none--
-	APPL1	CTLDOWN	AOCD	APPL1	Y	Y	Y	Y	Y	Y	TRANS	--none--

Enter B against APPL1 on AOCC and press Enter.

Complete the command parameters (STATE=RESTART, START=YES) and press Enter to issue the command. It will be sent to the target system and executed.

### Cross-Domain Sessions

Cross-domain sessions let you send commands to other domains using NetView RMTCMD, gateways, or Terminal Access Facility (TAF).

To display cross domain sessions, from the NetView command line, type A0C to invoke the SA z/OS main menu.

From the SA z/OS main menu, type 1 (Operator Interface) and press the Enter key. This takes you to the Main Operator Menu. On the command line, type 1 (CROSS DOMAIN) and press the Enter key. This leads to the Cross Domain Menu. From this menu you can display cross-domain sessions.

## Sending Commands to a Linux Target System

Use the ISQSEND command of processor operations to send a command to a Linux target system. The system previously must have been initialized successfully, using the ISQXIII command. In order to see the responses on the NetView screen, the command ISQXMON must have been used to add you to the interested operator list of the target system.

Since the Linux operating system handles commands case sensitive, you have to use the "netvasis" prefix. In the following example, the Linux command ps -x is issued to show Linux process related information. Note, that NetView suppresses the "netvasis" prefix when redisplaying the entered command.



```

NCCF                               Tivoli Netview  IPVFJ TIL      05/28/01 15:19:09
* IPVFJ  ISQSEND FCLX2 OC PS -X
U IPVFJ  ISQ901I FCLX2 SC ISQ417I CMD STATUS(ACCEPTED)
U IPVFJ  ISQ901I FCLX2  OC PS -X
U IPVFJ  ISQ017I ISQSEND COMPLETED SUCCESSFULLY.
U IPVFJ  ISQ901I FCLX2  OC  PID TTY  STAT  TIME COMMAND
U IPVFJ  ISQ901I FCLX2  OC   1 ?   S     0:00 init
U IPVFJ  ISQ901I FCLX2  OC   2 ?   SW    0:00 ckMCHECK!
U IPVFJ  ISQ901I FCLX2  OC   3 ?   SW    0:00 ckeventd!
U IPVFJ  ISQ901I FCLX2  OC   4 ?   SW    0:00 ckswapd!
U IPVFJ  ISQ901I FCLX2  OC   5 ?   SW    0:00 ckreclaimd!
U IPVFJ  ISQ901I FCLX2  OC   6 ?   SW    0:00 cdbufush!
U IPVFJ  ISQ901I FCLX2  OC   7 ?   SW    0:00 ckupdated!
U IPVFJ  ISQ901I FCLX2  OC  306 ?   SW    0:00 ckeventd!
U IPVFJ  ISQ901I FCLX2  OC  309 ?   SW    0:00 ckeventd!
U IPVFJ  ISQ901I FCLX2  OC  312 ?   SW    0:00 ckeventd!
U IPVFJ  ISQ901I FCLX2  OC  315 ?   SW    0:00 ckeventd!
U IPVFJ  ISQ901I FCLX2  OC  436 ?   SN    0:00 syslogd -M 0
U IPVFJ  ISQ901I FCLX2  OC  450 ?   SN    0:00 klogd
U IPVFJ  ISQ901I FCLX2  OC  469 ?   SN    0:00 crond
U IPVFJ  ISQ901I FCLX2  OC  488 ?   SN    0:00 inetd
U IPVFJ  ISQ901I FCLX2  OC  507 ?   SN    0:00 httpd
U IPVFJ  ISQ901I FCLX2  OC  554 console S    0:00 -bash
U IPVFJ  ISQ901I FCLX2  OC  603 console R    0:00 ps -X
U IPVFJ  ISQ901I FCLX2  OC  croot@boe1nfc2 /root!#
-----
???
```

Figure 27. Sending a Command to a Linux Target System

## Terminal Access Facility Full-Screen Sessions

The NetView terminal access facility (TAF) allows you to log on to multiple applications either on your system or other systems. *Tivoli NetView for z/OS Command Reference* describes in detail how to use TAF.

SA z/OS allows you to define TAF sessions in the customization dialog so that you do not have to set them up each time you want to use them. Refer to *IBM Tivoli System Automation for z/OS Defining Automation Policy* for more details.

### Determining the Status of TAF Full-Screen Sessions

Using the TAF Fullscreen Menu panel you can see the list of TAF fullscreen sessions that have been defined to SA z/OS for your operator ID, and determine which of these are active.

To see the list of TAF fullscreen sessions available to you:

1. From the NetView command line, type **Oper**. You see the Main Operator Menu.
2. On the Selection line, type **1** (CROSS DOMAIN) and press the Enter key. You see the Cross Domain Menu.
3. On the Action line, type **2** and press the Enter key. You see the TAF Fullscreen Menu.

**Note:** If no TAF sessions are defined for your operator ID, you will see a line mode message

```
AOF041I UNABLE TO FIND FULL SESSION ENTRIES
```

## Communicating with Other Systems

```
A0FK2DF          SA z/OS - Command Dialogs          Line
Domain ID  = IPUFM    --- TAF Fullscreen Display --- Date = 05/11/04
Operator ID = OPER1      System = AOC7              Time = 16:34:53

Cmd:  I Initialize    R Return to    T Terminate

Cmd Name      Application  System    Status    SRCLU      Session id
-----
-   TSO          IPSFM     KEY3      ACTIVE    TAFFJF35
-   IMS          IPSFN     KEY4      INACTIVE
-

Command ==>
PF1=Help      PF2=End          PF3=Return      PF6=Ro11
                PF9=Refresh      PF12=Retrieve
```

Figure 28. TAF Fullscreen Display Panel

This panel shows the settings of all fullscreen sessions that are defined in the policy database.

### Managing TAF Full-Screen Sessions

You can use the *TAF Fullscreen Display* panel to log on to other applications on the current or another system to return to the session or to terminate the session.

For each TAF session the following information is displayed:

**Name** Is the name of the session.

#### Application

Is the name of the application as defined in the VTAMLST data set.

#### System

Is the name of the system where the application runs. This name is defined in the policy database.

**Status** Is the current status of the TAF session.

#### SRCLU

Is the logical unit name of the session.

#### Session ID

Is the session ID as assigned by NetView.

You can use the following command codes:

**I** Initializes the fullscreen session.

**R** Returns to session.

**T** Terminates the fullscreen session.

## Gateway Sessions

In SA z/OS you can monitor other systems and issue commands to those systems using gateway sessions.

Gateways are useful because they:

- Save you the step of logging on to the NetView on the other system
- Let you communicate with another system even if you do not have an ID for that system
- Can provide a second NetView session with a system if you have an ID for that system

SA z/OS provides several panels that let you check your communication links with other systems and send commands to those systems. You can:

- Determine the status of gateway sessions
- Determine the status of paths. A path uses an intermediate system to connect two systems that are not directly linked
- Issue commands to other systems using gateway sessions

### **Determining the Status of Gateway Sessions**

You can determine which gateway sessions have been defined and see which of those are active. You may want to do this if you use SDF to monitor other systems, since SDF uses gateway sessions to receive information from remote systems.

You also need to check the status of a gateway session if you want to receive notification messages from other systems. (Notification messages are messages that document significant actions that SA z/OS has detected or taken.) Finally, you need to check the status of a gateway session if you want to issue a command to another system using a gateway session.

You can obtain information on gateway sessions from the Gateway Display panel or from SDF, if SDF has been customized to do this.

To access the Gateway Display panel, type DISPGW from any command line and press Enter.

For further information on the DISPGW command, see *IBM Tivoli System Automation for z/OS Operator's Commands*.

## Communicating with Other Systems

---

## Chapter 8. Scenario: Finding Out Why a Resource Is Not Available

For most operators, the key resource to be monitored is the application. You want to be sure that your enterprise's critical jobs and subsystems are getting the service they require and are operating satisfactorily. The scenario contained in this chapter gives you a general idea of how to use SA z/OS commands to discover and react to problems affecting your important applications.

### Consider the following scenario:

You get a phone call from one of your users: Resource MSCSK1 is not available. Find out why and start it.

Here are the steps to make the resource available:

1. Use the INGLIST command to find out details (especially the status information) about resource MSCSK1. Because you know the name of the resource, you can issue the command as follows on your NetView console:  
INGLIST MSCSK1
2. Suppose you get an output screen as shown in Figure 29 that shows that the compound status for MSCSK1 is SATISFACTORY. This results from the fact that the observed status, SOFTDOWN, fulfills the desired status of UNAVAILABLE even though you and your user expect it to be available. This needs further investigation.

```
INGKYST0          SA z/OS - Command Dialogs          Line 1 of 1
Domain ID = IPUFM  ----- INGLIST -----          Date = 11/07/06
Operator ID = SYSAD1      Sysplex = AOC9PLEX          Time = 14:10:36
CMD: A Update   B Start   C Stop    D INGRELS   E INGVOTE   F INGINFO
      G Members  H DISPTRG I INGSCHED J INGGROUP M DISPMTR / scroll
CMD Name      Type System  Compound  Desired   Observed  Nature
-----
E MSCSK1      APL  AOC9   SATISFACTORY UNAVAILABLE SOFTDOWN

Which votes are involved?

Command ==>
PF1=Help    PF2=End    PF3=Return  PF4=DISPSTAT  PF5=Filters  PF6=Roll
PF9=Refresh PF10=Previous PF11=Next   PF12=Retrieve
```

Figure 29. INGLIST Output

3. To find out what votes have caused the resource to be unexpectedly unavailable (SOFTDOWN), use the INGVOTE Command Dialog. Enter the action code E in the CMD column next to the MSCSK1 resource in the INGLIST command dialog panel, as shown in Figure 29.

- From the INGVOTE command dialog panel, as shown in Figure 30, you can see that there has been an operator request (MakeUnAvailable) with the highest priority (X'005A0000') to stop the resource; this has obviously been satisfied.

```

INGKYRQ0          SA z/OS - Command Dialogs          Line 1 of 4
Domain ID = IPUFM ----- INGVOTE -----          Date = 11/07/06
Operator ID = SYSAD1      Sysplex = AOC9PLEX          Time = 14:13:14

Resource ==> MSCSK1/APL/AOC9
System ==>                               System name, domain id or sysplex name

Cmd: C cancel request  K Kill request  S show request details
Cmd Action WIN Request/Vote Data
-----
C  STOP   Y  Request   : MakeUnAvailable
              Originator : OPERATOR(MSCH)
              Priority   : 005A0000    Must Be Down - Operator
              Status    : Winning/Satisfied

Command ==>
PF1=Help    PF2=End      PF3=Return    PF6=Roll
              PF9=Refresh  PF12=Retrieve

```

Figure 30. INGVOTE Output

- Because this request is persistent and can only be beaten by another request with priority FORCE, you must cancel it to allow the resource to become available again. (The request obviously stems from an operator's INGREQ command with high priority that is the strongest request you can issue with SA z/OS. See also "Automation Concepts" on page 13 for more information about the hierarchy of requests). Enter action code C in the CMD column next to the request as indicated in Figure 30.
- On the INGVOTE command dialog panel, you see confirmation that the request has been canceled. Press PF3 to go back to the INGLIST command dialog and press the PF9 key to refresh the information shown for resource MSCSK1. Assume that you see the output as shown in Figure 31 on page 149.

```

INGKYST0          SA z/OS - Command Dialogs      Line 1    of 1
Domain ID  = IPUFM  ----- INGLIST -----      Date = 11/07/06
Operator ID = SYSAD1      Sysplex = AOC9PLEX      Time = 14:16:00
CMD: A Update  B Start   C Stop    D INGRELS  E INGVOKE  F INGINFO
      G Members  H DISPTRG I INGSCHED J INGGROUP M DISPMTR / scroll
CMD Name      Type System  Compound  Desired   Observed  Nature
-----
F MSCSK1      APL  AOC9    AWAITING  AVAILABLE SOFTDOWN

```

**Refresh INGLIST status with PF9**

```

Command ==>
PF1=Help   PF2=End       PF3=Return  PF4=DISPSTAT  PF5=Filters  PF6=Roll
PF9=Refresh PF10=Previous PF11=Next   PF12=Retrieve

```

Figure 31. Refreshed INGLIST Output

What you now see is that the compound status for MSCSK1 is AWAITING (awaiting automation) and the observed status is still SOFTDOWN, which is not exactly what you might have wanted or expected. You want the observed and desired statuses to be AVAILABLE. There must be something else that is inhibiting this. Maybe there are dependency relationships to supporting resources that are not fulfilled.

- 7. Use INGINFO (action code F) to display more information about why MSCSK1 is still awaiting automation. Scroll down the INGINFO output panel until you reach a panel similar to the one shown in Figure 32.

```

INGKYIN0          SA z/OS - Command Dialogs      Line 29   of 197
Domain ID  = IPUFM  ----- INGINFO -----      Date = 11/07/06
Operator ID = SYSAD1      Sysplex = AOC9PLEX      Time = 14:17:37

Resource ==> MSCSK1/APL/AOC9          format: name/type/system
System    ==> _____ System name, domain ID or sysplex name

Backward Relationships :
AOC9/SYG/AOC9          HasMember

Forward Relationships :
MSCSK2/APL/AOC9      HasParent
NETVIEW/APL/AOC9      HasParent

Requests      :
Votes         :
Origin...

Command ==>
PF1=Help   PF2=End       PF3=Return  PF4=DISPINFO  PF6=Roll
PF7=Back   PF8=Forward  PF9=Refresh  PF12=Retrieve

```

Figure 32. INGINFO Output

From this output panel you find out that MSCSK1 has a *HasParent* dependency on the supporting resource MSCSK2/APL/AOC9.

The current situation as you have investigated it until now using the various commands is that an operator has made MSCSK1 unavailable by issuing a SOFTDOWN request and that MSCSK1 is dependent upon MSCSK2.

8. You now need to find out more about MSCSK2. To do this, proceed as in step 1 on page 147 using the INGLIST command:

INGLIST MSCSK\*

You get output similar to Figure 33

```

INGKYST0          SA z/OS - Command Dialogs          Line 1 of 2
Domain ID = IPUFM ----- INGLIST ----- Date = 11/07/06
Operator ID = SYSAD1          Sysplex = AOC9PLEX          Time = 14:19:32
CMD: A Update  B Start  C Stop  D INGRELS  E INGVOTE  F INGINFO
      G Members  H DISPTRG  I INGSCHED  J INGGROUP  M DISPMTR  / scroll
CMD Name      Type System  Compound  Desired  Observed  Nature
-----
MSCSK1        APL  AOC9    AWAITING  AVAILABLE  SOFTDOWN
E MSCSK2        APL  AOC9    SATISFACTORY  UNAVAILABLE  SOFTDOWN

Which votes are involved?

Command ==>
PF1=Help  PF2=End  PF3=Return  PF4=DISPSTAT  PF5=Filters  PF6=Roll
PF9=Refresh  PF10=Previous  PF11=Next  PF12=Retrieve

```

Figure 33. INGLIST Output for MSCSK\* Resources

This panel informs you that MSCSK2 is also not available (its observed status is SOFTDOWN), and that this even conforms to its desired status. This is obviously not what you want, so now you must find out the reason for this.

9. As in step 3 on page 147, you use action code E to find out the votes that unexpectedly caused the MSCSK2 resource to be unavailable (SOFTDOWN). SA z/OS might display a panel as shown in Figure 34 on page 151.



```

INGKYRQ0          SA z/OS - Command Dialogs      Line 1 of 4
Domain ID = IPUFM ----- INGVOTE -----      Date = 11/07/06
Operator ID = SYSAD1      Sysplex = AOC9PLEX      Time = 14:20:43

Resource ==> MSCSK2/APL/AOC9
System ==>          System name, domain id or sysplex name

Cmd: C cancel request  K Kill request   S show request details
Cmd Action WIN Request/Vote Data
-----
STOP  Y Request   : MakeUnAvailable_Only
        Originator : SCHEDULE
        Priority    : 00520000      Must Be Down - Schedule
        Status     : Winning/Satisfied

Command ==>
PF1=Help      PF2=End          PF3=Return      PF6=Ro11
              PF9=Refresh     PF12=Retrieve

```

Figure 34. INGVOTE Output for MSCSK2

From this panel you can see that there has been a request (MakeUnAvailable) generated by a service period (SCHEDULE) to stop the resource MSCSK2. This obviously has been satisfied.

- To find out more about the service period that has been defined for MSCSK2, on the INGLIST panel, enter action code **I** in the **CMD** column next to MSCSK2. Suppose that you see a display as shown in Figure 35. From this you can see that the resource MSCSK2/APL/AOC9 is scheduled with high priority to be down on Tuesdays between 14:10 and 16:00, and that the current time is 14:21:50, which is actually within the requested downtime (and indeed, the indicated date of 11/07/06 is a Tuesday).

```

INGKYSP2          SA z/OS - Command Dialogs      Date = 11/07/06
Domain ID = IPUFM ----- INGSCHED -----      Time = 14:21:50
Operator ID = SYSAD1      Sysplex = AOC9PLEX

Resource . . . : MSCSK2/APL/AOC9      Schedule . . . : MSCSCED8
Starting date => 11 / 07 / 2006      (mm/dd/yyyy)

C Date          Pri From-To      From-To      From-To      From-To      From-To
-----
11/07/06 Tue UP H 1320 1340 1400 1402
                   DN H 1300 1319 1341 1359 1410 1600
11/08/06 Wed UP H 1320 1340 1400 1402
                   DN H 1300 1319 1341 1359
11/09/06 Thu UP H 1320 1340 1400 1402
                   DN H 1300 1319 1341 1359
11/10/06 Fri UP H 1320 1340 1400 1402
                   DN H 1300 1319 1341 1359
11/11/06 Sat UP H 1320 1340 1400 1420 1500 1520 1540 1600 1640 2359
                   DN H 1300 1319 1341 1359 1440 1459 1521 1539 1620 1639

Command ==>
PF1=Help      PF2=End          PF3=Return      PF6=Ro11
PF7=Backward  PF8=Forward     PF9=Refresh     PF12=Retrieve

```

Figure 35. INGSCHED for MSCSK2

- To allow automation to make the MSCSK2 available and keep it available for the rest of the day, you could now specify an additional uptime schedule for Tuesdays, starting from now (14:26 PM) until midnight (24:00) as shown in Figure 36. This method is called a schedule override (because it overrides a schedule defined by automation policy).

**Note:**

Schedule overrides made by INGSCHED are permanent.

```

INGKYSP2          SA z/OS - Command Dialogs
Domain ID   = IPUFM   ----- INGSCHED -----   Date = 11/07/06
Operator ID = SYSAD1   Sysplex = AOC9PLEX           Time = 14:25:18

Resource . . . : MSCSK2/APL/AOC9          Schedule . . . : MSCSCED8
Starting date => 11 / 07 / 2006          (mm/dd/yyyy)

C Date          Pri  From-To    From-To    From-To    From-To    From-To
-----
11/07/06 Tue UP  H  1320 1340  1400 1402  1426 2400  _____
                   DN  H  1300 1319  1341 1359  1410 1600  _____
11/08/06 Wed UP  H  1320 1340  1400 1402  _____
                   DN  H  1300 1319  1341 1359  _____
11/09/06 Thu UP  H  1320 1340  1400 1402  _____
                   DN  H  1300 1319  1341 1359  _____
11/10/06 Fri UP  H  1320 1340  1400 1402  _____
                   DN  H  1300 1319  1341 1359  _____
11/11/06 Sat UP  H  1320 1340  1400 1420  1500 1520  1540 1600  1640 2359
                   DN  H  1300 1319  1341 1359  1440 1459  1521 1539  1620 1639

AOF099I FUNCTION SUCCESSFULLY COMPLETED
Command ==>
PF1=Help      PF2=End      PF3=Return    PF6=Roll
PF7=Backward PF8=Forward  PF9=Refresh   PF12=Retrieve

```

Figure 36. Schedule Override for MSCSK2

- Using PF3, go back to the INGLIST command dialog and press PF9 to see if any changes have occurred. From Figure 37 you can see that MSCSK2 is now available, but MSCSK1 is still SOFTDOWN, AWAITING automation.

```

INGKYST0          SA z/OS - Command Dialogs   Line 1 of 2
Domain ID   = IPUFM   ----- INGLIST -----   Date = 11/07/06
Operator ID = SYSAD1   Sysplex = AOC9PLEX           Time = 14:26:23
CMD: A Update  B Start  C Stop  D INGRELS  E INGVOTE  F INGINFO
      G Members  H DISPTRG  I INGSCHED  J INGGROUP  M DISPMTR  / scroll
CMD Name      Type System  Compound  Desired  Observed  Nature
-----
MSCSK1        APL  AOC9      AWAITING  AVAILABLE  SOFTDOWN
MSCSK2        APL  AOC9      SATISFACTORY  AVAILABLE  AVAILABLE

Command ==>
PF1=Help      PF2=End      PF3=Return    PF4=DISPSTAT  PF5=Filters    PF6=Roll
PF9=Refresh   PF10=Previous PF11=Next     PF12=Retrieve

```

Figure 37. Refreshed INGLIST Output

You might thus suppose that an automation flag is inhibiting startup or automation for this resource.

13. Next you check the automation flags for MSCSK1. You can do this by scrolling the INGLIST output from Figure 37 on page 152 to the right with PF11 until you reach a display similar to Figure 38.

```

INGKYST0          SA z/OS - Command Dialogs      Line 1 of 2
Domain ID = IPUFM  ----- INGLIST -----      Date = 11/07/06
Operator ID = SYSAD1      Sysplex = AOC9PLEX      Time = 14:27:20
CMD: A Update  B Start  C Stop  D INGRELS  E INGVOTE  F INGINFO
      G Members  H DISPTRG  I INGSCHED  J INGGROUP  M DISPMTR  / scroll
CMD Name      Type System  Automation  Startable  Health      Auto Hold
-----
_ MSCSK1      APL AOC9      IDLE       YES        N/A         YES NO
_ MSCSK2      APL AOC9      IDLE       YES        N/A         YES NO

Command ==>
PF1=Help  PF2=End      PF3=Return  PF4=DISPSTAT  PF5=Filters  PF6=Roll
PF9=Refresh PF10=Previous PF11=Next   PF12=Retrieve

```

Figure 38. Checking Automation Flags with INGLIST

However, the result shows that MSCSK1 is startable and automation is allowed for it (Auto=YES and Hold=NO). So what else might be preventing MSCSK1 from starting?

14. There might be a *trigger* for MSCSK1 whose events for its startup condition are not set, thus preventing the resource from starting. Scroll horizontally in your current INGLIST output until you see the column **Trigger** as shown in Figure 39.

**Note:** The DISPSTAT command cannot be used because it only displays service periods and triggers from downlevel SA z/OS systems.

```

INGKYST0          SA z/OS - Command Dialogs      Line 1 of 2
Domain ID = IPUFM  ----- INGLIST -----      Date = 11/07/06
Operator ID = SYSAD1      Sysplex = AOC9PLEX      Time = 14:28:33
CMD: A Update  B Start  C Stop  D INGRELS  E INGVOTE  F INGINFO
      G Members  H DISPTRG  I INGSCHED  J INGGROUP  M DISPMTR  / scroll
CMD Name      Type System  Starttype  Stoptype  Trigger  Schedule
-----
_ MSCSK1      APL AOC9      -----  -----  MSCTRG1
_ MSCSK2      APL AOC9      -----  -----  -----  MSCSCED8

Command ==>
PF1=Help  PF2=End      PF3=Return  PF4=DISPSTAT  PF5=Filters  PF6=Roll
PF9=Refresh PF10=Previous PF11=Next   PF12=Retrieve

```

Figure 39. Checking Triggers with INGLIST

From this panel, you see that a trigger, MSCTRG1, is connected to MSCSK1. See Figure 40 on page 154 for the automation scenario for resource MSCSK1:

- MSCSK1 is dependent on MSCSK2. So MSCSK2 must be up before MSCSK1 can be started.
- The availability of MSCSK2 is controlled by a schedule. So the time of day must be within one of the uptime schedules that are defined for this resource.

- MSCSK1 has a trigger connected to it. On top of all the other prerequisites for starting this resource, at least one startup condition for that trigger must be fulfilled to make this resource available.

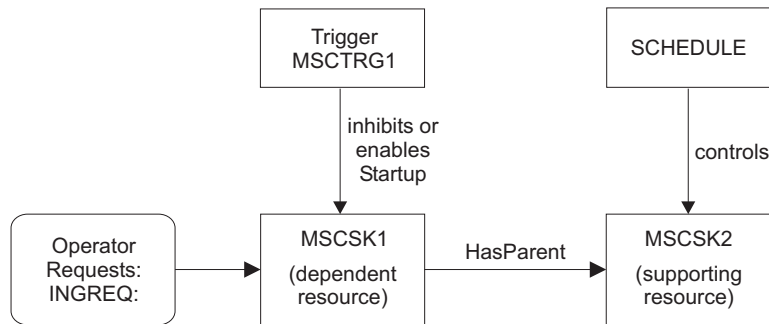


Figure 40. Complete Automation Environment for MSCSK1

15. In this step, you check the startup conditions for trigger MSCTRG1. You achieve this by entering action code **H** (for DISPTRG) in the **CMD** column next to MSCSK1. This displays triggers and their events for the selected resource. In this scenario, we get the DISPTRG output shown in Figure 41.

```

AOFLT000          SA z/OS - Command Dialogs          Line 1 of 2
Domain ID = IPUFM  ----- DISPTRG -----         Date = 11/07/06
Operator ID = SYSAD1      System = AOC9              Time = 14:29:47

Resource ==> MSCSK1/APL/AOC9
System ==> _____ System name, domain id or sysplex name
Trigger. . . : MSCTRG1 Observed status : SOFTDOWN

Cmd: S show details
Cmd Type      Events
-----
S  STARTUP    MSCEVTUP
_  SHUTDOWN   MSCEVTDN

Command ==>
PF1=Help      PF2=End      PF3=Return   PF6=Ro11
               PF9=Refresh  PF12=Retrieve
  
```

Figure 41. DISPTRG Command Dialog

In this output, you can see that trigger MSCTRG1 has a startup condition that is fulfilled if event MSCEVTUP is set. This is what you need to find out in the next step.

16. To see more details about an event (or about the condition), enter action code **S** in the **CMD** column next to the startup condition (event MSCEVTUP). This produces the output shown in Figure 42 on page 155.

```

AOFLT100          SA z/OS - Command Dialogs      Line 1 of 1
Domain ID = IPUFM  --- Trigger Condition List ---- Date = 11/07/06
Operator ID = SYSAD1      System = AOC9          Time = 14:30:46

Trigger . . . . : MSCTRG1
Condition type. : STARTUP
Resource. . . . : MSCSK1/APL/AOC9              Observed status : SOFTDOWN

Cmd: D show resources  S set event for resource  U unset event for resource
Cmd Event  Status      Unset  Description
-----
  S  MSCEVTUP  UNSET    UP    scheduling test event UP

Command ==>
PF1=Help      PF2=End          PF3=Return      PF6=Ro11
                PF9=Refresh      PF12=Retrieve

```

Figure 42. Trigger Condition List

- 17. From Figure 42, you can see that the event MSCEVTUP is not set (column **Status** contains UNSET). Enter action code **S** (to set the event for the resource) in the **CMD** column next to the event MSCEVTUP. this sets the event to fulfill the startup condition for trigger MSCTRG1, thus allowing the resource MSCSK1 to start.

SA z/OS will now put your environment into the status you desired. After a few minutes, you should be able to verify this. View the INGLIST command dialog for both resources involved (enter INGLIST MSCSK\*), as shown in Figure 43.

```

INGKYST0          SA z/OS - Command Dialogs      Line 1 of 2
Domain ID = IPUFM  ----- INGLIST ----- Date = 11/07/06
Operator ID = SYSAD1      Sysplex = AOC9PLEX      Time = 14:32:52
CMD: A Update  B Start  C Stop  D INGRELS  E INGVOTE  F INGINFO
      G Members  H DISPTRG  I INGSCHED  J INGGROUP  M DISPMTR  / scroll
CMD Name      Type System  Compound  Desired  Observed  Nature
-----
  _  MSCSK1      APL  AOC9      SATISFACTORY  AVAILABLE  AVAILABLE
  _  MSCSK2      APL  AOC9      SATISFACTORY  AVAILABLE  AVAILABLE

Command ==>
PF1=Help      PF2=End          PF3=Return      PF4=DISPSTAT  PF5=Filters  PF6=Ro11
                PF9=Refresh      PF10=Previous  PF11=Next     PF12=Retrieve

```

Figure 43. Verifying Automation Status

Both resources now have AVAILABLE for the observed and desired status and thus a compound status of SATISFACTORY.



---

## Chapter 9. Using I/O Operations

I/O operations is a highly flexible connection environment that expands the distance for channel and control unit communication.

You can use the SA z/OS I/O operations component:

- To get a single point of logical control for managing connectivity in your active I/O configurations.
- To have information presented on the operator console that you issue commands from or from the I/O Operations ISPF dialog. You can also use the I/O operations application program interface (API) to get information via a user-written call.
- As a tool for automating these 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/FICON director).

To fulfil the tasks of managing and automating I/O configurations, various commands are available, like for example, a command to define a dynamic connection between ports, commands to block or permit data transfer through ports or a series of commands to display I/O configuration information.

You can issue these commands either through the ISPF dialog (see Appendix B, "Using the I/O Operations ISPF Dialog," on page 223) or from the operator console.

More features of the I/O operations are, for example:

- For an I/O resource that passes through a dynamic switch, I/O operations can detect and report several types of I/O configuration mismatches. It reports these mismatches when the program is started or when the configuration is dynamically reconfigured.
- You can use I/O operations as a single point of control:

I/O operations lets you change the connectivity of I/O resources. From any one system where I/O operations is running, you may affect the I/O connectivity of a target system that I/O operations is also operating on, thereby providing a single point of control.

I/O operations performs these functions when you enter, or send, one of its connectivity commands to one of the hosts that the program is running on. That host issues the commands to the affected hosts. The hosts that can be affected by the command depend on several factors, including the domain of the issuing host.

With the feature of a single point of control, you can:

- Remove the dependency on a dynamic switch
- Manage the devices across hosts; that is, you can vary devices on or off
- Remove and restore channel paths among system images and have the pathing status automatically adjusted
- Manage switchable paths in a system-integrated manner

With I/O operations, you can display and control the following I/O resources:

- **Channels** of the following types:

- ESCON/FICON channels
- Parallel channels
- Coupling facility channels
- **Dynamic switches**, such as the ESCON/FICON Directors
- **Ports**, which are the interfaces on a dynamic switch
- **Control units**
- **I/O devices**
- and other objects

---

## Making Connectivity Changes

With I/O operations you can easily perform the following connectivity changes:

- **Remove and restore channel paths:** These commands not only instruct z/OS to reconfigure a CHPID, they also issue the appropriate vary path requests. In these commands, you can specify any CHPID that is defined to one of the hosts in the issuing host's group. The CHPID can be either switchable or point-to-point, ESCON/FICON or parallel.
- **Remove and restore devices:** These commands allow you to remove one, a range, or an array of devices from - and restore them to - one or an array of hosts in a group. These commands therefore expand the capabilities of the analogous Vary Device system commands, so that devices can be varied across system images—all within the issuing host's group.
- **Use switching commands, such as:**
  - Commands that affect only ports
  - Remove Switch
  - Restore Switch
  - Commands that define a chained path and remove these definitions

---

## Display Commands To Obtain Data About I/O Resources

With such commands, you can for example, retrieve data about one or all channels (CHPIDs) defined to the host image that the command was issued on. Or you can retrieve data about one, a range, or all the devices and their paths defined to the host image. You can also display data about the addressable port(s) on one or all switches allocated, or attached, to the issuing host.

Examples for this category of commands are:

- DISPLAY CHP
- DISPLAY DEVICE
- DISPLAY PORT
- DISPLAY RESULTS

---

## Connectivity Commands That Affect a Switch

With commands of this category you can for example permit or prohibit dynamic communication between two specified ports or between one port and all other ports on the same switch or the same ports on all the switches allocated to, or attached to, the issuing host.



## Connectivity Commands That Affect a Switch

Or, you can establish or remove a static, or dedicated, connection between two ports on the same switch or the same two ports on all the switches allocated to, or attached to, the issuing host.

Examples for this category of commands are:

- ALLOW
- PROHIBIT
- CONNECT
- DISCONNECT
- BLOCK
- UNBLOCK
- CHAIN
- UNCHAIN

---

## Display and Utility Commands Related to Connectivity Commands

With commands of this category you can for example display the specified number or all of failed vary path requests as a result of the last connectivity command.

Or you can build a list of the vary path requests that would be needed to synchronize path status with the physical settings of the specified switch.

You can also display or reset the number of seconds that the issuing hosts (I/O operations) will wait to receive votes from the I/O operations that from its voting set.

Examples for this category of commands are:

- DISPLAY VARY
- SYNC SWITCH
- RESET HOST
- DISPLAY TIMEOUT
- RESET TIMEOUT

---

## Utility Commands for General Processing

With such commands, you can for example, acquire or release exclusive control of I/O operations command processing for a specified user ID.

Or you can instruct the switch(es) to send their log data to the host that the issuing I/O operations is running on.

Another usage of such a command would be to ensure that the issuing I/O operations's configuration map is correlated with the active configuration on the switch(es).

Examples for this category of commands are:

- GETLOCK
- UNLOCK
- LOGREC
- RESET SWITCH

### Using the I/O Operations ISPF Dialog

A set of REXX EXECs is shipped with SA z/OS that – when installed as the I/O operations Interactive System Productivity Facility (ISPF) dialog – offers you a simple way to enter the same commands to I/O operations that you can enter at an z/OS operator console.

In addition to entering I/O operations commands, you can create entire switch configurations, starting with the default matrix of switch ports that is shipped with the dialog.

Detailed information about the I/O operations ISPF dialog is included in Appendix B, “Using the I/O Operations ISPF Dialog,” on page 223. The information in this section helps you get started using the dialog. For more information, refer to the online tutorial and online help panels supplied with the dialog.

---

### Using the z/OS Operator Console for I/O Operations Commands

Some I/O operations commands can be entered at the z/OS operator console. To enter a command in the z/OS environment, use the **Modify** command, which can be shortened to F. Specify the I/O operations procedure name that has been created as a member in your SYS1.PROCLIB. The member name for the procedure as supplied by SA z/OS is *INGEIO*.

- If you did not append an z/OS identifier to the Start command when I/O operations was started, the syntax is as follows:

```
/modiFy procname,I/O_operations_command
```

- If you did append an z/OS identifier to the Start command when I/O operations was started, the syntax is as follows. (NOTE: if the SA z/OS I/O operations component was started automatically by the automation supplied with SA z/OS, an identifier of "IO" was used by the Start command.)

```
/modiFy identifier,I/O_operations_command
```

For example, */F IO,D N CU.C00\* \** instructs I/O operations to display names of, and salient data about, the port or ports that have been assigned names with the common character string 'CU.C00' on the dynamic switches allocated to the issuing instance of I/O operations.

The NetView program provides an z/OS interface that allows a NetView REXX EXEC or an operator at a NetView console to issue z/OS operator commands. For example, the following Display Switch command can be coded in a REXX EXEC or issued at a NetView console:

```
MVS /F IO,D S *
```

(The output of this Display Switch command is displayed on the NetView console, not the z/OS operator console.)

When you use the NetView console to enter I/O operations commands, keep the following points in mind:

- Control authorization is not checked. The same I/O operations commands can be entered that can be entered at the z/OS operator console.

## Using the z/OS Operator Console for I/O Operations Commands

- Commands can be entered from different consoles to the same system operator ID. You should, therefore, have a central point of control for I/O operations processing, especially for its connectivity and utility commands.

## Using the z/OS Operator Console for I/O Operations Commands

---

## Chapter 10. SA z/OS Discovery Library Adapter

The SA z/OS Discovery Library Adapter (DLA) discovers automated resources and their relationships within a sysplex (that is, the scope of the automation manager) and generates output XML files. These files, often referred to as Books, conform to the Discovery Library IdML XML schema and Common Data Model (CDM). The Books can be loaded by various products, including Tivoli Change and Configuration Management Database (CCMDB) and Tivoli Business Service Management (TBSM) products.

---

### Overview

The SA z/OS Discovery Library Adapter (DLA), as shown in Figure 44, discovers SA z/OS configuration data, which is primarily SA z/OS resources and their relationships, for all systems that are controlled by the automation manager within a sysplex (including systems that are running a previous version of SA z/OS). It discovers this data for the following resources:

- SA z/OS subsystems (APLs)
- Application Groups (APGs)
- Monitor resources (MTRs)

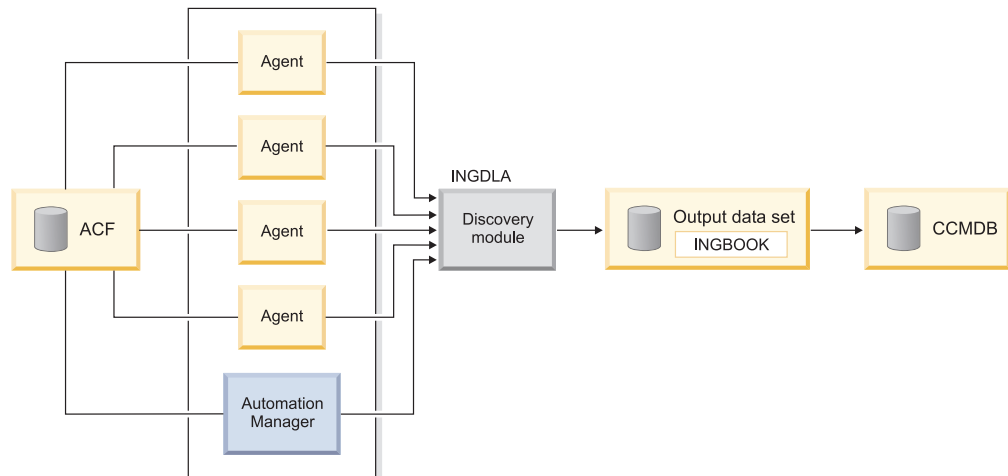


Figure 44. SA z/OS Discovery Library Adapter

Discovery runs in NetView via the INGDLA command issued from a system running SA z/OS 3.2. The command can be invoked either at will by an operator, for example, after a major configuration change, or it can be performed periodically (for example, daily or weekly), driven by a NetView timer.

The process flow of the SA z/OS DLA is as follows:

1. The INGDLA command queries the automation manager to obtain the list of automated resources (APLs, APGs, and MTRs) and their relationships.
2. Next, the command makes trips to the various automation agent to obtain information about the APL resources.
3. This data is then used to compose the IdML book that describes the SA z/OS configuration.

4. The IdML book is then written into the output data set (along with a few other members that control the generation of the book).
5. The IdML book is then downloaded to a system running Tivoli Application Dependency Discovery Manager (TADDM), where it can be loaded into CCMDB, along with data from other D:LAs.

The SA z/OS DLA is delivered with APAR OA23508 and must be installed on a system that is running SA z/OS 3.2. It must also be run in conjunction with the z/OS DLA, which provides data about operating systems and LPARs.

---

## Using the SA z/OS Discovery Library Adapter

Once you have installed the SA z/OS DLA, you can configure it using the INGCNTL command. This allows you to set the following defaults:

- The output data set name to contain the iDML book. It must be a pre-allocated (catalogued) PDS with attribute VB=3000. NetView must have UPDATE access to it.
- The name of the member that contains the iDML data (the default is INGBOOK).
- The name of the organization.
- The codepage that is used by NetView.

For more details, see *IBM Tivoli System Automation for z/OS Programmer's Reference*.

To discover your configuration data with the SA z/OS DLA, issue the INGDLA command from the NetView command line on any system in the sysplex. You can, for example, issue the command after major configuration changes or periodically, using NetView timers. See *IBM Tivoli System Automation for z/OS Operator's Commands* for more details.

Once you have run the SA z/OS DLA, you will need to transfer the IdML book (that is, the member that contains the book) to a system that is running Tivoli Application Dependency Discovery Manager (TADDM). The member is generated in UTF-8 format, and can be downloaded in binary format using FTP.

It is recommended that the name of the file on the TADDM system begins with ING320 (this is also referred to as the AppCode). The AppCode is used for naming IdML books, so that DLA administrators can easily sort books by product.

---

## Chapter 11. VM Second Level Systems Support

This feature provides processor operations support to control and monitor guest machines running under VM.

Processor operations allows an operating system to be IPLed into a processor, amongst other facilities. One such operating system is VM. Other operating systems can be IPLed within VM as guest machines. Of particular interest are LINUX guest machines, but MVS, VSE and even VM guest machines may be possible. (Lower levels of guest machines are not considered). Previously there was no effective way to enter commands to and receive messages from such a guest target system in order to validate that it had IPLed correctly, or that it is behaving correctly.

With second level guest machine support you can:

- Capture messages issued by the guest machine itself and route these back to the ProcOps process for display or automated processing, or both
- Send commands to the guest machine from ProcOps, either as operator requests or automated actions

---

### ProcOps Dialog

The ProcOps dialogs flow provides you with information about the guest systems hierarchy (see Figure 45):

- Hosting systems are flagged on the Status Summary panel ISQESUM.
- The PSMs of a hosting target system can be displayed by selecting the Target System Summary panel ISQETARG and the following PSMs panel ISQETRGG. The Target Hardware Summary panel for the PSM shows the initialized guest systems per PSM.
- The hosting target system of a guest system can be displayed by selecting the Target Hardware Summary panel ISQEHARP.

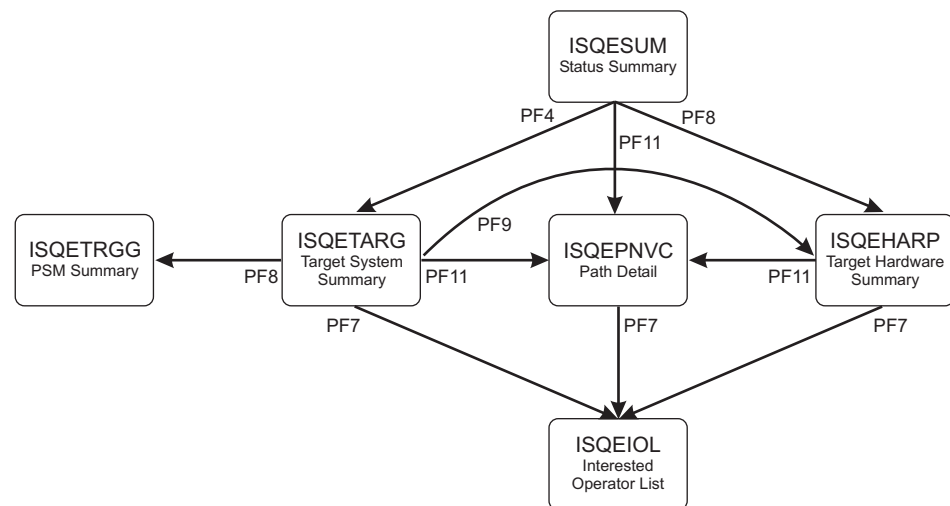


Figure 45. ProcOps Panel Flow

## Status Summary Panel (ISQESUM)

```

ISQESUM      SA z/OS Proc-Ops Target Status Summary      Updates: Dynamic
Control file in use: SATEST.OA02311.ISQCNTL
NMC Bridge: INACTIVE
Debug Mode: OFF

I isqxiii C isqxcls 0 isqxopt  A,B view netlogs E events

Cmd   Target System   Status   Focal Points - Primary: IPSFO Backup: IPUFM
BOEVMRAP (*)  UNKNOWN
CFA      CLOSED
CFB      INITIALIZED
CFF      INITIALIZED
CF1      INITIALIZED
CF61     INITIALIZED
CF62     INITIALIZED
KEYA     INITIALIZED
KEYB     INITIALIZED
KEYC     NOT ACTIVE
KEYZ     NOT OPERATING
KEY1     INITIALIZED
KEY2     INITIALIZED
KEY3     CLOSED
KEY4     INITIALIZED
KEY6     INITIALIZED
KEY6B    UNKNOWN
KEY7     NOT OPERATING
LNXOLI1  CLOSED
TEL01FE  CLOSED
TEL17FE  UNKNOWN
TEL30FE  UNKNOWN

Enter=Static PF1=Help PF3=Exit PF4=Tgt Sys Summary PF5=Debug On/Off
PF6=Roll    PF7=Up   PF8=Down PF9=Tgt HW Summary PF11=PATH Details PF12=Quit

```

Figure 46. Status Summary Panel ISQESUM

Targets that host other target systems are marked with (\*) that is *hosting system*.

## Target Hardware Summary Panel (ISQEHARP)

```

ISQEHARP      SA z/OS Proc-Ops THW Summary      Updates: Dynamic

Target Hardware Name: YORAMA      Type: PSM
VM Host System Name: VM123
Target Hardware Description: ProcOps Service Machine for KEYn systems
Initialized Target Systems:
KEY6      KEY7      KEY8

Communication Path Error Status:

Last Significant Message:

Enter=Static PF1=Help PF3=Exit PF6=Roll PF7=Oper1st PF11=PATH DtlS PF12=Quit

```

Figure 47. Target Hardware Summary Panel for PSMs ISQEHARP



```

ISQEHARP          SA z/OS Proc-Ops THW Summary          Updates: Dynamic
Target Hardware Name: YORAMA      Type: eServer      Mode: LPAR
                                   Model: RX4             Serial Number: 51-0064523
Target Hardware Description: Host for VM systems
Initialized Target Systems:
SYS1      (*) SYS2      SYS3

Communication Path Error Status:

Last Significant Message:

Enter=Static PF1=Help PF3=Exit PF6=Roll PF7=Oper1st PF11=PATH DtlS PF12=Quit

```

Figure 48. Target Hardware Summary Panel for Non-PSMs ISQEHARP

*Processor Type* PSM indicates that it is a ProcOps Service Machine and that the initialized target systems are Guest systems.

For PSMs:

- The Hosting System Name is displayed
- Mode, Model, and Serial Number are not displayed.

For real hardware:

- Initialized Target Systems that are hosting target systems are flagged with an (\*).

## Target System Summary Panel (ISQETARG)

```

ISQETARG          SA z/OS Proc-Ops Target System Summary      Updates: Dynamic

Target System
  Name           : KEY6____ (*)
  Description    :
  O.S.           : MVS
  Load Profile   :
  Status         : INITIALIZED
  Attention      : EXCEPTIONS

Target Hardware
  Name           : YORAMA           Mode : LPAR
  Attention      :
  Path Status    : ACTIVE

Target Hardware LPAR
  Name           : KEY6           Mode : ESA
  Image Profile  : KEY6

Last Significant Message:

Enter=Static PF1=Help PF3=Exit PF6=Roll PF7=Oper List PF8=Guests
PF9=Target Hardware PF11=Path Detail PF12=Quit

```

Figure 49. Target System Summary Panel ISQETARG

If the target system is a hosting system then:

- (\*) = Hosting system is displayed
- PF8=Guests is displayed. PF8 displays panel ISQETRGG.

## Guests Summary Panel (ISQETRGG)

```

ISQETRGG          SA z/OS Proc-Ops PSMs/Guests Summary      Updates: Dynamic

VM Host System Name:

Defined ProcOps Service Machines:
  PSM01          PSM02

Initialized Target Systems:
  PSM01/Linux1  PSM01/Linux2  PSM01/Linux3  PSM02/MVS1
  PSM02/VSE1

Enter=Static PF1=Help PF3=Exit PF6=Roll PF7=Up PF8=Down PF12=Quit

```

Figure 50. PSMs/Guests Summary Panel ISQETRGG

This panel displays the guest (that is, hosted) systems for a target system that is a hosting system. The guest systems are displayed for each associated PSM.

**VM Host System Name**

Displays the name of the hosting system.

**Defined ProcOps Service Machines**

Lists PSMs defined in the control file to run on the hosting system.

The number of PSMs per hosting VM system is limited to 12. (This is checked by the configuration dialog.)

**Initialized Target Systems**

Lists target systems currently associated with the hosting system. The target systems are identified by PSM/Target.

The number of guest systems being controlled by a PSM is limited to 60. (This is checked by the configuration dialog.) Therefore the total number of guest systems running on a VM host is limited to 720.

**PATH Detail Panel (ISQEPNVC)**

```

ISQEPNVC          SA z/OS Proc-Ops  PATH Detail          Updates: Dynamic

PSM Name          : SAFOS              Connection Type       : TCPIP

Message Server Port : 4711_            Command Server Port  : 4712_
IP Stack           : STACK---

PSM IP Address or Hostname:
999.999.999.999

Security Resource Name : DEIBMIP1.IP3T1200
Connection Authorization: PROCOPS      Lock Holder         :

PATH Status       :

Poll Frequency    : 40
Poll Retries      : 2

Last Significant Message:

Enter=Static      PF1=Help         PF3=Exit          PF6=Roll
PF7=Oper List     PF12=Quit

```

Figure 51. PATH Detail Panel ISQEPNVC

For PSM hardware TCPIP address, Message Server port, Command Server port, IP stack, and Security resource name are displayed.

**Security Resource Name**

Identifier used for SAF checking (for example, by RACF®).

**PSM Console Mode**

Normally the PSM runs disconnected. In this mode, its regular, unsolicited messages are not seen, even though they are recorded by the Logger thread in its log data set. To assist problem determination, these regular messages are captured by the Message Handler, queued to the Message Queue, and routed to the current ProcOps focal point by the Message Server. They should then be seen in the NetView log.

## PSM Console Mode

When a user that is logged on to the PSM issues a PSM request, a CP command or a CMS command, the responses are always displayed to the user. So, if a user logs on to the PSM no unsolicited messages are seen. These unsolicited messages can be made visible by issuing the command:

```
CP SET VMCONIO OFF
```

on the VM console. If the Message Handler is restarted for any reason (for example, when the first guest machine on the host is initialized using the ISQXIII command), the unsolicited messages to the VM console may be again lost. Reissue the CP SET command to restore message display at the VM console.

If an ISQTRACE command is issued that sets any thread trace to the ON state, the messages issued by the PSM are automatically displayed at the VM console, and not sent to NetView. This is to avoid possible message loops, because each message sent may cause additional messages to be created.

Note that if indentation of messages is active, the PSM messages returned to NetView also appear indented. If any PSM message spills to another line, that new line is a separate message sent to NetView.

If an ISQTRACE command is issued, and all traces are set to the OFF state, the flow of unsolicited messages is once again routed to NetView. This means that the effect of any earlier CP SET VMCONIO OFF command is then lost.

When messages are displayed on the VM console, they may also be spooled to a VM output file or a reader file. This includes any trace messages.

When a user is logged on to the PSM virtual machine, the following points should be noted.

- If messages are sent to the VM console they may cause the PSM to halt execution temporarily when or shortly after the screen fills with messages. Consider entering the CP TERMINAL MORE command to control the effect of this (possibly as a command in an initial EXEC that invokes the PSM Control Program). For example, issuing CP TERMINAL MORE 0 0 means that the VM console never locks up when regular messages are issued.
- Some messages are highlighted and held. For example, the XAUTOLOG command causes such messages. When the screen fills (even if CP TERMINAL MORE has been issued) the session goes into HOLDING mode and requires operator intervention.
- Any CP command can be executed by prefixing it with CP. Any CMS command can be executed by prefixing it with CMS. However any command is executed by the Command Server thread and serializes that thread. If such a command requires an extended period of time to execute it can prevent the Command Server thread from responding to requests. If a full screen command (such as XEDIT) is entered, all threads of the PSM Control Program stop executing until the user leaves the full screen command.

---

## Starting the PSM Control Program

The PSM Control Program runs in its own virtual machine. To start the PSM Control Program manually, logon to the virtual machine in the usual way. The PSM Control Program can be started by simply entering its program name, as selected by the programmer at installation time. For example, if the programmer named the program ISQPSM, simply enter this in response to a Ready prompt:

## Starting the PSM Control Program

```
Ready; T=0.04/0.05 19:52:14
isqpsm
ISQMA0001I PSM Main Thread entered
ISQMA0101I Thread 3 started as CSERV
ISQMA0002I Waiting for Command Server thread to initialize
ISQCS0710I Command Server thread entered
ISQCS0205I Command Server valid FP addresses are 9.164.172.63 9.164.156.189 9.99.99.99 66.77
ISQMA0101I Thread 4 started as CNSERVER
ISQMA0101I Thread 5 started as LOGGER
ISQCN0710I Console Server thread entered
ISQCN0711I Console Server thread initialized
ISQCS0711I Command Server thread initialized      ISQLG0710I Logger thread entered
ISQLG0711I Logger thread initialized
```

Initially PSM messages are displayed on the VM console. These may be captured to a disk file by spooling the console file to a reader file. To initiate spooling enter a CMS SPOOL command; for example:

```
spool console start *
```

Then start the PSM, as above. When you wish to create a complete file in the reader, enter a command:

```
spool console close
```

Spooling continues, and you may use the above spool console close command to create successive reader files. When spooling is to stop, enter the command:

```
spool console stop
```

(This does *not* write the remaining records to a reader file.)

If the above commands are to be entered while the PSM Control Program is running, the *spool* commands should be prefixed with *cms*:

```
cms spool console close
```

To automate the process of starting the PSM Control Program when the PSM virtual machine starts, add the ISQPSM command to the PROFILE EXEC of the PSM virtual machine.

On occasions, you may wish to start the PSM Control Program with traces active for threads that are started automatically. Before issuing the command to start the PSM, enter a CMS command of the following form:

```
GLOBALV ISQTrace.comp ON
```

Where *comp* is the trace option as used in the ISQTRACE request. (Usually one of CSERV, TCPIP, LOGGER or CNSERVER). Enter one command for each trace required.

---

## Stopping the PSM Control Program

To stop the PSM Control Program normally, first logon to the PSM virtual machine. Because the PSM virtual machine runs disconnected, the first action is to issue a BEGIN command to resume operation.

```
LOGON PSM
  ICH70001I PSM LAST ACCESS AT 17:08:54 ON MONDAY, OCTOBER 20, 2003
z/VM Version 4 Release 3.0, Service Level 0202 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES: 0038 RDR, 0003 PRT, NO PUN
RECONNECTED AT 20:25:10 CST WEDNESDAY 10/20/03
```

```
begin
```

## Stopping the PSM Control Program

Then enter the STOPALL request. The PSM control program should then close its threads.

After a short while (about ten seconds) the CMS prompt should be given:

```
stopall
ISQCN0402I Command <stopall> entered
HCPMFS057I PSM not receiving; not authorized
ISQCS0218I Request <stopall> done.
ISQCN0708I Console Server thread canceled
ISQCN0709I Console Server thread ending...
ISQCS0708I Command Server thread canceled
ISQLG0708I Logger thread canceled
ISQCS0709I Command Server thread ending...
ISQLG0709I Logger thread ending...
ISQMA0005I Main Thread ending (others may still be running)
Ready; T=0.09/0.10 20:30:59
```

Sometimes it is not possible to use STOPALL to stop the PSM. In these cases, enter the HX command (halt execution) followed by B (BEGIN), repeatedly until the Ready prompt is obtained:

```
hx
CMS
b
DMSRXS1419E EventSignal failed for event CONNSOCK; RC=8 Reason=108
Ready; T=0.14/0.14 20:36:31
```

The DMSRXS1419E message is normal

---

## Restarting the PSM Control Program

After a PSM Control Program has stopped control returns to CMS and a CMS Ready prompt is issued to a logged on user. At this point it is advisable to enter a HX command to terminate any running threads.

The PSM Control Program can be restarted by entering its program name, for example, ISQPSM.

When the program restarts any trace setting that were set from the last execution are retained and are still in effect. If you wish to set off a trace before resuming execution enter the CMS command:

```
GLOBALV ISQTrace.comp OFF
```

Where *comp* is the trace options as used in the ISQTRACE request. Enter such a command for each trace that is to be set off.

---

## Chapter 12. LPAR Management Functions

The two SA z/OS hardware interfaces, BCP internal interface (BCPii) and Processor Operations (ProcOps), offer a common set of hardware commands to manage and control the logical partitions of your System z™ and zSeries processor hardware. In addition, the management of processor activation profiles and querying of CPC and LPAR information is available.

In previous SA z/OS releases, only ProcOps users could exploit this functionality in their own automation scripts with the fully documented ISQCCMD programming interface. GDPS® or SA z/OS Parallel Sysplex Automation customers were limited to the subset of hardware commands that were externalized by these applications.

---

### Functional Overview

This section describes the following functions:

- “LPAR Management”
- “Activation Profile Management”
- “CPC and LPAR Information Queries”

For a detailed description of the ISQCCMD common commands listed in the following sections, see *IBM Tivoli System Automation for z/OS Operator's Commands*.

### LPAR Management

This function allows you to query and set the following:

- The defined LPAR capacity
- The current LPAR minimum and maximum weights
- The LPAR weight capping Enabled flag
- The Workload Management Enabled flag
- The name of a Capacity Group Profile

The ISQCCMD common command ICNTL provides these functions.

### Activation Profile Management

This function allows you to query and set values of the following CPC activation profiles:

- Reset
- Image
- Load
- Group

The ISQCCMD common command PROFILE provides these functions.

### CPC and LPAR Information Queries

This collection of queries allows you to obtain information about:

- Consolidated CPC and LPAR details for all configured partitions
- CPC type, model, serial number, mode, status, or degraded mode details

- CPC image (LPAR) name, mode, status, active operating system type and version
- CPC neighborhood information

The following ISQCCMD common commands provide this information:

- CPCDATA
- GETSSTAT
- GETSINFO
- GETISTAT
- GETIINFO
- GETCLUSTER
- GETSDGR

---

## Enabling the Functions

Before you can enable the LPAR management functions, make sure all hardware interface-specific installation and customization steps are complete:

- For ProcOps SNMP connections, see “Understanding the Processor Operations SNMP Interface” and “Step 8: Preparing the Hardware” in *IBM Tivoli System Automation for z/OS Planning and Installation*.
- For BCPii connections, see “Understanding the BCP Internal Interface” and “Step 8: Preparing the Hardware” in *IBM Tivoli System Automation for z/OS Planning and Installation*.

## Processor Operations

The common command sets are implemented with the ISQCCMD command, the ProcOps common hardware command interface.

If you are a ProcOps user, nothing extra needs to be enabled. You must however do the following:

1. Define your processor hardware and LPARs to use the SNMP connection type and make ProcOps-specific definitions for each of your systems in the SA z/OS policy database. Select the correct processor LPAR for each system.
2. Add PROCOPS to the TOWER.SA statement in the NetView style sheet, for example:

```
TOWER.SA = SYSOPS PROCOPS
```

## BCP Internal Interface

With the BCP internal interface, you use either GDPS or the SA z/OS Parallel Sysplex enhancement functions, or both. You must do the following:

1. Define your processor hardware and LPARs to use the INTERNAL connection type.
2. Make sure each of your systems in the SA z/OS policy database has the correct processor and LPAR.
3. In a sysplex environment or system cluster, you also need to make sure that all systems that need to communicate with each other over the BCPii are members of the same SYSPLEX or STANDARD group (or both). You must therefore define corresponding SA z/OS groups of type SYSPLEX or STANDARD, or both.
4. If you want to enable the LPAR Management function, add PROCOPS to the TOWER.SA statement in the NetView style sheet, for example:

```
TOWER.SA = SYSOPS PROCOPS
```



No other ProcOps-specific definitions are needed

You do not need to start ProcOps to use the ISQCCMD command for LPAR management functions over the BCPii.

---

## Usage Considerations

Bear in mind the following:

- If you activated a ProcOps FP environment for SNMP-connected processor hardware without enabling the BCPii for GDPS or the Parallel Sysplex<sup>®</sup> enhancement applications, you can use the ISQCCMD LPAR management functions like any other ProcOps commands.
- If you have defined and activated a BCPii environment for GDPS or the Parallel Sysplex enhancements, without defining and starting ProcOps, you can use the ISQCCMD LPAR management functions that are documented in this chapter, but no other ProcOps commands or functions. As the only prerequisite, the PROCOPS SA z/OS subtower must be enabled.
- If you have defined both the ProcOps FP and the BCPii on the same NetView, operation in parallel is supported. The following rule applies:
  - If a ProcOps or LPAR management command that is valid for both connection types is entered with a processor name or target system name as a parameter, the ProcOps resource tables are searched first. If the resource name cannot be found in the ProcOps tables, the BCPii tables are then scanned to resolve the name.

As a result, if you use a ProcOps control file that contains the same processors and systems names as the processors and systems in your ACF fragments that were built for BCPii (INTERNAL) connections, the ProcOps path is used for your LPAR management commands.

You can avoid such conflicts by using different policy database entry names for processors and systems in ProcOps than those you use with a BCPii (INTERNAL) connection.



---

## Part 3. Using SA z/OS on Workstations

This part describes how to use SA z/OS on a workstations. It contains the following:

- Chapter 13, "Using the NetView Management Console for SA z/OS," on page 179



---

## Chapter 13. Using the NetView Management Console for SA z/OS

The NetView Management Console (NMC) is a part of the NetView product and is used for monitoring network resources. As SA z/OS is based on NetView, it also exploits the graphical capabilities of the NMC.

You can monitor and control your whole enterprise with the NMC. It is also possible to monitor your enterprise from a single workstation (single point of control). This is valid for system operations as well as for processor operations. For general information about the NMC see *NetView Management Console User's Guide*. This chapter only describes the extensions that have been added for monitoring SA z/OS resources.

---

### Introduction

The communication between SA z/OS and the NMC is maintained by the SA z/OS topology manager. An SA z/OS topology agent on each target system retrieves the enterprise data from the automation manager. The SA z/OS topology manager on the focal point provides the information into RODM. GMFHS takes the information from RODM and presents it in a graphical form on the NMC workstation. The following sketches the main tasks of the SA z/OS topology manager for system operations. For processor operations, the mode of operation is similar.

At initialization time, the SA z/OS topology manager retrieves SA z/OS information for one or more sysplexes from the respective automation managers. To do that, it contacts an SA z/OS topology agent that resides on a system within the sysplex to be monitored. The SA z/OS topology agent obtains the information from the automation manager of the sysplex and passes it to the SA z/OS topology manager.

The retrieved information includes all resources, for example, applications (type APL), application groups (type APG), systems (SYS), and groups of systems (GRP) with their mutual dependencies, and relationships. The SA z/OS topology manager stores these resources together with their dependency information within RODM.

The SA z/OS topology manager calls BLDVIEWS to build the views that are necessary for these resources to be displayed on the NMC. The definitions for the views must be supplied by the users.

This process is illustrated in Figure 52 on page 180. The arrows with broken lines represent requests, those with solid lines stand for delivery of information. This figure shows an idealized system configuration. Any constellation of systems is supported according to the abilities of SA z/OS. For example, all components can be installed on a single system.

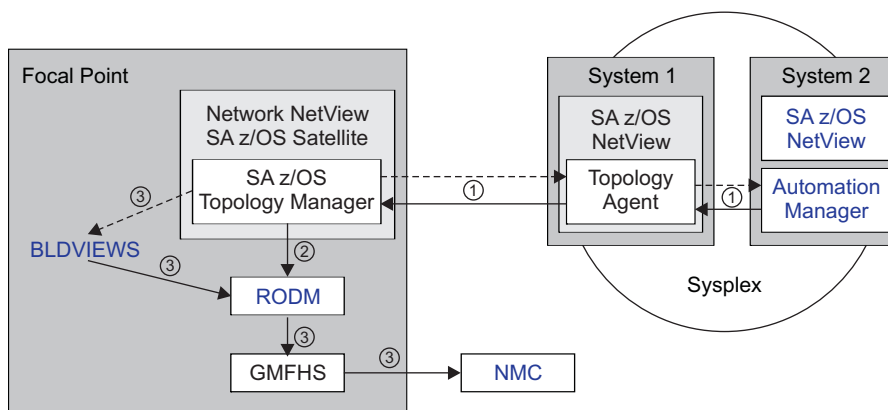


Figure 52. SA z/OS-NMC Communication Overview

**Note:** In SA z/OS, a single system can be regarded as a special sysplex with one system only.

At runtime, the SA z/OS topology manager performs the following tasks:

- It updates the NMC whenever the status of a displayed resource changes.
- It dynamically displays events like WTORs or tape mount requests on the NMC.
- Users can issue SA z/OS commands from the NMC that will be routed by SA z/OS to the appropriate system for processing.
- It resynchronizes its resource structure for one or more sysplexes with the respective automation managers. When this happens, the views affected by possible changes are rebuilt.

## Multiple Focal Point Support

Typically two NMC focal points are used to survey the NMC targets. Each focal point is the backup of the other one. They always show the same information about the targets (hot backup). Both focal points have to be configured within the SA z/OS customization dialogs (> 2 GRP > NMC DEFINITIONS) in the **Primary** field of the NMC Focal Point Domains section.

Separate, independent heartbeats are required for multiple NMC focal points. Each focal point needs its own heartbeat (a heartbeat timer on the target system and deadman timer on the focal point system).

## Starting the NMC

Before you start the NMC, make sure that the following conditions are satisfied:

- The NMC server must be up and running and must be communicating with NetView.
- In order to issue interactive commands (see “Interactive Commands” on page 189), you must have set up your NetView 3270 Management Console correctly. See *IBM Tivoli System Automation for z/OS Planning and Installation* for details.

After you have started the NMC, the **Sign On** panel is displayed. Enter the user ID and the password in the respective fields. The user ID must be identical to that of an active NetView host session. In the **Host Machine** field you must enter the address of the NMC server. If you do not know the correct address, contact your system administrator.

When you have signed on successfully, the **Tivoli NetView for z/OS Management Console** is displayed. The left part of that window contains the *Business Tree*. If you select a view from the *Business Tree*, it is displayed in the right part. The SA z/OS views are descendants of the **Networking** view, as shown in Figure 53.

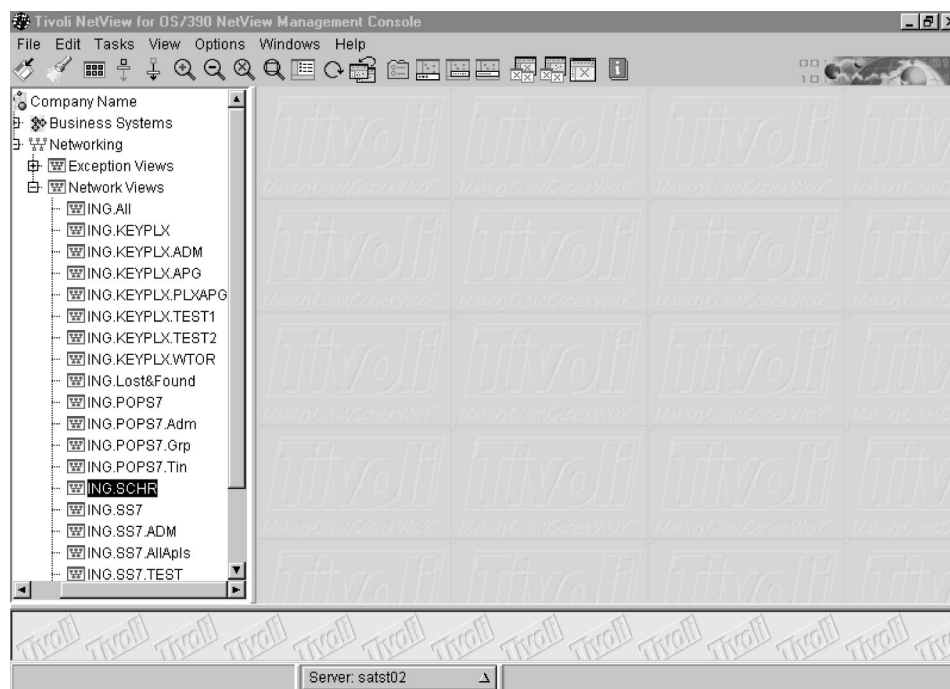


Figure 53. Tivoli NetView for z/OS Management Console

## Monitoring SA z/OS on the NMC

On the NMC workstation, you can see the graphical visualization of what is automated in system operations and processor operations.

For system operations, the automation manager knows about the configuration of the sysplex and the resources running in it. The SA z/OS topology manager communicates with these automation managers.

For processor operations, it is the processor operations focal point that knows about the processor operations configuration and therefore, the SA z/OS topology manager communicates with it.

Besides the resource information retrieved from the automation managers, the SA z/OS topology manager maintains SA z/OS objects that have no counterpart within the automation manager. The following sections explain the representation of these two types of SA z/OS resources on the NMC.

### Regular Resources

SA z/OS resources in RODM that also exist in the automation manager are called *regular*. For example, systems (SYS resource type), system groups (GRP), applications (APL), application groups (APG), and Monitor resources (MTR) are regular resources. Figure 54 on page 182 shows a network view of regular resources.

## Monitoring SA z/OS on the NMC

Note that the figures in this chapter are based on the system AOC7 and sysplex SS7.



Figure 54. Network View for Regular Resources

The members of a group are represented on the NMC by subviews. For instance, if you double click on the application group `SS7.AOC7.APG.CICSGRPJSHP` of Figure 54, the following view will be displayed:

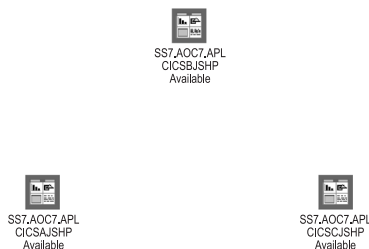


Figure 55. Subview with Applications of an Application Group

## Minor Resources, Anchors and Tokens

The SA z/OS topology manager distinguishes between *major* and *minor* resources. Major resources, such as applications or systems, exist independently. Minor resources are subcomponents of applications (or MVS components) that cannot exist independently, but are always associated with a major resource. Typical examples for minor resources would be the spool of a JES application or transactions of a CICS application. Minor resources have their own automation flags so that you can, for example, switch on automation for an application, but switch it off for one of its subcomponents. This is one main reason for introducing the distinction between major and minor resources.

The minor resources that are defined in the policy database and known to SA z/OS are a special type of regular resources. They are retrieved by the SA z/OS topology manager from an automation manager (either during initialization of the SA z/OS topology manager or when the information in RODM must be resynchronized with the automation manager).

In addition to regular minor resources, the SA z/OS topology manager also knows *dynamic* minor resources. Dynamic minor resources are posted to the SA z/OS



topology manager at runtime with the INGPOST command. You can also change the state of an existing dynamic resource with INGPOST. For more information on the INGPOST command, see *IBM Tivoli System Automation for z/OS Programmer's Reference*. In contrast to regular resources, you can delete dynamic minor resources in RODM from the NMC.

To post a dynamic minor resource to the SA z/OS topology manager, you must specify the major resource that this minor resource belongs to. When the SA z/OS topology manager finds the major resource within RODM it connects the new minor resource with the major resource and displays it immediately in all (network) views that contain the major resource.

The major resource specified to INGPOST can be a regular resource or an *anchor*. Anchors are special objects that exist in RODM, but not in the automation manager. They play the role of major resources and serve two purposes:

- They are used to display dynamic minor resources that either the specified major resource or the specified sysplex does not exist for in RODM. See “‘Lost&Found’-Anchors” on page 185.
- Anchors are used to display and group minor resources that derive from events such as WTORS and tape mount requests.

### Anchors and Events

The SA z/OS topology manager is able to represent events such as WTORS or tape mount requests on the NMC. The events are posted to it with the INGPOST command. The SA z/OS topology manager treats them as dynamic minor resources; these resources are called *event resources*. Event resources are connected to an anchor as their major resource. All events connected to one anchor belong to the same event type. Thus there are, for example, anchors for WTORS, and anchors for tape mount requests.

If specified in the Customization Dialog the SA z/OS topology manager creates one anchor per sysplex for several event types, for example for WTORS and tape mount requests. Whenever an event of such a type is posted to the SA z/OS topology manager, it is attached to the appropriate anchor of the respective sysplex, and all views that include this anchor are immediately updated.

Thus, when a WTOR originating from system AOC7 is posted to the SA z/OS topology manager and AOC7 belongs to sysplex SS7, this WTOR is attached as a minor resource to the (generated) WTOR anchor of sysplex SS7 and dynamically added to any view that includes the WTOR anchor of SS7.

Besides this you can define your own anchors for any events in the customization dialogs. You must post these events to the SA z/OS topology manager with the INGPOST command.

Figure 56 on page 184 shows a view that contains WTOR, TAPE, and SYSPLEX anchors for the sysplex SS7 with event resources attached to some of them. It also shows the HEARTBEAT anchor that is automatically created.

## Monitoring SA z/OS on the NMC

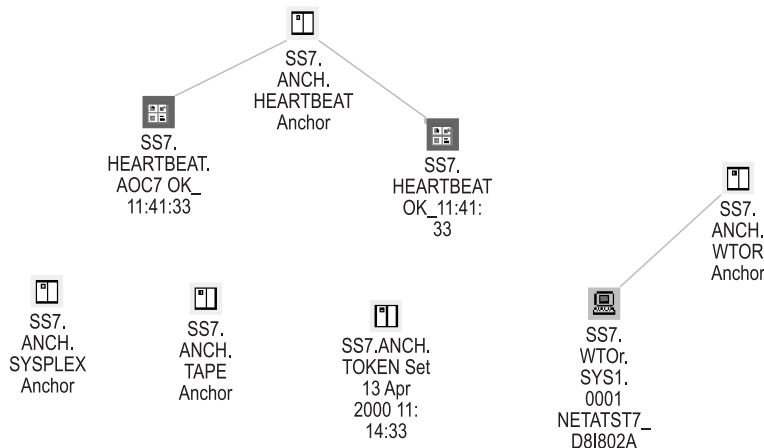


Figure 56. Anchor View

The anchors and their minor resources have the following function:

### Heartbeat

These resources represent the connection between the SA z/OS topology manager and the sysplex. Usually, two minor resources are attached to the heartbeat anchor. One icon represents the heartbeat for the sysplex (the right one in Figure 56), the other is the heartbeat for the system that is currently being used to monitor the sysplex (the left one in Figure 56).

For more information on heartbeats, see *IBM Tivoli System Automation for z/OS Defining Automation Policy*.

**Note:** When the communication between the SA z/OS topology manager and the target system(s) fails, icons are created to represent the status: you will find a heartbeat icon with a corresponding label, such as *APLX.HEARTBEAT.NOTARGET* and a sysplex group labelled for example, *APLX.GRP.NoTarget NoCommunication*. The heartbeat icon is there to represent the state. On the sysplex group you can initiate an `INGTOPO REBUILD <system>` or an `INGTOPO SWEEPSYS <system>`.

### SYSPLEX

Minor resources like coupling facilities, couple data sets or sysplex timers are attached to the SYSPLEX anchor if they have been defined in the customization dialogs.

Additionally, all minor resources that are not related to a specific major resource are attached to the SYSPLEX anchor.

### WTOR

Write To Operator with Reply (WTOR) messages are z/OS messages that inform you of various conditions and require a reply of the operator. In Figure 56, one WTOR is attached to the WTOR anchor.

In the label of this WTOR, 'WTOR' is the major resource name, and 'SYS1.0001' the minor resource name. The additional information is the message content.

By defining a view that contains the WTOR anchors of all sysplexes in the enterprise, you can monitor all WTORS at a single glance. For defining views, see "Creating Views" on page 192.

**Tape mount request**

When a tape mount request for sysplex SS7 is posted to the SA z/OS topology manager, it is attached to the tape mount request anchor of this sysplex.

**'Lost&Found'-Anchors**

To post a new minor resource to the SA z/OS topology manager, you must specify the major resource that this minor resource belongs to. When the SA z/OS topology manager cannot find the major resource in RODM, it attaches the minor resource to one of several special anchors that serve to display such unconnected minor resources.

There is one such anchor at sysplex level, which collects minor resources whose major resource does not exist in RODM, but whose sysplex is known to the SA z/OS topology manager. If even the sysplex is unknown, the SA z/OS topology manager attaches the minor resource to an anchor with the prefix 'Lost&Found'. There are 'Lost&Found'-anchors for WTORs, tape mount requests, heartbeats, and for sysplex related resources.

Figure 57 displays a view that comprises the 'Lost&Found'-anchors.



Figure 57. View with 'Lost&Found'-Anchors

**Tokens**

A token is an object that is generated automatically by the SA z/OS topology manager. The date and time when the SA z/OS topology manager was initialized is appended to the name of the token. In the **Data1** field of the resource properties, you can find the original ACF token or the processor operations token of the monitored system. This can be used for diagnostic purposes.

**Properties of SA z/OS Objects on the NMC Workstation**

All resources are represented by icons. Dependencies are displayed as lines; no distinction is made between start and stop dependencies. Association of a minor resource to a major resource is also represented with a line. The state of the resources is indicated by the color of the icon. The colors are mapped to the compound states of the resource as follows:

GREEN	Satisfactory
LIME	InAutomation
CYAN	Awaiting
WHITE	Inhibited
YELLOW	Degraded
PINK	Denied
RED	Problem
GREY	Status not known (NMC specific)

## Monitoring SA z/OS on the NMC

BLACK	Resource deleted in RODM (NMC specific)
-------	---

For more information on the compound status, see “Compound Status” on page 39.

The labels of the icons consist of three (or four) parts:

1. The sysplex name as defined in the INGTOPOF file.

For more information about the INGTOPOF file, see *IBM Tivoli System Automation for z/OS Planning and Installation*.

2. A derivative of the major resource name.

This part of the label differs from the resource name within the automation manager in that the order of its components is reversed and the slashes are replaced by dots (for example, the standard resource name JES2/APL/SYS1 becomes SYS1.APL.JES2). The reason for this reversal is explained in “Names” on page 193.

3. Possibly a minor resource name.

The minor resource name of the JES2 spool in Figure 54 on page 182 is SPOOL. When a minor resource name contains more than one part, the separator slashes are also replaced by dots, but no reversal of the components takes place (for example, TRAN/ABC123 would become TRAN.ABC123).

4. Additional information.

With major resources, this additional information is essentially the compound status. In most cases, it is supplemented by a suffix that indicates whether the status refers to starting or stopping the resource. Thus there is an INHIBITED\_START state and an INHIBITED\_STOP state. The SATISFACTORY compound state is replaced by the pair AVAILABLE and UNAVAILABLE.

For the minor resource in Figure 54 on page 182, the additional information is 'OK'.

For more information on the labels, see “Names” on page 193.

If you have the appropriate CPE profile setup for your user ID, you can view and possibly edit the **Resource Properties** of the displayed resources (see also *NetView Management Console User's Guide*).

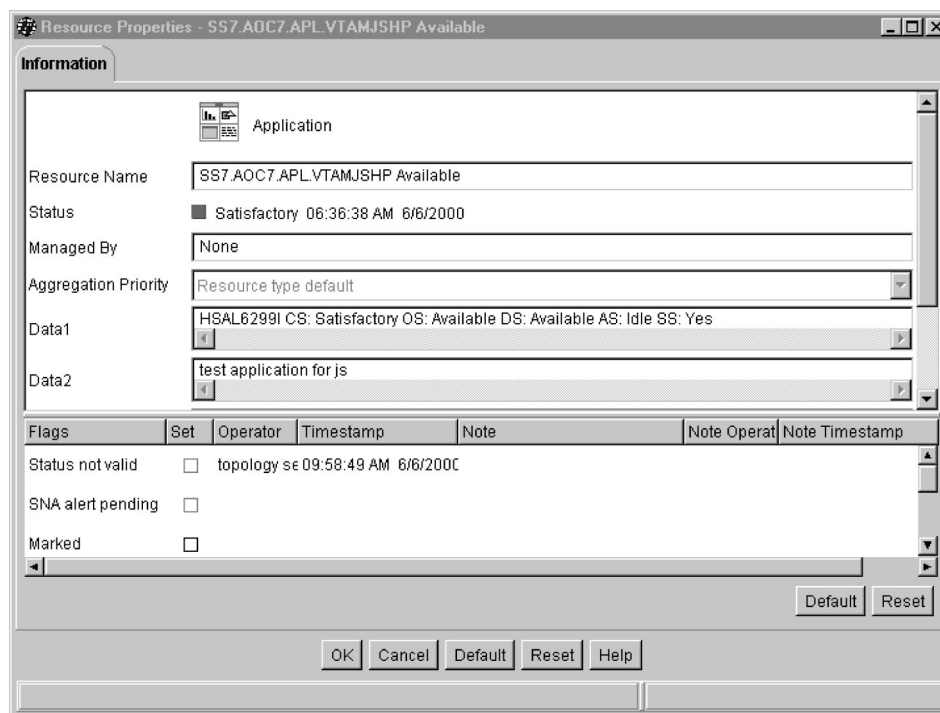


Figure 58. Resource Properties Window

For regular major resources, the **Data 1** field contains detailed status information, and the **Data 2** field can contain a comment, for example from an operator. For dynamic minor resources (see “Minor Resources, Anchors and Tokens” on page 182), the **Data 1** field corresponds to the message parameter of the INGPOST command, **Data 2** corresponds to the userdata parameter of the INGPOST command, and **Data 3** can be changed by using the D3 parameter of the INGPOST command; see *IBM Tivoli System Automation for z/OS Programmer’s Reference* for more information on INGPOST. **Data 4** is not currently used.

SA z/OS information is presented on the NMC in *network* views or in *exception* views. Network views represent the topology of resources, that is, grouping of resources and their connections. Figure 54 on page 182 is an example for a network view. Exception views display resources that have certain properties in common.

## Dynamic Behavior of the SA z/OS Topology Manager

In a running SA z/OS enterprise, you can imagine scenarios where new systems are IPLed while the SA z/OS topology manager is already running, or new systems are added to a sysplex and so on. The following sections describe how the SA z/OS topology manager behaves in these events.

### SA z/OS Topology Manager Initialization: INGTOPO INIT

If the systems are set up accordingly, SA z/OS automatically issues the INGTOPO INIT command after it has received the completion code of the start of the NetView MultiSystem Manager. If this is not the case, you have to start the SA z/OS topology manager manually by issuing the INGTOPO INIT command.

With the INGTOPO INIT call, the SA z/OS topology manager queries the INGTOPOF file and contacts each of the specified target systems (the specified

## Dynamic Behavior of the SA z/OS Topology Manager

sysplexes or the processor operations focal point). For the contacted target systems, it creates the corresponding objects in RODM.

### System Detection

When a system that SA z/OS is installed on becomes active, it sends a signal to the SA z/OS topology manager. The SA z/OS topology manager, which is already running, checks whether the system belongs to either a sysplex or a processor operations focal point it knows (that is, one that is defined in the INGTOPOF file) and whether this sysplex or processor operations focal point was incommunicative until now. If both conditions are fulfilled, the SA z/OS topology manager performs the initialization process for this sysplex or processor operations focal point. In all other cases the SA z/OS topology manager ignores the signal. Note that these signals never reach any operator when the SA z/OS topology manager is not present.

### The Heartbeat Function

When the SA z/OS topology manager makes contact with a system that can act as the representative of a sysplex, it launches a *heartbeat* timer on that system and a *deadman's* timer for that sysplex on the focal point system. The heartbeat timer sends information to the focal point system at a regular interval to indicate that the target system is still in good health. This causes the deadman's timer on the focal point to be reset. If the reset does not occur before the deadman's timer expires, the SA z/OS topology manager assumes that the current representative cannot respond. Processing in such a case is described in "Takeover when a System Goes Down."

### Takeover when a System Goes Down

If the SA z/OS topology manager needs to assume that the sysplex representative cannot respond, it tries to find another one according to the specified systems in the in the SYSPLEX statement in the INGTOPOF file. Therefore the sequence of the specified SYSPLEX and PROCOPS statements and the sequence of the specified systems is relevant. If it finds a new representative, communication continues with the new one and representation on the NMC continues. If it does not find one, it considers communication with the respective sysplex to be interrupted; all resources of the sysplex assume the color representation of the status *Unknown*.

---

## Issuing Commands from the NMC

In order to issue a command against a resource, click on this resource with the right mouse button to open a context menu. This menu contains only those system operations or processor operations commands that are applicable to the respective resource. Some SA z/OS commands are not provided because they cannot be reasonably connected to a resource.

When the state of the resource is affected by the command, the state information (event component of the label, color of the icon) is actualized on the NMC.

The following sections describe the different command types.

### Direct Commands

Some commands, for example INGINFO, need no user specified parameters. They are immediately executed after having been selected in a command submenu. Their output is displayed in the log of the NMC.

## Dialog Commands

Most SA z/OS commands accept or need parameters that you must specify. When you select such a command from a submenu, a window will open where you can enter the appropriate values. Their output is displayed in the log of the NMC.

If you select, for example, the command `INGTHRES` for the resource `SS7.AOC7.APL.VTAMJSHP` in the view of Figure 54 on page 182, the appropriate parameter window is opened as shown in Figure 59.

Figure 59. Parameter Window for `INGTHRES`

All parameter windows consist of two parts. The upper part contains the fields where you can enter the parameter values. The lower part displays the operator command that is generated from your input in the upper part. All you have to do is to enter the required values or choose the required option in the parameter fields (which has already been done in Figure 59) and issue the command by clicking on one of the two **send** buttons.

You can edit the command string in the **Command string** box at the bottom of the window. However, this is not necessary and may even lead to errors if you do not have a thorough understanding of the `INGRTCMD` command. “The `INGRTCMD` Command” on page 191 contains a brief summary of the `INGRTCMD` mechanism.

## Interactive Commands

There are commands that require a continued dialog before they can be processed. Such commands are routed to a NetView 3270 Management Console and are processed there; this presupposes that you have set up the 3270 console correctly.

The `INGREQ STOP` command provides an example:

After clicking on the resource `SS7.AOC7.APL.VTAMJSHP` in Figure 54 on page 182 and selecting the entry **SA Operations Commands**, a submenu is opened that

## Issuing Commands from the NMC

includes the **INGREQ Stop** entry. If you select this command, the request is forwarded to a NetView 3270 console, which would look as follows:

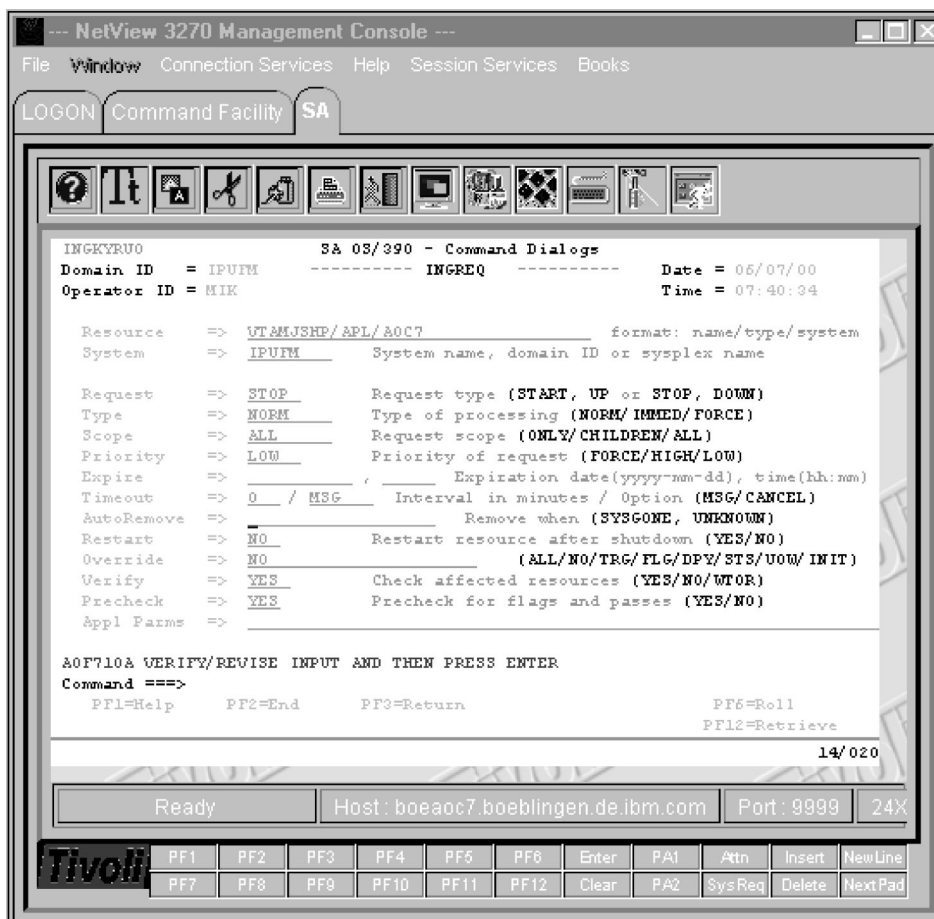


Figure 60. NetView Console with INGREQ Panel

After pressing Enter you are presented with a list of the resources affected by the command (SS7.AOC7.APL.VTAMJSHR and SS7.AOC7.APG.CICSGRPSHP) . If you confirm the request the resources are stopped. The view of Figure 54 on page 182 is actualized, and now looks as follows:





Figure 61. Actualized View

Note the modified event components in the label of the resources SS7.AOC7.APL.VTAMJSHP and SS7.AOC7.APG.CICSGRPJSHP.

**Note:** Depending on your setup, at the initial start of the NetView 3270 Management Console, triggered by an SA z/OS command that needs to be transferred to the NetView 3270 Management Console, you might be confronted with messages, or even problems (e.g. failing logon), that are not related to the original SA z/OS command. Handle these messages or problems as appropriate and retry the command.

## The INGRTCMD Command

When you select a resource and then choose an operator command, a template of that command is generated. In this template, a place holder value is specified for all parameters of the command that an attribute of the selected resource can be assigned to as its value. The place holder points to this attribute. The place holders all have the format *?name..*. Thus, for example, *?RSC.* refers to the resource name (which would be VTAMJSHP/APL/AOC7 in our example) of the object you clicked on.

Before the command is forwarded to the appropriate target system, these place holders must be replaced by real values. This is done by the command INGRTCMD. In order to be able to substitute the placeholders, INGRTCMD must be passed the ID of the object whose attributes are to supply the parameter values, so that the format of the command is as follows:

```
INGRTCMD object_id command_string
```

In the parameter window, the *object\_id* is specified in the form %OBJECTID%, which is a reference to the RODM object ID of the selected resource. The command string displayed in Figure 59 on page 189 would thus be converted by INGRTCMD into

```
INGTHRES VTAMJSHP/APL/AOC7,CRIT=(2,02:00) outmode=line
```

and would be routed to the system representing the sysplex SS7.

---

### Creating Views

The RODM objects representing SA z/OS resources must be included in a view in order to be displayed on the NMC. The definition of these views is up to you. The SA z/OS topology manager, however, offers some support for executing view definitions and updating the views whenever necessary. In order to exploit this feature, you must specify your view definitions in the format required by the NetView BLDVIEWS tool. Some familiarity with this tool and with the terminology and class structure of RODM (see *Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*) will be assumed in the following description.

The files containing your BLDVIEWS view definitions can be introduced to the SA z/OS topology manager in the INGTOPOF configuration file (for information on the INGTOPOF file, see *IBM Tivoli System Automation for z/OS Planning and Installation*). When the SA z/OS topology manager is started, it will, after constructing the SA z/OS objects, create the views in RODM by calling BLDVIEWS for all the definition files that it finds in the INGTOPOF file. BLDVIEWS will create a view object in RODM for every definition and connect every resource that meets the selection criteria specified in the definition with this view object.

**Note:** You can also call the BLDVIEWS tool independently of the SA z/OS topology manager to update views or create further views. However, these will not be automatically updated when new major resources are created in RODM.

### Network Views and Exception Views

NetView provides different types of views. For displaying SA z/OS resources, only *network views* and *exception views* are used.

Network views are intended to display the structure of a network. Therefore, all relationships between resources are displayed on the NMC. Furthermore, network views are static with regard to their structure; unless the link between a view element and the view object is removed or a new link is established, they always contain the same elements.

Exception views, on the other hand, display resources that have certain properties in common, for example, all resources that are in an exception state and marked by an operator. The relationships are not displayed in an exception view. Exception views are dynamic; although the underlying structure does not change, new elements are included in the view when a resource acquires the appropriate combination of properties (for example, when an operator marks a resource that is in an exception state), and elements are removed when they change any of their relevant properties (for example, when the resource is unmarked again).

A BLDVIEWS definition of a view typically consists of two parts. With the VIEW (for network views) or EVIEW (for exception views) statement you specify the type and properties of the view.

The second part of the definition specifies the resources that are to appear in the view. All SA z/OS resources in RODM are Non-SNA (GMFHS Managed Real) resources. Thus, you can use either the **MyName** or the **DisplayResourceName** field as a selection criterion for the Non-SNA resources to be included in the view. Note that you can specify more than one selection criterion.

The following are a typical network view and a typical exception view definition. First the network view:

```
VIEW=ING.SS7.APPL,ANNOTATION='Sysplex SS7'
NONSNA='SS7.AOC7.*',QUERYFIELD=DRN
```

The NONSNA statement must be read as "Include in the view all Non-SNA resources whose **DisplayResourceName** field contains a value beginning with SS7.AOC7.." For more information on view definitions for the BLDVIEWS tool see the online help that is shipped with NetView.

A typical exception view:

```
EVIEW=ING.SS7.EXC,ANNOTATION='Exceptions SS7',DSF=NOXCPT,MF=OFF
NONSNA='SS7.AOC7.APL.*',QUERYFIELD=DRN
```

Note the interplay between the selection criterion of the NONSNA statement and the filter parameters of the EVIEW statement. In contrast to the network view, the NONSNA statement for an exception view only specifies the *candidates* for appearance in the view. Which of these candidates are in fact included depends on the filter parameters. In the example definition, the exception state filter (DSF parameter) and the mark filter (MF parameter) are set so that only those candidate resources are displayed that are in an exception state and marked by an operator.

In order to be able to specify the selection criteria correctly, you must know how the names of the SA z/OS resources in the RODM **MyName** and **DisplayResourceName** fields are constructed. This is explained in the following section.

## Names

Every SA z/OS resource in RODM has two names: an *object name* (RODM **MyName** field) and a *display name* (RODM **DisplayResourceName** field). The object name identifies the object uniquely within RODM; the display name appears as the label of the icon that represents the object on the NMC. When defining a view to BLDVIEWS, you can use either of these names as a selection criterion for the objects to be included in the view.

Because the names for system operations and processor operations objects are somewhat different, they are described separately.

### Names for System Operations Objects

The object names for system operations objects have the following format:

```
sysplex_name.{name/type/system | name/type | ANCHOR/MJR/anchor_name}
[minor_res_name]
```

Here *sysplex\_name* is the name of a sysplex as defined by a SYSPLEX statement in the INGTOPOF file.

The display names have the following format:

```
sysplex_name.{system.type.name | type.name | anchor_name.MJR.ANCHOR}
[minor_res_name] [ev_summ]
```

Here again, *sysplex\_name* is the name of a sysplex as defined in the INGTOPOF file. Note that the order of the component names for major resources is reversed in the display name. This is done to provide more flexibility when specifying filter criteria for views. Note further that slashes (/) are replaced by dots in the names

## Creating Views

of minor resources. *ev\_summ* stands, in the case of a regular major resource, for the current compound status of the resource. With minor resources it stands for various information, for example for the message content of a WTOR or the timestamp of a heart beat.

The following are some examples for both name types:

1. Application TSO on system AOC7 in sysplex SS7:

**Object Name:**

SS7.TSO/APL/AOC7

**Display Name:**

SS7.AOC7.APL.TSO

2. Regular minor resource SPOOL of application JES2JSHP on AOC7 in sysplex SS7 in state OK:

**Object Name:**

SS7.JES2JSHP/APL/AOC7.SPOOL

**Display Name:**

SS7.AOC7.APL.JES2.SPOOL OK

3. WTOR outstanding on system AOC7 in sysplex SS7:

**Object Name:**

SS7.ANCHOR/MJR/WTOR.AOC7/0007

**Display Name:**

SS7.WTOR.MJR.ANCHOR.AOC7.0007 NETCTST7\_DSI802A

4. 3. Coupling facility on system AOC7 in sysplex SS7:

**Object Name:**

SS7.ANCHOR/MJR/SYSPLEX.CF/CF01/AOC7

**Display Name:**

SS7.SYSPLEX.MJR.ANCHOR.CF/CF01/AOC7 Connected

**Note:** The display name can be customized. For further information, refer to "Appendix B, Syntax for INGTOPOF File" in *IBM Tivoli System Automation for z/OS Planning and Installation*.

### Names for Processor Operations Objects

The object names for processor operations objects have the following format:

```
procops_name.name/type[/focal_point_system]/1st_level_min[/2nd_level_min  
[/3rd_level_min]]
```

Here *procops\_name* is the name of a focal point configuration as defined by a PROCOPS statement in the INGTOPOF file.

The display names have the following format:

```
procops_name.[focal_point_system.]type.name.1st_level_min[.2nd_level_min  
[.3rd_level_min]] [ev_summ]
```

Here again, *procops\_name* is the name of a focal point configuration as defined in the INGTOPOF file. As with the system operations names, the order of the major resource component names is reversed, and slashes (/) are replaced by dots. *ev\_summ* stands for the current processor operations state of the resource.

## SA z/OS Topology Manager Commands

There are two commands that you can use to communicate with the SA z/OS topology manager: INGTOPO and INGPOST. For more information on these commands, refer to *IBM Tivoli System Automation for z/OS Programmer's Reference* and *IBM Tivoli System Automation for z/OS Operator's Commands*.

### Processor Operations on the NMC

Processor operations deals with controlling operations of the processor hardware. One or more of these processors are controlled from a focal point. There can be more than one focal point. It is these focal points that the SA z/OS topology manager contacts for gathering information. A focal point with its associated target processors is called a processor operations focal point configuration or, if no confusion is likely to arise, simply a *focal point configuration*.

Processor operations is independent from system operations. Accordingly, when the same enterprise component (for example, a system) is integrated into processor operations and also into system operations, the SA z/OS topology manager creates two representations of it in RODM. For these reasons it will usually be advisable to keep the processor operations objects apart from the system operations objects in the NMC views. See "Creating Views" on page 192 for defining views.

The following figure shows a network view of a processor operations configuration.

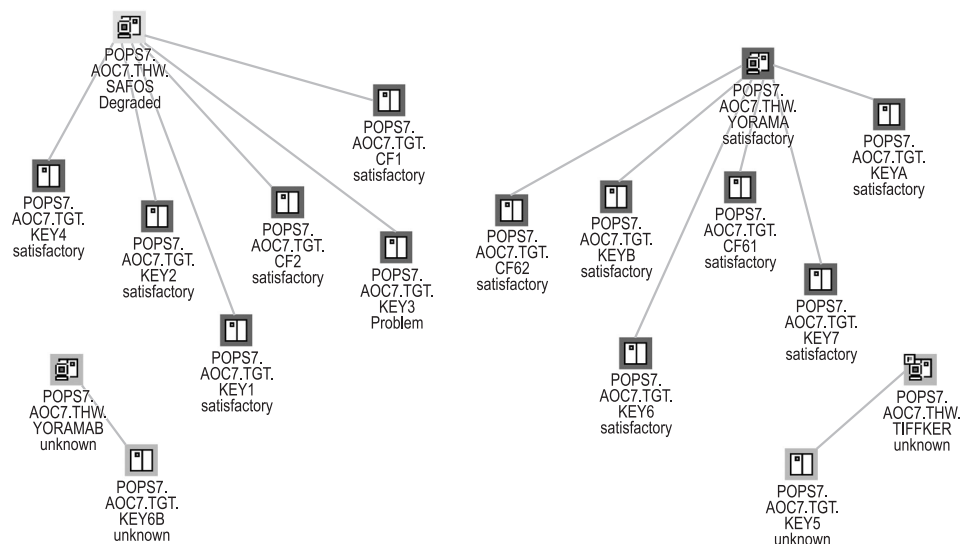


Figure 62. Processor Operations View

The SA z/OS topology manager knows four kinds of processor operations objects that can be distinguished by the type component of their labels. This label is constructed as follows:

1. The name of a focal point configuration as defined in the INGTOPOF file.  
For the INGTOPOF file, see *IBM Tivoli System Automation for z/OS Planning and Installation*.
2. The resource name.  
This consists of the name itself, the type identifier, and optionally a system name.

## Processor Operations on the NMC

There are four resource types, whose names are derived (with one exception) from the corresponding processor operations type names by omitting the prefix ISQ. The names have the following meanings:

- TGT** This characterizes a target system.
- THW** This characterizes a target processor.
- PGR** This characterizes a group of target systems.

**Note:** This type identifier is different from the second part of the corresponding processor operations type name, which is ISQGRP. The reason for this deviation is that GRP is already used within system operations.

**SBG** This characterizes a subgroup of target systems.

3. A minor resource name consisting of one, two or three parts. The parts are separated by dots.

The states of processor operations objects are interpreted in terms of NMC states. You can view the complete processor operations status in the **Data 2** field of the **Resource Properties** window.

---

## Monitoring Sysplex Resources

The SA z/OS topology manager detects and provides objects that represent sysplex resources (for example, coupling facilities and couple data sets). Also sysplex resource specific commands are provided so that you can issue them in the same way as described in “Issuing Commands from the NMC” on page 188.

---

## Problem Management

Problem management facilities offered by the NMC workstation consist of the following aspects described in the subsequent sections:

- “Exception Views”
- “Marked and Unmarked Resources” on page 197
- “Updating Operator Data” on page 197
- “Problem Management of Minor Resources” on page 197

These sections are an outline of suggestions how you can use the combination of these facilities to establish an efficient problem management.

### Exception Views

Exception views are typically used for problem management. It is recommended that you define your exception views in pairs, where the difference between the two views is that one of them contains only unmarked resources, and the other only marked resources. For example, the marked resources could be those being worked upon by an operator, the unmarked ones are not yet handled. The exception views are actualized dynamically; if you mark a resource within a view that contains only unmarked resources, the resource is immediately removed from that view and added to the corresponding view with marked resources.

The following view, for example, shows one of the resources also occurring in Figure 54 on page 182 that is in an exception state.



Figure 63. Exception View

For more information on network and exception views, see “Network Views and Exception Views” on page 192.

## Marked and Unmarked Resources

The feature of marked and unmarked resource as introduced in “Exception Views” on page 196 is provided for every resource via the resources command menu. The NMC workstation just lets you mark resources. Marked resources are displayed with a flag. You define within your enterprise what the meaning of the mark should be. So for example, you can distinguish resources in different states of attendance. Marked and unmarked resources dynamically change within the appropriately defined views according to your marking actions.

Markings processed through the SA z/OS marking commands (offered in the popup menu) can be persistently stored by SA z/OS. This is the advantage over the standard marking function as provided by native NMC.

## Updating Operator Data

You can write messages into the **Data 2** field of the **Resource Properties** window from the NMC. This can be used for problem management. For example, when you mark a resource that is in an exception state in order to indicate that you are working on the problem you can add in this way a comment for your colleagues.

You call this function by selecting **SA Icon Commands** and then **Update Operator Data**. A window appears where you can enter your comment. This will immediately be written into the **Data 2** field of the selected resource.

The advantage of this function, as compared to the standard function for information exchange between operators in the lower part of the **Resource Properties** window is that the comments can be persistently stored.

## Problem Management of Minor Resources

Minor resources inherit the view properties of the related major resources. Thus they appear in the same views as the major resource.

## Problem Management

For example, if a major resource were able to appear in an exception view, and a minor resource of this major resource runs into a problem, this minor resource will appear in the exception view (independently of whether the major resource is visible). However, because minor resources are bound to a major resource, they can only appear in views where you would expect the major resource.

You can apply problem management on minor resources just as you would do for major resources; that is, you can mark and unmark minor resources and you can also append a comment to them as described in “Updating Operator Data” on page 197. Additionally, in contrast to major resources, you can delete minor resources.



---

## Part 4. TEC Notification by SA z/OS

This part describes how to use TEC Notification by SA z/OS. It contains the following:

- Chapter 14, "Using TEC Notification by SA z/OS," on page 201



---

## Chapter 14. Using TEC Notification by SA z/OS

This chapter describes how to use TEC Notification by SA z/OS. After an introduction, “SA z/OS Messages, Alerts and Tivoli Events” on page 202 describes how SA z/OS messages and NetView alerts are mapped to Tivoli events. “Operation of TEC Notification by SA z/OS” on page 211 contains the information required to monitor Tivoli events from the Tivoli desktop. You find information on how to install TEC Notification by SA z/OS in *IBM Tivoli System Automation for z/OS Planning and Installation*.

---

### Understanding the Use of TEC Notification by SA z/OS

SA z/OS is an availability management product and covers several aspects of automation and enterprise monitoring of z/OS for a single system as well as in a sysplex environment.

However, automation may fail and in such cases, messages or alerts indicate that administrator intervention is required. In those cases, SA z/OS sends out messages to the NetView operators or the system operator, or both. The system operator is responsible for handling all events reported by SA z/OS from a single point of control for systems automation.

It may be desirable, however, to run SA z/OS integrated with Tivoli, so that events reported by SA z/OS, as well as other events reported in a distributed heterogeneous environment, are received at and handled from a single point of control. So SA z/OS is enhanced in a way that SA z/OS notifies the Tivoli Enterprise Console<sup>®</sup> (TEC) in cases when SA z/OS issues messages and alerts indicating critical situations. Such messages and alerts are forwarded to the TEC event server.

Thus, a Tivoli administrator running the Tivoli Enterprise Console may become aware of this event. The Tivoli administrator may also be notified when such an event is received by running an appropriate TEC rule that, for example, triggers a Tivoli notice to be sent to the Tivoli administrator’s notice board.

You may also correlate such an event with other events from the distributed heterogeneous environment so that the coincidence of correlated events may trigger a certain action.

Thus, a Tivoli administrator can use the Tivoli Enterprise Console as a single point of control for monitoring and handling events from both the distributed environment and SA z/OS.

This function of SA z/OS is called *TEC Notification by SA z/OS*. This notification takes advantage of the *Tivoli Event/Automation Service*. In this environment, the source of notification is of no importance. All events that occur in your enterprise that originate from Tivoli NetView, SA z/OS, etc. are handled from the Tivoli Enterprise Console, which is the single point of control for Tivoli administrators.

The Event/Automation Service allows you to effectively manage both distributed and z/OS events from a single interface on a single console. You can exploit the event correlation and automation capabilities of the Tivoli Enterprise Console for

## Understanding the Use of TEC Notification by SA z/OS

all events in the enterprise. You can find more information on the Event/Automation Service in *Tivoli NetView for z/OS Customization Guide*.

### A Note on Terms:

A selection of SA z/OS messages is mapped to Tivoli events. In contrast to traditional Tivoli events, these new events mapped from SA z/OS messages will be called **SA z/OS events** in this documentation.

## Introduction of TEC Notification by SA z/OS

The SA z/OS notification may notify Tivoli Enterprise Console (TEC) about an automation problem on z/OS by sending an event to the TEC event server.

For this purpose, on z/OS systems, messages or alerts are transformed into Tivoli events and sent to the *TEC event server* that runs on a Tivoli managed node in your network.

These events in turn may cause a notification of a Tivoli administrator on the TEC, may be correlated with other events on the TEC event server, or may result in opening a trouble ticket for example, dependent on what you programmed at the TEC event server.

The TEC event server is a notification target for SA z/OS. Note that not all messages, but only messages indicating critical situations and alerts are forwarded as TEC events to the TEC event server using the respective NetView Event/Automation Service Adapter.

A Tivoli administrator who wants to deal with a problem indicated by an event forwarded to the TEC event server by SA z/OS needs access to the affected z/OS system. For this purpose, you may use the Tivoli NetView 3270 Management Console. With TEC Notification by SA z/OS, the TEC administrator may log on to the NetView operator console by starting the NetView 3270 Management Console from the TEC console by executing a task. See “How TEC Notification by SA z/OS Works” on page 212 for a description on how to achieve that.

**Note:** Forwarding of SA z/OS messages to TEC will not start until SA z/OS and the Event/Automation Service are up and running. SA z/OS messages issued during SA z/OS startup will not be forwarded to TEC.

---

## SA z/OS Messages, Alerts and Tivoli Events

The complete set of SA z/OS messages is not mapped to Tivoli events; only an appropriate selection is handled by TEC Notification by SA z/OS. Table 9 on page 203 shows the messages that are mapped to Tivoli events. Table 10 on page 206 shows a list of available Tivoli events mapped from SA z/OS messages and how they are integrated into the existing Tivoli event class structure. Table 11 on page 209 shows the alerts that are mapped to Tivoli events.

For details on the format of message mapping or alert mapping see the respective shipped files in hlq.SINGSAMP: *INGMFMT* for message mapping and *INGACDS* for alert mapping.

Section “TEC Rules for SA z/OS Events” on page 209 describes the correlation rules between certain SA z/OS events defined by TEC Notification by SA z/OS.

## Mapping SA z/OS Messages to Tivoli Events

In this section:

- You can find information about the Tivoli event class structure.
- Table 9 shows how SA z/OS messages are mapped to Tivoli events (SA z/OS events).
- Table 10 on page 206 shows how SA z/OS messages are mapped to all involved SA z/OS events.

For more information about the meaning of SA z/OS messages, refer to *IBM Tivoli System Automation for z/OS Messages and Codes*.

Event classes are defined hierarchically, with child classes inheriting attribute value defaults from the parent. The SA z/OS event classes follow a simple hierarchy.

All SA z/OS events are instances of event classes that inherit from *NV390MSG\_SA* that, in turn, inherits from event class *NV390MSG\_Event*, the top level class of all event classes. All event classes defined for the NetView message adapter inherit from this class. This inheritance structure is outlined in Figure 64 on page 206.

The following attributes are used in event group filters. They are set by the NetView z/OS Message Adapter and overridden by TEC Notification by SA z/OS:

### sub\_source

All events originating from SA z/OS messages are issued by SA z/OS system operations. This is indicated by `sub_sources="SA0S390_SysOps"`.

The following attributes are common for all SA z/OS events:

### date\_msg

Gives the date and time when the message is generated. This may differ from the date and time the event that it is mapped to is generated (see the **date** attribute). The format is "MMM dd, hh:mm:ss" as in *date* and *date\_reception*, for example, **Nov 03, 18:34:27**.

### system\_id

Is a 1-8 character identifier of the z/OS system image that the message was issued on.

### sysplex\_id

Is a 1-8 character identifier of the sysplex that the message originated from. Because it is not always applicable, the default is *N/A*.

### os\_version

z/OS system product level of the z/OS system that the message was generated on.

Table 9. Mapping SA z/OS Messages to SA z/OS Events

SA z/OS Message	Tivoli Event Class	Tivoli Severity
AOF200I	SA390MSG_CmdUnexpectedCmdResult	CRITICAL
AOF201I	SA390MSG_ClistFailed	CRITICAL
AOF203I	SA390MSG_ClistAuthorizationFailed	WARNING
AOF204I	SA390MSG_ClistInvalidMissingParms	WARNING
AOF205A	SA390MSG_ClistFailedTimerExpired	CRITICAL
AOF206I	SA390MSG_CmdFailed	CRITICAL
AOF211E	SA390MSG_ACFNoAutoOps	CRITICAL

## SA z/OS Messages, Alerts and Tivoli Events

Table 9. Mapping SA z/OS Messages to SA z/OS Events (continued)

SA z/OS Message	Tivoli Event Class	Tivoli Severity
AOF214A	SA390MSG_EnvironmentNotInitialized	CRITICAL
AOF216I	SA390MSG_EnvironmentNotInitialized	CRITICAL
AOF218I	SA390MSG_CmdFailedACFSaveInProgress	WARNING
AOF221I	SA390MSG_ACFNoPrimaryFocalPoint	WARNING
AOF222I	SA390MSG_ACFNoBackupFocalPoint	WARNING
AOF228I	SA390MSG_NoInactiveResourceInScope	WARNING
AOF230I	SA390MSG_ACFNoTimeoutValue	WARNING
AOF231I	SA390MSG_ACFTimeoutValueInvalid	CRITICAL
AOF232I <sup>1</sup>	SA390MSG_ACFNoTimeoutValue	WARNING
AOF241I	SA390MSG_ClistInvalidMissingParms	CRITICAL
AOF244I	SA390MSG_ShutdownDelayed	HARMLESS
AOF248I	SA390MSG_ShutdownAlreadyInProgress	WARNING
AOF253I	SA390MSG_TimerNotFound	WARNING
AOF255I	SA390MSG_ACFInvalidUserExit	CRITICAL
AOF256I	SA390MSG_ACFNoOptionalExits	CRITICAL
AOF292I	SA390MSG_CannotAssignReplyId	CRITICAL
AOF293I	SA390MSG_NoOutstandingReply	CRITICAL
AOF294I	SA390MSG_SpoolShortageCmdsExhausted	CRITICAL
AOF303I	SA390MSG_RequestedStateAlreadySet	WARNING
AOF305I	SA390MSG_ProcessorNotInControl	WARNING
AOF309I	SA390MSG_TimerForCatchupExecuted	HARMLESS
AOF312E	SA390MSG_AutoFlagAlreadyOff	WARNING
AOF314I	SA390MSG_TimerForCatchupNotExecuted	CRITICAL
AOF315I	SA390MSG_TimerDeleted	HARMLESS
AOF324A	SA390MSG_ACFInvalidEntry	CRITICAL
AOF350E	SA390MSG_ClistFailed	CRITICAL
AOF355E	SA390MSG_ClistFailed	CRITICAL
AOF356E	SA390MSG_ClistFailed	CRITICAL
AOF501I	SA390MSG_ThresholdActionTaken	WARNING
AOF561A	SA390MSG_ACFNoDrainCmds	WARNING
AOF562I	SA390MSG_JES2DrainFailed	CRITICAL
AOF563A	SA390MSG_UnknownResourceType	WARNING
AOF564A	SA390MSG_ShutdownHaltedActiveJobDev	WARNING
AOF565A	SA390MSG_DrainCmdOff	WARNING
AOF568I	SA390MSG_OutBoundGTWStatus	WARNING
AOF569I	SA390MSG_InBoundGTWStatus	WARNING
AOF575A	SA390MSG_RecoveryNotInProgress	CRITICAL
AOF576A	SA390MSG_RecoveryAutoFlagOff	WARNING
AOF577E	SA390MSG_ThresholdRecoveryHalted	WARNING
AOF580I	SA390MSG_AutoFlagOff	CRITICAL

## SA z/OS Messages, Alerts and Tivoli Events

Table 9. Mapping SA z/OS Messages to SA z/OS Events (continued)

SA z/OS Message	Tivoli Event Class	Tivoli Severity
AOF582A	SA390MSG_RecoveryFailed	CRITICAL
AOF583I	SA390MSG_AutoFlagOff	CRITICAL
AOF584I	SA390MSG_ACFNoCmds	CRITICAL
AOF585I	SA390MSG_AutomationAlreadyInProgress	WARNING
AOF587I	SA390MSG_ThresholdRecoveryContinued	WARNING
AOF591E	SA390MSG_RecoveryAutoFlagOff	WARNING
AOF593A	SA390MSG_ShutdownCmdsExhausted	CRITICAL
AOF608A	SA390MSG_WarmStartFailure	CRITICAL
AOF609A	SA390MSG_InvalidReply	MINOR
AOF701A	SA390MSG_MsgTableNotLoaded	CRITICAL
AOF703E <sup>1</sup>	SA390MSG_CmdsExhausted	CRITICAL
AOF719I	SA390MSG_ClistUnexpectedCmdResult	CRITICAL
AOF739I	SA390MSG_ACFNoJESSSubsystem	CRITICAL
AOF745E	SA390MSG_ShutdownNoFinalMsg	WARNING
AOF746A	SA390MSG_ShutdownSuspended	CRITICAL
AOF747A	SA390MSG_ShutdownStuck	CRITICAL
AOF748E	SA390MSG_ShutdownNoFinalMsgResume	WARNING
AOF750I	SA390MSG_ShutdownCmdRplyFailed	CRITICAL
AOF752E	SA390MSG_ShutdownAutoFlagOff	CRITICAL
AOF759A	SA390MSG_RecoveryACFCMDFailed	CRITICAL
AOF760E	SA390MSG_ClistTrapped	CRITICAL
AOF766I	SA390MSG_ACFEntryMissed	CRITICAL
AOF768I	SA390MSG_ACFEntryMissed	CRITICAL
AOF793E	SA390MSG_ACFInconsistentDependencies	CRITICAL
AOF797A	SA390MSG_ACFLoadStopped	CRITICAL
AOF859I	SA390MSG_ACFInstructionIncomplete	CRITICAL
AOF860I	SA390MSG_ACFInstructionRejected	WARNING
AOF861I	SA390MSG_ACFRetractRejected	WARNING
AOF862I	SA390MSG_ACFBadInstruction	CRITICAL
AOF863I	SA390MSG_ACFDuplicateAssert	WARNING
AOF864I <sup>1</sup>	SA390MSG_ACFModifyToAssert	WARNING
AOF865I <sup>1</sup>	SA390MSG_ACFInheritedDataModified	WARNING
AOF866I	SA390MSG_ACFInvalidKeyword	WARNING
AOF867I	SA390MSG_ACFInconsistentData	CRITICAL
AOF868I	SA390MSG_ACFNoJESSSubsystem	CRITICAL
AOF905I	SA390MSG_XCFCommandFailed	WARNING
AOF906I	SA390MSG_MVSCCommandFailed	WARNING
AOF914E	SA390MSG_RecoveryFailed	CRITICAL
AOF924A	SA390MSG_RecoveryFailed	CRITICAL
AOF927E	SA390MSG_RecoveryFailed	CRITICAL





## SA z/OS Messages, Alerts and Tivoli Events

Table 10. SA z/OS Events and Corresponding Mapped SA z/OS Messages (continued)

Tivoli Event Class	SA z/OS Message	Tivoli Severity
SA390MSG_ACFInstructionIncomplete	AOF859I	CRITICAL
SA390MSG_ACFInvalidEntry	AOF324A	CRITICAL
SA390MSG_ACFInvalidUserExit	AOF255I	CRITICAL
SA390MSG_ACFNoAutoOps	AOF211E	CRITICAL
SA390MSG_ACFNoBackupFocalPoint	AOF222I	WARNING
SA390MSG_ACFNoCmds	AOF584I	CRITICAL
SA390MSG_ACFNoDrainCmds	AOF561A	WARNING
SA390MSG_ACFNoJESSubsystem	AOF739I AOF868I	CRITICAL
SA390MSG_ACFNoOptionalExits	AOF256I	CRITICAL
SA390MSG_ACFNoPrimaryFocalPoint	AOF221I	WARNING
SA390MSG_ACFNoTimeoutValue	AOF230I AOF232I	WARNING
SA390MSG_ACFTimeoutValueInvalid	AOF231I	CRITICAL
<i>SA390MSG_ACFUpdateInvalid</i>	base class	
SA390MSG_ACFBadInstruction	AOF862I	CRITICAL
SA390MSG_ACFDuplicateAssert	AOF863I	WARNING
SA390MSG_ACFInheritedDataModified	AOF865I	WARNING
SA390MSG_ACFInstructionRejected	AOF860I	WARNING
SA390MSG_ACFInvalidKeyword	AOF866I	WARNING
SA390MSG_ACFModifyToAssert	AOF864I	WARNING
SA390MSG_ACFRetractRejected	AOF861I	WARNING
SA390MSG_AutoFlagAlreadyOff	AOF312E	WARNING
<i>SA390MSG_AutoFlagOff</i>	AOF580I AOF583I	CRITICAL
<i>SA390MSG_AutomationAlreadyInProgress</i>	AOF585I	WARNING
<i>SA390MSG_CannotAssignReplyId</i>	AOF292I	CRITICAL
<i>SA390MSG_Clist</i>	base class	
SA390MSG_ClistAuthorizationFailed	AOF203I	WARNING
SA390MSG_ClistFailed	AOF201I AOF350E AOF355E AOF356E	CRITICAL
SA390MSG_ClistFailedTimerExpired	AOF205A	CRITICAL
SA390MSG_ClistInvalidMissingParms	AOF204I AOF241I	WARNING
SA390MSG_ClistTrapped	AOF760E	CRITICAL
SA390MSG_ClistUnexpectedCmdResult	AOF719I	CRITICAL
<i>SA390MSG_CmdFailed</i>	AOF206I	CRITICAL
<i>SA390MSG_CmdFailedACFSaveInProgress</i>	AOF218I	WARNING
<i>SA390MSG_CmdsExhausted</i>	AOF703E	CRITICAL

## SA z/OS Messages, Alerts and Tivoli Events

Table 10. SA z/OS Events and Corresponding Mapped SA z/OS Messages (continued)

Tivoli Event Class	SA z/OS Message	Tivoli Severity
<i>SA390MSG_CmdUnexpectedCmdResult</i>	AOF200I	CRITICAL
<i>SA390MSG_DrainCmdOff</i>	AOF565A	WARNING
<i>SA390MSG_EnvironmentNotInitialized</i>	AOF214A AOF216I	CRITICAL
<i>SA390MSG_InBoundGTWStatus</i>	AOF569I	WARNING
<i>SA390MSG_InvalidReply</i>	AOF609A	MINOR
<i>SA390MSG_JES2DrainFailed</i>	AOF562I	CRITICAL
<i>SA390MSG_MsgTableNotLoaded</i>	AOF701A	CRITICAL
<i>SA390MSG_NoInactiveResourceInScope</i>	AOF228I	WARNING
<i>SA390MSG_NoOutstandingReply</i>	AOF293I	CRITICAL
<i>SA390MSG_OutBoundGTWStatus</i>	AOF568I	WARNING
<i>SA390MSG_ProcessorNotInControl</i>	AOF305I	WARNING
<i>SA390MSG_RequestedStateAlreadySet</i>	AOF303I	WARNING
<i>SA390MSG_SpoolShortageCmdsExhausted</i>	AOF294I	CRITICAL
<i>SA390MSG_SubsystemRecovery</i>	base class	
<i>SA390MSG_RecoveryACFCMDFailed</i>	AOF759A	CRITICAL
<i>SA390MSG_RecoveryAutoFlagOff</i>	AOF576A AOF591E	WARNING
<i>SA390MSG_RecoveryFailed</i>	AOF582A	CRITICAL
<i>SA390MSG_RecoveryNotInProgress</i>	AOF575A	CRITICAL
<i>SA390MSG_SubsystemShutdown</i>	base class	
<i>SA390MSG_ShutdownAlreadyInProgress</i>	AOF248I	WARNING
<i>SA390MSG_ShutdownAutoFlagOff</i>	AOF752E	CRITICAL
<i>SA390MSG_ShutdownCmdsExhausted</i>	AOF593A	CRITICAL
<i>SA390MSG_ShutdownCmdRplyFailed</i>	AOF750I	CRITICAL
<i>SA390MSG_ShutdownDelayed</i>	AOF244I	HARMLESS
<i>SA390MSG_ShutdownHaltedActiveJobDev</i>	AOF564A	WARNING
<i>SA390MSG_ShutdownNoFinalMsg</i>	AOF745E	WARNING
<i>SA390MSG_ShutdownNoFinalMsgResume</i>	AOF748E	WARNING
<i>SA390MSG_ShutdownStuck</i>	AOF747A	CRITICAL
<i>SA390MSG_ShutdownSuspended</i>	AOF746A	CRITICAL
<i>SA390MSG_Threshold</i>	base class	
<i>SA390MSG_ThresholdActionTaken</i>	AOF501I	WARNING
<i>SA390MSG_ThresholdRecoveryContinued</i>	AOF587I	WARNING
<i>SA390MSG_ThresholdRecoveryHalted</i>	AOF577E	WARNING
<i>SA390MSG_Timer</i>	base class	
<i>SA390MSG_TimerDeleted</i>	AOF315I	HARMLESS
<i>SA390MSG_TimerForCatchupExecuted</i>	AOF309I	HARMLESS
<i>SA390MSG_TimerForCatchupNotExecuted</i>	AOF314I	CRITICAL
<i>SA390MSG_TimerNotFound</i>	AOF253I	WARNING

Table 10. SA z/OS Events and Corresponding Mapped SA z/OS Messages (continued)

Tivoli Event Class	SA z/OS Message	Tivoli Severity
SA390MSG_UnknownResourceType	AOF563A	WARNING
SA390MSG_WarmStartFailure	AOF608A	CRITICAL

## Mapping NetView Alerts to Tivoli Events

SA z/OS processor operations indicates severe failures that require operator intervention by issuing NetView alerts.

These alerts are forwarded to the Tivoli Enterprise Console via the alert adapter. Generic mapping of the NetView generic alerts applies, that is, no new event classes are introduced, except those that are defined by the NetView z/OS alert adapter.

The only attribute default that is changed when alerts are issued by SA z/OS processor operations is the default of the *sub\_source* attribute. Whereas the alert adapter sets NET as the default, the *sub\_source* attribute has the value SAOS390\_ProcOps when alerts are issued by SA z/OS.

The following table describes the resulting mapping of SA z/OS processor operations alerts to the Tivoli Enterprise Console event classes:

Table 11. Mapping SA z/OS Alerts to SA z/OS Events

AlertId	SA z/OS Processor Operations Alert	Tivoli Event Class
1	COMMUNICATION TASK FAILED	SNA_Software_Program_Error
3	DCCF DETECTED	SNA_Software_Program_Error
4	WAIT STATE DETECTED	SNA_Software_Program_Error
5	CONSOLE LOST	SNA_Input_Output_Device_Error
6	CONSOLE ACCESS LOST	SNA_Input_Output_Device_Error
7	COULD NOT OBTAIN SYSTEM CONSOLE	SNA_Input_Output_Device_Error
8	TARGET HARDWARE PROBLEM	SNA_Equipment_Malfunction

## TEC Rules for SA z/OS Events

A selected set of SA z/OS messages is mapped to Tivoli events and forwarded to the Tivoli Enterprise Console. Those events can be filtered at the Tivoli Enterprise Console, correlated with other events, or trigger an action.

With TEC Notification by SA z/OS you receive a set of sample rules correlating SA z/OS events.

The following sections list the SA z/OS messages that correlation rules are provided for and the semantics of those rules is shortly described.

### Correlation Rules for SA z/OS Message AOF564A

#### SA z/OS Message

```
AOF564A {time: JES2 SHUTTING DOWN - message - OPERATOR INTERVENTION
REQUIRED | title | task_names| job_names| device_names}
```

## SA z/OS Messages, Alerts and Tivoli Events

Where:

- *message* is one of:

DEVICES NOT DRAINED  
JOBS STILL EXECUTING  
TASKS STILL EXECUTING  
JOBS STILL EXECUTING, DEVICES NOT DRAINED  
TASKS STILL EXECUTING, DEVICES NOT DRAINED  
JOBS AND TASKS STILL EXECUTING  
JOBS AND TASKS STILL EXECUTING TO DEVICES NOT DRAINED

- *title* is one of:

Tasks ...  
Devices ...  
Jobs ...

### Description

JES2 is being shut down and there are jobs, tasks or devices that have not been drained. This is indicated by:

AOF564A *time* : JES2 SHUTTING DOWN - *txt* -  
OPERATOR INTERVENTION REQUIRED.

This message is followed by a number of messages of type:

AOF564A *time* : *title*  
AOF564A *time* : {*device\_names* | *job\_names* | *task\_names* }

These detail the jobs or tasks or devices causing the shutdown halt.

### Event Class

SA390MSG\_ShutdownHaltedActiveJobDev

### Event Severity

WARNING

### Correlation Rules:

sa390msg\_related\_564\_rule1  
sa390msg\_related\_564\_rule2  
sa390msg\_related\_564\_rule3  
sa390msg\_related\_564\_rule4  
sa390msg\_related\_564\_rule5  
sa390msg\_related\_564\_rule6

These rules link events that originate from messages:

AOF564A *time* : {*device\_names* | *job\_names* | *task\_names* }

to an event that originates from the message:

AOF564A *time* : JES2 SHUTTING DOWN - *txt* - OPERATOR INTERVENTION REQUIRED.

Further devices, jobs or tasks that are listed in the above message are copied to the *user2* attribute of the event that originates from:

AOF564A *time* : JES2 SHUTTING DOWN - *txt* - OPERATOR INTERVENTION REQUIRED.

When events are received that originate from messages of type:

AOF564A *time* : *title*

the information about the type of resource that is causing the shutdown halt is added to the *user1* attribute of the event that originates from:

AOF564A *time* : JES2 SHUTTING DOWN - *txt* - OPERATOR INTERVENTION REQUIRED.

If the status of an event that originates from a message of the following type is changed:

```
AOF564A time : {device_names | job_names | task_names }
```

This is propagated to the linked message that originates from a message of type:

```
AOF564A time : JES2 SHUTTING DOWN - txt - OPERATOR INTERVENTION REQUIRED.
```

by adding this information to the *user3* and *user4* attributes.

---

## Operation of TEC Notification by SA z/OS

This chapter describes the use of TEC Notification by SA z/OS. It is separated in the following sections describing the appropriate aspects of using TEC Notification by SA z/OS:

- “Startup of TEC Notification by SA z/OS”
- “How TEC Notification by SA z/OS Works” on page 212
- “Termination of TEC Notification by SA z/OS” on page 214

The starting point for all Tivoli-related operations after a successful startup of TEC Notification by SA z/OS is the Tivoli desktop.

## Startup of TEC Notification by SA z/OS

This section lists:

- The actions you need to perform to prepare the monitored z/OS systems to route messages and alerts to Tivoli (see “Preparing the Host”)
- The actions that need to be performed on the monitoring Tivoli Enterprise Console to display incoming events (see “Displaying Events on the Tivoli Enterprise Console” on page 212).

### Preparing the Host

#### Prerequisites:

- Tivoli NetView for z/OS is up and running.
- Automation table with fragment INGMTEC included is active.
- System Automation for z/OS is initialized.

If all prerequisites are fulfilled, the autotask has been started and the notification classes for the autotask have been set. Message AOF646I (Tivoli Enterprise Console Notification Service is initialized) is issued to indicate that the system is correctly customized and initialized.

#### Actions and System Responses:

1. On the z/OS focal point system where the message adapter is running, you need to start the Event/Automation Service. Issue the command:

```
S IHSAEVNT
```

If you issue this command as shown here without any options, the Event/Automation Service will use the defined initialization and adapter configuration files. For more information on this command, see *Tivoli NetView for z/OS Customization Guide*.

2. As a result of the previous step, the message adapter and the alert adapter, if configured, become initialized. When these are completed, this is indicated by the IHS0124I messages:

## Operation of TEC Notification by SA z/OS

```
IHS0124I: Message Adapter task initialization complete.
IHS0124I: Alert Adapter task initialization complete.
```

3. Message IHS0124I for the message adapter triggers the execution of the Notification Service initialization procedure, INGR TIN. This procedure sets the environment global variables and as a result of this procedure, routing of SA z/OS messages to TEC is enabled. This is indicated by message AOF642I:  
TIVOLI ENTERPRISE CONSOLE NOTIFICATION SERVICE IS ENABLED
4. Alert routing must be enabled manually. Depending on where the alert adapter is running, either on the SA z/OS processor operations focal point system or the SA z/OS system operations focal point system, you need to issue the NetView command:  
NPDA SRFILTER TECROUTE PASS E NTFY

to set the TECROUTE filter for alerts of type *NOTIFICATION* to **PASS**.

The message

```
BNJ1341I SRF/SRFILTER COMMAND ACCEPTED
```

is returned. The correct setting for the filter can be verified by issuing the command NPDA DFILTER TECROUTE. The output shown below should be displayed on the NetView console with the action for the event type NTFY set to PASS.

```
NETVIEW          SESSION DOMAIN: IPUFJ   BALI   06/10/97 10:33:57
NPDA-20A          * CURRENT FILTER STATUS *          REC 1 TO 2 OF 2
                  FILTER TYPE: ALERT FORWARDING TO T/EC
```

```
SEL# ACTION DATA  ETYPE FTYPE  ----- RESNAME, TYPE, OR ADAPTADR -----
( 1) PASS  ....  NTFY
**** DEFAULT ACTION **** BLOCK ****
```

## Displaying Events on the Tivoli Enterprise Console

### Prerequisites:

1. Enter the appropriate Tivoli command to start the Tivoli desktop.
2. TEC event server starts. If this is not the case, you can start it, for example, from the **Start-up** pull-down menu of the **EventServer** icon.

### Task Sequence:

1. Start the Tivoli Enterprise Console.
2. Open the event groups specific for SA z/OS to have incoming events displayed by double clicking on the appropriate icon.

## How TEC Notification by SA z/OS Works

### Host Part

SA z/OS issues messages using the module AOFMSG, which generates and forwards the messages to the registered auto operator task %AOFTECTASK%. This autotask processes the automation table fragment INGMTEC. If the message matches the filter criteria, it is either directly provided to the local message adapter (in case you work in a local configuration) or it is sent to the focal point system (for a distributed configuration). You find a description of both the local and the distributed environment in *IBM Tivoli System Automation for z/OS Planning and Installation*.

SA z/OS alerts are forwarded to the Networking NetView focal point system and provided to the alert adapter, which routes them to TEC.

## Tivoli Enterprise Console Part

Incoming events are displayed in the corresponding event group (SA SYSOPS Events). Predefined correlation rules are applied if applicable. TEC Notification by SA z/OS provides a Tivoli task that enables you to invoke the NetView 3270 Management Console from the Tivoli Enterprise Console no matter where the NetView 3270 Management Console is installed.

As shown in Figure 65, select an event that you want to invoke the NetView 3270 Management Console for, and from the **Selected** pull-down menu select **Task Execution**.

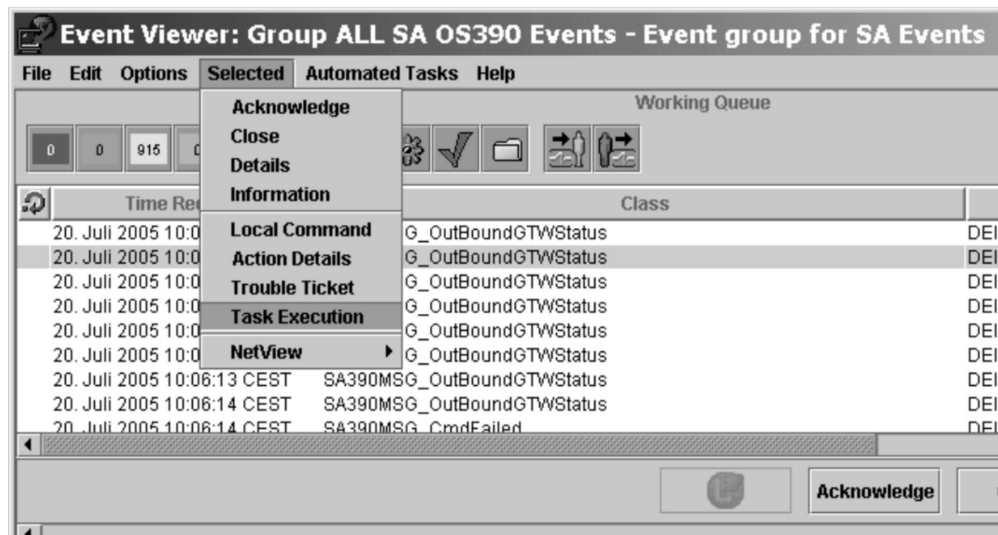


Figure 65. Invoking the TEC Task Execution Dialog for Events

This opens the Task Execution dialog, as shown in Figure 66 on page 214. You can also open the Task Execution dialog from a selected event by right clicking on the event and selecting **Task Execution** from the context menu.

## Operation of TEC Notification by SA z/OS

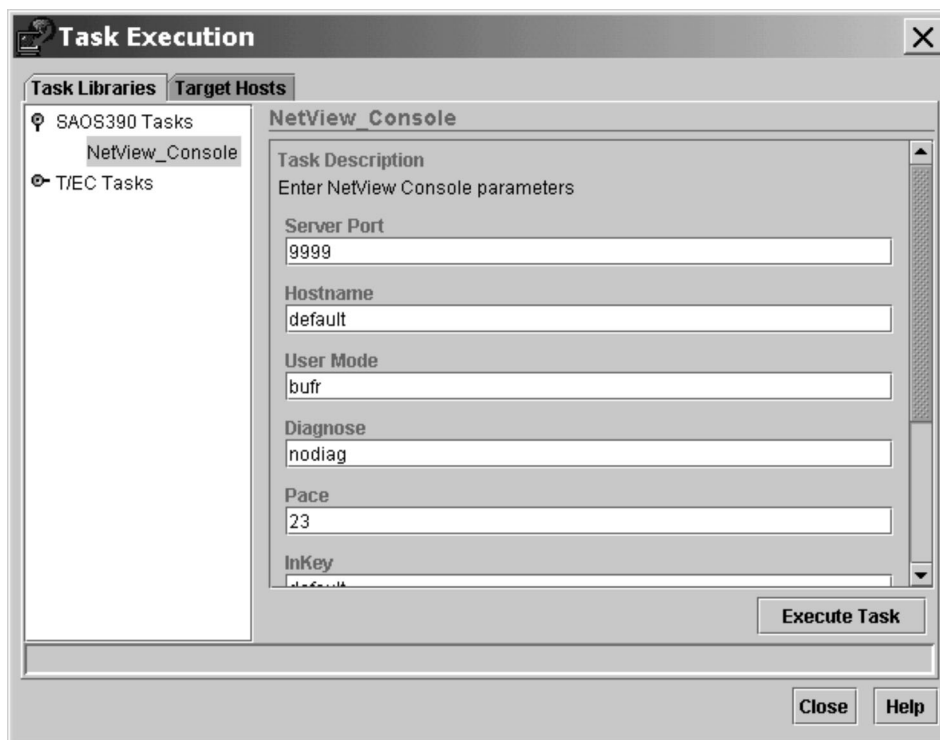


Figure 66. TEC Task Execution Dialog

From the Task Execution dialog, you can invoke the NetView 3270 Management Console by:

- Double clicking on a task library (in this example: SAOS390 Tasks).
- Double clicking on a task (in this example: *Start NetView Console*).
- Optionally changing the default configuration parameters for the startup of the NetView 3270 Management Console.
- Clicking on the **Execute Task** push button.

Before you reach the NetView 3270 Management Console screen, a NetView Logon Panel prompts you for your operator ID and the password.

With the NetView 3270 Management Console you can log on to the NetView of the z/OS system that, as the event source, triggered the event. As a system operator or Tivoli administrator, you can then repair any damage on the monitored z/OS system directly from your Tivoli Enterprise Console by entering NetView commands at the NetView 3270 Management Console.

## Termination of TEC Notification by SA z/OS

### Host Part

Terminating the SA z/OS message notification to TEC is achieved by deregistering the auto operator task *AUTOTEC* as a notification target for SA z/OS messages. You cancel this notification for the auto operator *AUTOTEC* using the NetView command:

```
/AUTOTEC: INGNFY OFF
```

Terminating SA z/OS alerts notification to TEC is achieved by setting the TECROUTE filter to BLOCKED:

```
NPDA SRFILTER TECROUTE BLOCK E NTFY
```



### **Tivoli Enterprise Console Part**

After terminating the notification of SA z/OS messages and NetView alerts on the host, you will only see incoming traditional Tivoli events on the Tivoli Enterprise Console.

## Operation of TEC Notification by SA z/OS

---

## Part 5. Appendixes



---

## Appendix A. Automation Initialization Options

At the start of SA z/OS initialization a multiline write-to-operator (MLWTO) message, AOF767I, is issued describing the options available and system default names and actions. The lines of the MLWTO define the options that can be specified for SA z/OS initialization. Each line is listed and explained below.

AOF767I is followed by either an AOF603D or an AOF606D WTOR. SA z/OS replies automatically to an AOF603D after two minutes unless this has been changed as part of your site's automation policy. SA z/OS does not automatically reply to an AOF606D message.

The appearance of AOF767I, AOF603D and AOF606D can be suppressed by setting the variable AOFINITREPLY to 0 (see appendix "Global Variables to Enable Advanced Automation" in *IBM Tivoli System Automation for z/OS Defining Automation Policy*).

A description of the options follows:

- . STOP - CANCEL AUTOMATION  
This line is always present. It indicates that you can reply **STOP**. Doing so causes SA z/OS to issue an AOF605I and stop running. You must recycle NetView to restart SA z/OS.
- . PAUSE - SUSPEND AUTOMATION  
This line is present unless SA z/OS initialization has previously been paused. It indicates that you can reply **PAUSE**. Doing so causes SA z/OS to go into a paused state. An AOF604I message is issued, the AOF767I message is reissued, and an AOF606D WTOR is issued. You should not automate the reply to the AOF606D WTOR, as this disables the pause function.  
  
If the automation manager is (re-)started or refreshed (via the INGAMS or MVS modify command) while the automation agent is paused, the reply is cancelled and the automation agent initialization is redriven from the beginning.
- . TABLE= - OVERRIDE DEFAULT AUTOMATION TABLE  
This line is always present. It indicates that you can reply **TABLE=new\_table**. Doing so causes SA z/OS to load the NetView automation table *new\_table* rather than the ones whose names are specified in the automation control file being loaded. The NetView automation table name specified here overrides a table name stored in the warm start cache. The NetView automation table name specified must exist. If SA z/OS is unable to load the automation table it issues an AOF701I message telling you this, and an AOF715D WTOR asking you to specify the name of an alternative automation table.
- . NOSTART - DO NOT AUTOMATE SUBSYSTEM STARTUP  
This line is always present. It indicates that you can reply **NOSTART**. Doing so causes SA z/OS to initialize the status of all subsystems to AUTODOWN, rather than DOWN and set the resource's hold flag on the automation manager to YES. This means that after initial status determination is complete, SA z/OS does not go on to start the subsystems. When you do want the subsystems started, issue an appropriate SA z/OS command to achieve this. If VTAM is up you can log onto NetView and issue the commands there.

## Automation Initialization Options

- . NOSAVE - DO NOT SAVE DATA TO CACHE  
This line is present if the DSISVRT task, which is used for the warm start cache, is operational. It indicates that you can reply **NOSAVE**. Replying **NOSAVE** instructs SA z/OS not to save the data loaded from the automation control file into the warm start cache. The default action (if you do not reply **NOSAVE**) after a cold start is to save the data into the warm start cache.
- . WARM - PERFORM A WARM START FROM CACHE  
This line is present if SA z/OS has determined that the warm start cache is available and contains data from an automation control file that was generated by the SA z/OS customization dialog. It indicates that you can reply **WARM**. Doing so causes SA z/OS to start from the warm start cache.
- . REFRESH - PERFORM A WARM START FROM CACHE, REFRESH CHANGED DATA  
This line is present if SA z/OS has determined that the warm start cache is available and contains data from an automation control file that was generated by the SA z/OS customization dialog. This indicates that SA z/OS has detected changed data on disk. Specifying **REFRESH** means that SA z/OS starts from the warm start cache and picks up changed data from the disk.
- . COLD - PERFORM A COLD START FROM DISK  
If you specify **COLD**, SA z/OS will start with data from the disk and will not use any data from the cache.
- . ENTER - CONTINUE WITH *default*  
where *default* is one of **WARM**, **REFRESH** or **COLD**. Refer to the appropriate description in this list.

**NOTE: COLD START NOT POSSIBLE, NO ACF MEMBER AVAILABLE**

This line is present if no valid automation control file has been found on disk. If the warm start cache is valid (that is, the cache timestamp token matches the Automation Manager token) then a WARM start will be attempted.

**NOTE: WARM START NOT POSSIBLE, DSISVRT NOT AVAILABLE**

This line is present if SA z/OS has determined that the warm start cache is unavailable. It indicates that the DSISVRT task could not be started.

**NOTE: NO USABLE ACF IN CACHE**

This line is present if SA z/OS has determined that the warm start cache is available, but the data in it cannot be used.

**NOTE: CACHE NEEDS TO BE REFRESHED**

This line is present if SA z/OS has determined that the warm start cache is available, but needs refreshing. If the ACF updates will only affect the Automation Agent then a COLD start will be the default.

If the ACF updates affect the Automation Manager then a WARM start will be the default. When ACF updates affect the Automation Manager an INGAMS REFRESH is required before a COLD start is possible.

**NOTE: REFRESH REQUIRED BUT NOT POSSIBLE**

This line is present if SA z/OS has determined that the warm start cache is available, and needs refreshing, but the ACF member is invalid or missing. In this case SA z/OS will proceed with a WARM start from cache.

**NOTE: CACHE IS CURRENT, NO UPDATE REQUIRED**

This line is present if SA z/OS has determined that the warm start cache is available, and does not need refreshing.

**NOTE: DEFAULT IS: *default***

This line indicates the default (one of **WARM**, **REFRESH** or **COLD**).

A sample of how the AOF767I message could look, is in Figure 67.

```
AOF767I AUTOMATION OPTIONS: 497
. STOP      - CANCEL AUTOMATION
. PAUSE     - SUSPEND AUTOMATION
. NOSTART   - DO NOT AUTOMATE SUBSYSTEM STARTUP
. NOSAVE    - DO NOT SAVE AUTOMATION CONTROL FILE
. COLD      - COLD START FROM DISK
. TABLE=   - OVERRIDE DEFAULT AUTOMATION TABLE(S)
. ENTER     - CONTINUE WITH 'COLD'
NOTE: NO USABLE ACF IN CACHE
NOTE: DEFAULT IS: COLD
68 AOF603D ENTER AUTOMATION OPTIONS OR 'R' (REDISPLAY)- DOMAIN IPSNO
```

*Figure 67. Sample AOF767I Message*

If the message on your screen looks significantly different from the sample shown, ask your automation programmer whether he used the initialization exit *AOFEXI01* to modify the message.





---

## Appendix B. Using the I/O Operations ISPF Dialog

### Notes:

1. Refer to *IBM Tivoli System Automation for z/OS Planning and Installation* for instructions on how to install and to modify the ISPF dialog.
2. The IBM-supplied REXX code, including the EXECs, panels, messages, and tables, is supported, provided it has not been modified. Therefore, if you intend to modify IBM-supplied code, it is advisable to retain two copies: the IBM-supplied copy and the modified
3. For more information on ISPF, refer to the ISPF publications that can be found in the z/OS Internet Library at <http://www.ibm.com/servers/eserver/zseries/zos/bkserv/>.

---

## Getting Started

To use the I/O operations ISPF dialog:

- You must already be logged on to your user ID and using ISPF.
- The I/O operations ISPF dialog must be available.

To see whether the I/O operations ISPF dialog is available, display the ISPF/PDF Primary Option panel on the ISPF main menu.

- If I/O operations is displayed as an option on the ISPF main menu, select it.  
The I/O operations ISPF main menu, which is shown in Figure 68 on page 224, should now be displayed. If it is not, consult your system administrator or *IBM Tivoli System Automation for z/OS Planning and Installation* for further instructions.
- If I/O operations is not displayed on the ISPF/PDF Primary Option panel:
  1. Select option 6 (Command) on the ISPF/PDF Primary Option panel and press Enter.
  2. If the library that is required by the customization dialog (ING.CUSTOM.IHVCONF) has been allocated using the TSO logon procedure or using a CLIST of your own, enter `INGDLG SELECT(IOCONNECT) ALLOCATE(NO)`  
(To issue the `INGDLG` command, the `ING.SINGIREX` data set must be allocated in the `SYSPROC DD` statement of your logon procedure.)  
The I/O operations ISPF main menu, which is shown in Figure 68 on page 224, should now be displayed. If it is not, or if your TSO logon procedure is not set up to allocate the data set that is required by the ISPF dialog, consult your system administrator or *IBM Tivoli System Automation for z/OS Planning and Installation* for further instructions.

---

## Starting with the I/O Operations ISPF Main Menu

This section briefly describes the I/O operations ISPF main menu, which is shown in Figure 68 on page 224.

## Starting with the I/O Operations ISPF Main Menu

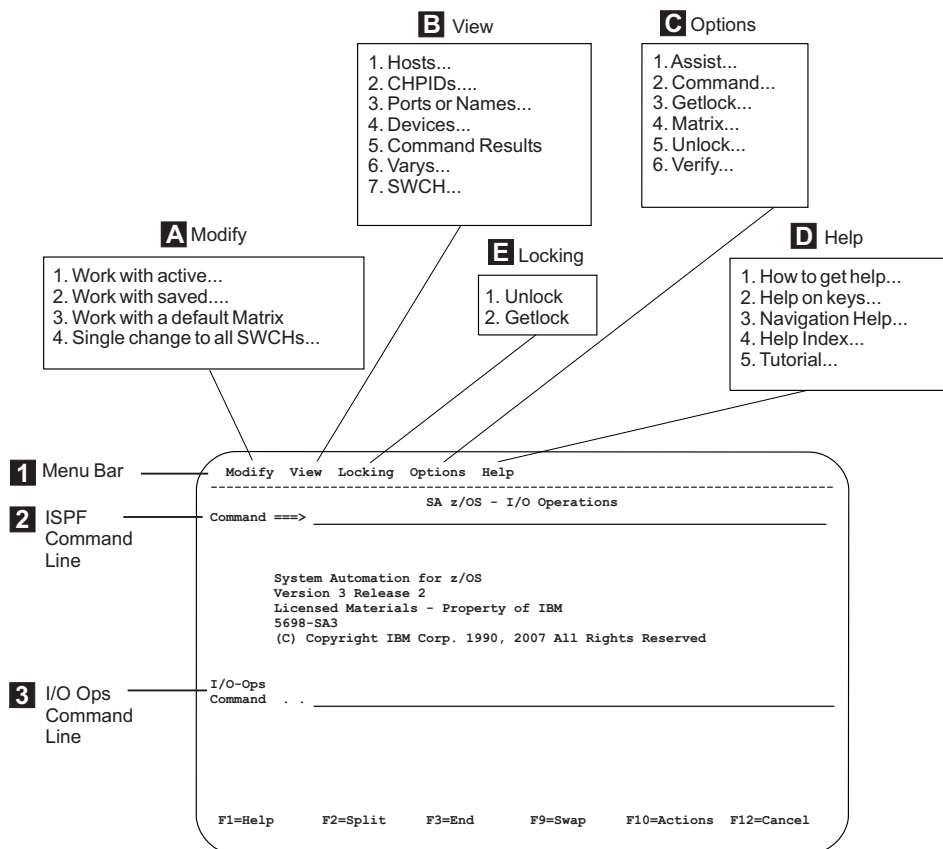


Figure 68. I/O operations ISPF Main Menu

**1** **Menu Bar** that you select menus **A** through **E** from.

### **A** **Modify Menu**

Lets you modify a switch configuration in matrix format and issue a change to all switches that are allocated to the issuing SA z/OS I/O operations. Place the cursor in Modify and press Enter. See "The Modify Menu" on page 225 for more information.

### **B** **View Menu**

Prompts you for the operands of the Display commands and then displays the data you request. Place the cursor in View and press Enter. See "The View Menu" on page 228 for more information.

### **C** **Options Menu**

Lets you customize the IBM-supplied defaults for your user ID. Place the cursor in Options and press Enter. See "The Options Menu" on page 230 for more information.

### **D** **Help Menu**

Shows you how to display the online help and the online tutorial. Place the cursor in Help and press Enter. See "The Help Menu" on page 232 for more information.

### **E** **Locking Menu**

This is a simple way to enter the Getlock and Unlock commands. Enter 1 or 2. If access is controlled by RACF, the UNLOCK command requires only Read authorization, but the GETLOCK command requires the Control level. See *IBM Tivoli System Automation for z/OS Operator's Commands* for

## Starting with the I/O Operations ISPF Main Menu

information about authorization required for I/O operations commands and for information about the GETLOCK and UNLOCK commands.

### 2 ISPF Command Line

On the ISPF command line, you can enter any applicable ISPF command. For more information on ISPF, refer to the ISPF publications that can be found in the z/OS Internet Library at [http://www.ibm.com/servers/eserver/zseries/zos/bkserv/..](http://www.ibm.com/servers/eserver/zseries/zos/bkserv/)

### 3 I/O Operations Command Line

On the I/O operations command line, you can enter any SA z/OS I/O operations command. See *IBM Tivoli System Automation for z/OS Operator's Commands* for a complete list of these commands.

---

## The Modify Menu

### Modify

- |   |
|---|
| 1. Work with active...<br>2. Work with saved...<br>3. Work with a default Matrix...<br>4. Single change to all SWCHs... |
|---|

Select a Subtask	Action (Press Enter when Needed)
1. Work with active (matrix)	<ul style="list-style-type: none"><li>a. Select a switch from the list in the Select SWCH panel.</li><li>b. Go to "A Sample Way to Modify a Switch Matrix" on page 226 for more information.</li></ul>
2. Work with saved (matrix)	<ul style="list-style-type: none"><li>a. Select a saved configuration from either:<ul style="list-style-type: none"><li>- Host (saved in ISPF)</li><li>- Switch</li></ul></li><li>b. Follow directions.</li></ul>
3. Work with a default Matrix	Go to "A Sample Way to Modify a Switch Matrix" on page 226 for more information.
4. Single change to all SWCHs	<ul style="list-style-type: none"><li>a. Select a command on the Single Change to All SWCHs panel.</li><li>b. Enter the required information in the fields on the panel (for example, enter a port address or name for the Block command).</li><li>c. Press F6.</li><li>d. If the Verify panel is displayed, verify the information and press Enter.</li><li>e. When processing is complete, do one of the following:<ul style="list-style-type: none"><li>- Repeat this sequence of steps to make another change.</li><li>- Press F3 to exit.</li></ul></li></ul>

After you have created or changed a switch configuration in matrix format, you can activate it or make it operational on the designated switch, or you may save it on the host or on a selected ESCON/FICON Director. The issuing SA z/OS I/O operations must be running and the switch must be allocated to it.

## The Modify Menu

You cannot activate the IBM-supplied default matrix until you have changed its default switch device number of 0000 to a number that is valid for your configuration. (See the last entry in the table under “Sample Changes You Can Make to a Port” on page 227.)

It is very advisable to save an active switch matrix, even if this is not mandatory.

### A Sample Way to Modify a Switch Matrix

When a switch matrix is displayed, you can make the following changes on it; you do not have to enter commands on the command line.

1. If you want to assign a name to the switch, enter the name in the **SWCH Name** field.

**Note:** Press Enter if you want to see the change highlighted.

2. To make a change to a port, check whether that port’s horizontal row is displayed on the matrix. If it is not:
  - To scroll down the left column of ports, use F8 to scroll to the higher port addresses and F7 to scroll in the other direction (backwards).
  - To scroll across the columns of ports in the matrix, use F11 to the right across the screen and PF10 to scroll to the left.

Some sample changes are shown under “Sample Changes You Can Make to a Port” on page 227.

3. After you have finished making changes, consider whether you want to save or to activate the switch configuration, or both.

**Note:** It is strongly recommended that you save a switch configuration before activating it.

4. To save a switch configuration:
  - a. Press F5.
  - b. On the Select Destination of Save Configuration panel, first choose between saving at a switch or at the host. Then, either select a name from the list that is presented, or enter a new name and description.
  - c. Press Enter to save the matrix of the switch configuration.

If you are replacing another switch configuration, another panel will prompt you to verify the change. In that case, press Enter again.
  - d. If you want to activate the switch configuration, refer to the next step.
5. To activate the switch configuration:
  - a. Press F6 on the screen of the matrix.
  - b. Verify the information.
  - c. Press Enter.

**Note:** If the saving or the activation of a switch configuration fails you can press the function key F2 to get the reason (or reasons) why your action failed. As an alternative to the function key you can issue the command RESULTS from the command line.

## Sample Changes You Can Make to a Port

I/O operations Command	Action
Block a port	<ul style="list-style-type: none"> <li>Type <b>B</b> at the intersection point between the port row and the B column. For example, if you type <b>B</b> in the B column at D5's row, D5 will be blocked.</li> </ul>
Unblock a blocked port	<ul style="list-style-type: none"> <li>Type <b>U</b> or a space at the intersection point between the port row and the B column. For example, type <b>U</b> over the B in the B column at D5's row to unblock port D5.</li> </ul>
Connect two ports statically (dedicated)	<ul style="list-style-type: none"> <li>Type the port address of the 2nd port at the intersection point between the 1st port and the C column. For example, type <b>C3</b> in the C column at port row E4 to connect C3 and E4 statically.</li> </ul>
Disconnect a static (dedicated) connection	<ul style="list-style-type: none"> <li>Type spaces over the port address at the intersection point between the port row and the C column. For example, type spaces over C3 in the C column at port row E4 to disconnect C3 and E4 from each other.</li> </ul>
Prohibit a dynamic connection	<ul style="list-style-type: none"> <li>Using <b>PF10</b> and <b>F11</b> to scroll right and left, display the port column of the 2nd affected port.</li> <li>Type <b>P</b> at the intersection point between the 1st port's row and the 2nd port's column. For example, type <b>P</b> at the intersection of port row D7 and port column E1 to prohibit these ports from connecting dynamically to each other.</li> </ul>
Allow a dynamic connection	<ul style="list-style-type: none"> <li>Using <b>PF10</b> and <b>F11</b> to scroll right and left, display the port column of the 2nd affected port.</li> <li>Type <b>A</b> at the intersection point between the 1st port's row and the 2nd port's column. For example, type <b>A</b> at the intersection of port row D7 and port column E1 to allow these ports to connect dynamically to each other.</li> </ul>
Copy Allows and Prohibits from one port row to another port row	<ul style="list-style-type: none"> <li>Type <b>C</b> on the blank line to the left of the port row you plan to use as a source.</li> <li>Type <b>T</b> on the blank line to the left of the port row you plan to use as a target.</li> </ul>
Name or rename a port	<ul style="list-style-type: none"> <li>Type a port name at the intersection point between the port row and the Name column. For example, type <b>PRINTER1</b> in the Name column of port row C5 to name C5 PRINTER1. It is recommended that the name be in uppercase letters. (Refer to <i>IBM Tivoli System Automation for z/OS Planning and Installation</i> for suggestions for naming ESCON/FICON Director ports.) You can remove a name by typing blanks over it.</li> </ul>
Change the switch device number	<ul style="list-style-type: none"> <li>Type a switch device number in the <b>Target SWCH Number</b> field. (This is a required action if you start with the IBM-supplied default matrix of 0000.)</li> </ul>

## The View Menu

View

- |  |
|--|
| <ol style="list-style-type: none"> <li>1. Hosts...</li> <li>2. CHPIDs....</li> <li>3. Ports or Names...</li> <li>4. Devices...</li> <li>5. Command Results</li> <li>6. Varys...</li> <li>7. SWCH...</li> </ol> |
|--|

If you select the View menu, the subtasks shown on the left are displayed. When the selected view is displayed:

- Use F7 to scroll backward and F8 to scroll forward to view the information.
- Press F3 to return to the display selection screen.

Selection	Option	Action
1. Hosts	The SA z/OS I/O operations base programs that are known to the issuing SA z/OS I/O operations.	Type * on the 1st blank line. See the DISPLAY HOST command in <i>IBM Tivoli System Automation for z/OS Operator's Commands</i> .
	Detailed information about the SA z/OS I/O operations base program specified.	Type the VTAM name of an SA z/OS I/O operations on the 1st blank line. See the DISPLAY HOST command in <i>IBM Tivoli System Automation for z/OS Operator's Commands</i> .
	A list of all SA z/OS I/O operations base programs that have the specified switch allocated to them.	Type a switch device number on the 2nd blank line. See the DISPLAY HOST command in <i>IBM Tivoli System Automation for z/OS Operator's Commands</i> .
2. CHPIDs	A list of all CHPIDs with their associated information	Type * on the blank line. See the DISPLAY CHP command in <i>IBM Tivoli System Automation for z/OS Operator's Commands</i> .
	Detailed information about a specified CHPID	Type a CHPID number on the blank line. See the DISPLAY CHP command in <i>IBM Tivoli System Automation for z/OS Operator's Commands</i> .

Selection	Option	Action
3. Ports	A list of all ports with the same name	Type the name of a named port on the 1st blank line. A generic port name may be used. See the DISPLAY NAME command in <i>IBM Tivoli System Automation for z/OS Operator's Commands</i> .
	Details about a port name or port address on a specified switch	Type either a port address on the 2nd blank line or a port name on the 3rd blank line.  Then, type a switch device number on the 4th blank line. See the DISPLAY PORT command in <i>IBM Tivoli System Automation for z/OS Operator's Commands</i> .
	Information about all ports, only blocked ports, or only statically connected ports on a specified switch	Type A, B, or C on the 5th blank line.  Then, type a switch device number on the 6th blank line. See the DISPLAY PORT command in <i>IBM Tivoli System Automation for z/OS Operator's Commands</i> .
4. Devices	A list of all devices attached to a specified switch	Type * on the 1st blank line.  Then, type a switch device number on the 2nd blank line. See the DISPLAY DEVICE command in <i>IBM Tivoli System Automation for z/OS Operator's Commands</i> .
	Detailed information about a specific device with relation to all switches	Type a device number on the 1st blank line.  Then, type * on the 2nd blank line. See the DISPLAY DEVICE command in <i>IBM Tivoli System Automation for z/OS Operator's Commands</i> .
	Detailed information about a specified device relative to a specified switch	Type a device number on the 1st blank line.  Then, type a switch device number on the 2nd blank line. See the DISPLAY DEVICE command in <i>IBM Tivoli System Automation for z/OS Operator's Commands</i> .
	Multiple devices on a specified switch	Type a range of device numbers on the 1st blank line.  Then, type a switch device number on the 2nd blank line. See the DISPLAY DEVICE command in <i>IBM Tivoli System Automation for z/OS Operator's Commands</i> .

## The View Menu

Selection	Option	Action
5. Command Results	A list of command results with relation to each SA z/OS I/O operations	See the DISPLAY RESULTS command in <i>IBM Tivoli System Automation for z/OS Operator's Commands</i> .
6. Varys	A list of failed vary path requests with relation to a specific SA z/OS I/O operations host	Type the SA z/OS I/O operations VTAM name on the blank line. See the DISPLAY VARY command in <i>IBM Tivoli System Automation for z/OS Operator's Commands</i> .
7. SWCH	A list of all switches allocated, or attached, to the issuing instance of I/O operations.	Type * on the blank line. See the DISPLAY SWITCH command in <i>IBM Tivoli System Automation for z/OS Operator's Commands</i> .
	Details about a specific switch	Type a switch device number on the blank line. See the DISPLAY SWITCH command in <i>IBM Tivoli System Automation for z/OS Operator's Commands</i> .

---

## The Options Menu

### Options

1. Assist...
2. Command...
3. Getlock...
4. Matrix...
5. Unlock...
6. Verify...

In the I/O operations ISPF dialog, a number of options are shipped with IBM-supplied defaults. You can customize these defaults to your ISPF user ID. Except for the **Verify** option, your customized options remain in effect for your user ID across SA z/OS I/O operations sessions.

- Do either of the following:
  - On the dialog's main menu, select the **Options**, and select one of the entries.
  - Enter OPTIONS in the command area of a dialog panel
- Make a selection (1–6).
- If you want to save your change, press **F3** or enter **END**

### 1. Assist

Describes the return codes and reason codes in the messages. Selections are:

- To receive explanatory text and specific messages (IBM-supplied default)
- Not to receive explanatory text on the return codes and reason codes.

### 2. Command

Specifies a set of default connectivity options. Selections are:

- NOForce, Vary, BACkout (IBM-supplied defaults)
- NOForce, Vary, NOBackout
- NOCheck, Vary, BACkout



4. NOCheck, Vary, NOBackout
5. Force, Vary, NOBackout
6. Force NOVary, NOBackout

### 3. Getlock

Obtains processing control. (See the GETLOCK command in *IBM Tivoli System Automation for z/OS Operator's Commands* for more information.)

Selections are:

1. Never take control from another user.
2. Be prompted with the **Getlock** panel so you can decide whether you want to take control (IBM-supplied default).
3. Always take control. If a command is being processed by another user, however, processing can be interrupted and the command can fail.

### 4. Matrix

Specifies the changes on the switch configuration that will be activated by SA z/OS I/O operations. (Select them with any character.) Selections are:

- ALL commands (overrides other options; IBM-supplied default)
- Allow and Prohibit
- Connect and Disconnect
- Block and Unblock
- Port names
- Switch name

### 5. Unlock

Relinquishes control of SA z/OS I/O operations command processing. (See the UNLOCK command in *IBM Tivoli System Automation for z/OS Operator's Commands* for further information.) Selections are:

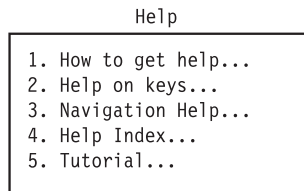
1. No automatic unlock. You must enter the Unlock command.
2. Automatic unlock when you press **F3** or **END** from the I/O operations ISPF main menu (IBM-supplied default).
3. Automatic unlock after every command.

### 6. Verify

Requires you to verify a command before the dialog activates it. Contrary to the other options, you can tailor the **Verify** option for the remainder of the session only. The default for this option resets to the IBM-supplied default at the beginning of each session. Selections are:

1. Verify before activating (IBM-supplied default).
  - If you press **Enter** after entering a connectivity command, the dialog will continue.
  - If you press **F3** or **END**, command processing is halted, and the preceding screen will be displayed.
2. Do not verify.

# The Help Menu



1. How to get help:
  - To get the entries shown above, select the **Help** menu from the I/O operations ISPF main menu.
  - To get field-oriented help, press **F1** on any input field on a screen.
  - To get screen-oriented help, press **F1** on any full screen. If you press **F1** twice in a row, an ISPF help panel, not an I/O operations ISPF help panel, is displayed.
2. Help on keys... explains the I/O operations ISPF dialog settings of the following function keys: F1 (Help); F3 (End); F5 (Save); F6 (Activate).
3. Navigation Help... helps you navigate through the panel, or screen, flow of the I/O operations ISPF dialog.
4. Help Index... lets you choose among a selected number of topics that are listed alphabetically.
5. Tutorial ... Do one of the following:
  - Select the **Help** menu on the main menu. Then:
    - a. Select **Tutorial**
    - b. Select one of the topics shown in on the screen below.
  - Enter TUTORIAL on a panel of the dialog. Then:
    - a. Press **Enter**
    - b. Select one of the topics on the tutorial main menu shown below.

```
SA z/OS - IO-Ops ISPF Tutorial Main Menu
```

```
Select a specific topic by typing the topic number on the command
line and pressing Enter.
```

```
1      Overview of IO-Ops and SWCHs
2      Displaying and Changing SWCH Configuration Data
3      Displaying Other Types of Information
4      Setting Options
5      Guidelines and Precautions
6      Understanding and Using the Matrix Screen
7      Making a Single Change to All SWCHs
8      Connectivity Command Options
```

```
COMMAND ==>
```

```
F1= Help      F2=          F3= END      F4= RETURN  F5=          F6=
F7= Help      F8=          F9=          PF10=       F11= Index  F12=
```

---

## Appendix C. Problem Determination

This appendix contains various subtopics that deal with methods of avoiding or detecting problems during SA z/OS automation. The following topics are discussed:

- “Concurrent User Access within the Customization Dialog”
- “Maintaining System Status during a Status Forwarding Path Failure”
- “Debugging and Tracing” on page 234

---

### Concurrent User Access within the Customization Dialog

The customization dialog allows you the editing of certain entry types by multiple users at the same time. To prevent simultaneous usage of the same resource, a serialization is implemented using the ENQ and DEQ macros with global resources. If an installation runs several systems in a global resource serialization (GRS) complex, but GRS is not installed, it should be ensured that the following major names (qname of ENQ resp. DEQ macro) are propagated:

<b>INGPDB</b>	This name is used by SA z/OS to serialize access within the customization dialog. It is mandatory to have this major name distributed among all systems where the customization dialog may run. Otherwise different users may overwrite their changes and data may get lost.
<b>SYSDSN</b>	This name is used by SA z/OS to obtain the user ID of a resource holder in case a policy database cannot be accessed. If this major name is not distributed, messages indicating that a resource cannot be accessed would not show a user ID.
<b>SPFEDIT</b>	This name is used by SA z/OS to obtain the user ID of a resource holder in case an ISPF table, which holds the data within a policy database, cannot be accessed. If this major name is not distributed, messages indicating that a resource cannot be accessed would not show a user ID.

---

### Maintaining System Status during a Status Forwarding Path Failure

SA z/OS uses processor operations functions and XCF to attempt to maintain the current status of a system, even if there is a failure in the status forwarding path. SA z/OS system operations uses XCF for both passive and active monitoring of system status. As long as there is one SA z/OS in a sysplex with an active status forwarding path, SA z/OS is able to use XCF to maintain system status for all systems in the sysplex.

SA z/OS system operations receives alerts generated by processor operations for target systems monitored by processor operations. If, for example, a target system enters a wait state or a DCCF condition, SA z/OS will use the resulting alert from the processor operations functions to update the system status, even if the status forwarding path between SA z/OS and the target system is down.

---

### Debugging and Tracing

This chapter describes the tracing and debugging facilities that are available with SA z/OS. The following topics are discussed:

- “Automation Manager State Trace Table”
- “Using Trace Services for the Automation Manager”
- “Using AOCTRACE” on page 236
- “How to Use a Log with INGHIST” on page 236
- “How to Use the Diagnostic Option of INGAMS” on page 237
- “Tracing and Debugging for ProcOps and the BCP Internal Interface” on page 242
- “Using Trace Services for I/O Operations” on page 243

To collect debugging information you can also use the command INGLKUP REQ=COLLECT.

#### Automation Manager State Trace Table

The event handler trace back table is written to SYSLOG or to the Message Logger. It is the most important debug tool when you want to understand the event flow. The table is written using the HSAM1399I message due to one of the following:

- Address space termination (written automatically to the Message Logger)
- */F jobname,\$TRACESTATE* is written to SYSLOG

#### Using Trace Services for the Automation Manager

To trace the control flow of a process, SA z/OS uses the **MVS Component Trace Facility** with all its capabilities.

This requires an External Writer as soon as the trace is started. The writer is internally started if it is not already active.

Before the trace can be activated the trace data set must be allocated. The recommended data set name is 'SYS1.HSA.*sysname*.CTRACE01'. An appropriate JCL is provided in the SINGSAMP SA z/OS sample library as the member HSACTWR. In addition, the JCL of the component trace's external writer must be cataloged in SYS1.PROCLIB. For more information on customizing the component trace refer to *IBM Tivoli System Automation for z/OS Planning and Installation*.

To activate the trace on a z/OS console, enter the command:

```
TRACE CT,ON,COMP=HSAAM
```

When the system responds with a WTOR for the trace options, reply as follows:

```
xx,WTR=HSACTWR,OPTIONS=(ALL|set_of_options),END
```

where *set\_of\_options* is one of the following:

Option	Activates the Trace In:
ALL	All of the following options
B[ase]	SA z/OS service routines using z/OS system services
COM[munication]	The communications framework
CON[trol]	The base framework

Option	Activates the Trace In:
E[vent]	The Sysplex Event Handler
F[ramework]	The service sustaining the LE environment
I[nstruction]	The services encoding or decoding the instruction streams
L[ock]	The service creating and deleting locks
MQ	WebSphere MQ services
R[equest]	The beginning and ending of a work item (request) processing
S[chedule]	The service overriding service period definitions
W[orkitem]	All components working on work items

Deactivate the trace with the command:

```
TRACE CT,OFF,COMP=HSAAM
```

This command is automatically issued when the last automation manager in a single image terminates while the trace is active.

Before the trace records can be analyzed by IPCS the trace data set or sets must be made available to IPCS. This is done with the operator command:

```
TRACE CT,WTRSTOP=HSACTWR
```

If the related message indicates that more than one data set contains trace data, you must use the IPCS command COPYTRC to combine the data sets into a new one.

To analyze the trace data, you must use the IPCS command CTRACE. A JCL is provided for the IPCS batch interface. The command supports several parameter and, at the very least, you must adapt the following:

- COMP(HSAAM)
- USEREXIT(HSAPSTUE)
- DSNNAME(*trace\_data\_set\_name*)
- OPTIONS(*(filter\_options)*)

where *filter\_options* is a combination the following items, separated with a comma:

- BASE
- COMMUNICATION
- CONTROL
- EVENT
- FRAMEWORK
- HEALTHCHECK
- INSTRUCTION
- LOCK
- MQ
- REQUEST
- SCHEDULE
- WORKITEM
- AM=*xcf\_member\_name*
- CALLID=(*callid[,callid]*)
- KEY=(*key[,key]*)
- METHOD=(*method[,method]*)
- MODULE=(*module[,module]*)

## Debugging and Tracing

- TASKID=(*taskid*[,*taskid*])
- TOKEN=(*token*[,*token*])

The keyword options are connected by a logical AND. The values of a keyword and the remaining options are connected by a logical OR. Apart from AM, each keyword option supports up to 5 different values separated by a comma or a space. Imbedded spaces or commas in a value require delimiters, such as '...' or "...".

### JCL Sample for Printing the Trace Records

The JCL sample member HSAJPTRC is stored in the SA z/OS SINGSAMP library. The data set names of the STEPLIB, IPCSDDIR, and TRACEDS DDnames must be changed to their actual values before the job is submitted. This is also true for all of the OPTIONS parameters that are listed because the values of the keywords are only sample values.

## Using AOCTRACE

The AOCTRACE command can be used to enable or disable the automation debugging facility, either globally or for specific clists (REXX routines).

The syntax of the AOFTRACE command and additional information can be found in *IBM Tivoli System Automation for z/OS Operator's Commands*.

Information about using AOCTRACE to trace user-written automation procedures is provided in *IBM Tivoli System Automation for z/OS Customizing and Programming*.

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

## How to Use a Log with INGHIST

With the INGHIST command you can specify the request to display a log with automation manager messages. For this purpose, type: INGHIST REQ=LOG on the command line to reach a panel similar to the one shown in Figure 69.

```
INGKYH11          SA z/OS - Command Dialogs          Line
Domain ID = IPUFM  ----- INGHIST -----          Date = 05/26/00
Operator ID = USER1      Sysplex = AOC7PLEX          Time = 11:41:12

Start time ==> 2000-05-26 10:41:00    format: yyyy-mm-dd hh:mm:ss
End time   ==>

                                Logger Messages
Date      Time
-----
2000-05-29 11:06:36          INGXS   AOC7$$$1
      HSAM1308I SA z/OS PRIMARY AUTOMATION MANAGER INITIALIZATION
      COMPLETE, TYPE=COLD.
2000-05-29 11:06:36          INGXS   AOC7$$$1
      HSAM1311I Processing config part,
      'SATEST.PROD.ACF.G0042V00(Z995CSCH)'.SCHEDULE.2000052314260
2000-05-29 11:06:36          INGXS   AOC7$$$1
      HSAM1310I RC=2 REAS=4384 OPCODE=08020001 (MAIN.2)
      ApplyConfiguration.

Command ==>
PF1=Help      PF2=End          PF3=Return      PF6=Roll
              PF9=Refresh      PF12=Retrieve
```

Figure 69. INGHIST Logger Messages

This panel displays automation manager messages that are written to the system logger. Two or more lines are displayed for each message depending on the size of the message text. The first line contains the date and time when the message was issued. It also shows the XCF member name of the automation manager and the XCF group name the automation manager belongs to. The second and following line(s) contain the message.

## How to Use the Diagnostic Option of INGAMS

From the INGAMS command dialog, you can select option **D Diagnostic**. This leads to the panel shown in Figure 70 where you can initiate the following diagnostic functions:

- Writing a snapshot of the state image queue
- Starting or stopping recording
- Displaying work item statistics

```

INGKYAM4                SA z/OS - Command Dialogs
Domain ID = IPSFO      ----- INGAMS -----      Date = 09/15/07
Operator ID = NETOP1   Sysplex = KEY1PLEX          Time = 13:29:22

Specify or revise the following data:

Action ==>  ___      1. Write Snapshot
                   2. Start recording
                   3. Stop recording
                   4. Work item Statistics

Data set name ==> _____

System          ==> _____ System name, domain id or sysplex name

Command ==>
PF1=Help      PF2=End      PF3=Return
                                           PF6=Ro11
                                           PF12=Retrieve
    
```

Figure 70. INGAMS Diagnostic Functions Panel

In this panel’s fields, enter the following information:

### Action

Specify the action to be done. For options 1–3, you need to fill in the **Data set name** and **System** fields.

Enter 4 to display work item statistics. For details about enabling and using work item statistics, see “Work Item Statistics” on page 238.

### Data set name

Specify the name of the data set that will hold the snapshot data.

|  
|  
|  
|  
|  
|  
|

The snapshot data set and the recording data set should be preallocated with RECFM FB and LRECL 80. The snapshot and recording data sets must be sequential files.

It is strongly recommended that you make your allocation of the snapshot data set according to the snapshot size information that you receive when you select option **B Show Details** for an automation manager from the INGAMS command dialog. This displays the information shown in

Figure 71 showing the snapshot size (highlighted in the example).

```

Diagnostic Info
Snapshot size      : 986132
Number of resources  : 521
Number of requests   : 55
Number group requests : 49
    
```

Figure 71. INGAMS Command Dialog, Show Details Panel

**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.

**System**

This is 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.

**Work Item Statistics**

To display work item statistics, you enter 4 in the **Action** field of the INGAMS Diagnostic Functions Panel (Figure 70 on page 237) without specifying a data set name or system. This displays a panel similar to Figure 72.

```

INGKYAM5          SA z/OS - Command Dialogs      Line 1   of 8
Domain ID = IPSFO  ----- INGAMS -----      Date = 09/15/07
Operator ID = NETOP1      Sysplex = KEY1PLEX      Time = 13:29:57

Snapshot: 2007-09-15 13:29:55  Queue: 0          CPU time: 73.960

Cmd: S Details

          Since
Cmd Task      Status (secs) 13:48 -1 -2 -3 -4 -5 -6 -7 -8 -9 -10
-----
- ADMIN      IDLE
- QUERY1     IDLE          4  5  10          2  5 11  6
- QUERY2     IDLE
- QUERY3     IDLE
- UPDATER1   IDLE          1
- UPDATER2   IDLE
- UPDATER3   IDLE
- UPDATER4   IDLE

Command ==>
PF1=Help      PF2=End          PF3=Return      PF6=Roll
                PF9=Refresh      PF12=Retrieve
    
```

Figure 72. INGAMS Work Item Statistics Display

This panel shows history information about the work items processed by the automation manager. The automation manager keeps track of the last 300 work items processed by each of the tasks that build the automation manager kernel.

**Note:** The history information is not persistent or available when the automation manager runs on a system that is running an earlier version of SA z/OS. The following data is shown:



**Snapshot**

The snapshot timestamp shows the time of the query.

**Task** The name of the task.

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

**BUSY** The task is processing a work item.

**IDLE** The task is waiting for work.

**NOTIDLE**

The task has completed work but is not waiting for work.

**UNKNOWN**

History records have not yet been defined for this task.

**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 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 history query (INGAMS DIAG REQ=STATS) was made or refreshed.

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 73.

```

INGKYAM6                SA z/OS - Command Dialogs      Line 1    of 300
Domain ID   = IPZFA      ----- INGAMS -----      Date = 09/28/07
Operator ID = XJIVENS    Sysplex = SATPLEX          Time = 13:51:52

                                          Snapshot: 2007-09-15 13:29:55

Task      Day      Starting Elapsed
          Day      Time     in secs Busy Work item
-----
QUERY1    11:53:36  0.043   GET_RESOURCE_DATA
QUERY1    11:54:11  0.096   GET_RAW_RESOURCE_DATA
QUERY1    11:54:11  0.067   GET_RAW_RESOURCE_DATA
QUERY1    11:54:11  0.069   GET_RAW_RESOURCE_DATA
QUERY1    11:54:12  0.067   GET_RAW_RESOURCE_DATA
QUERY1    11:54:12  0.067   GET_RAW_RESOURCE_DATA
QUERY1    11:54:13  0.067   GET_RESOURCE_DATA
QUERY1    11:54:14  0.066   GET_RESOURCE_DATA
QUERY1    11:54:15  0.067   GET_RESOURCE_DATA
QUERY1    11:54:16  0.066   GET_RESOURCE_DATA
QUERY1    11:54:17  0.067   GET_RESOURCE_DATA
QUERY1    11:59:11  0.091   GET_RAW_RESOURCE_DATA

Command ==>
PF1=Help    PF2=End    PF3=Return    PF6=Ro11
             PF8=Forward PF9=Refresh   PF12=Retrieve
    
```

Figure 73. INGAMS Work Item History Display

**Work Item Lifecycle Recording:** Work item lifecycle recording is an internal diagnostic tool that you should use only if required by SA z/OS service.

Lifecycle recording provides enhanced debugging to track down lost requests during automation agent-automation manager communication and other automation manager-related problems.

Lifecycle recording records checkpoints along the flow of a work item. The following flows are reported:

## Debugging and Tracing

- Work items that flow from an automation agent to the primary automation manager (for example, status updates)
- Work items that flow from an automation agent to the primary automation manager plus the response (for example, INGLIST)
- Orders that flow from the PAM to an automation agent

*Using Lifecycle Recording:* By default, lifecycle recording is disabled. When enabled, the automation agent and the primary automation manager (PAM) write life cycle records to a data space. Each automation agent and the PAM have their own data space. Lifecycle recording must be enabled for the PAM and at least one automation agent. It can be enabled for the PAM from any automation agent, however it must be enabled for the automation agent from that automation agent.

Before lifecycle recording can be enabled, the size of the data space must be defined. This can be done automatically with the INGRXQRY command that enables lifecycle recording.

When the problem has been captured, the data spaces must be off-loaded to external data sets of the same size via the INGRXQRY command. A sequential data set is therefore required for the PAM and each automation agent that lifecycle recording was enabled for.

The allocation of the data sets might be done automatically if the appropriate RACF definitions are given to the NetView that issues the lifecycle recording command.

*Managing Lifecycle Recording with the INGRXQRY Command:* You can enable or disable lifecycle recording by issuing the following INGRXQRY commands in NetView:

```
INGRXQRY LCR ON size;dsname
```

Switches on lifecycle recording for the local automation agent.

```
INGRXQRY LCR OFF
```

Switches off lifecycle recording for the local automation agent.

```
INGRXQRY LCRM ON size;dsname
```

Switches on lifecycle recording for the PAM.

```
INGRXQRY LCRM OFF
```

Switches on lifecycle recording for the PAM.

**Note:** If you issue the command with the size parameter, a new data space is created and the old data space is deleted. The dsname parameter specifies the data set that should be used later when the data space is saved. The total length of *size;dsname* can be a maximum of 60 bytes.

You can offload the data collected for lifecycle recording for the local automation agent or the primary automation manager, as follows:

- **Local automation agent:**
  - Use INGRXQRY LCR SAVE *dsname* if you used INGRXQRY LCR ON *size* and did not specify a data set name or you want to save the data to a new data set.
  - Use INGRXQRY LCR SAVE if you used INGRXQRY LCR ON *size;dsname* to save the data to the data set that you specified.
- **Primary automation manager:**

- Use INGRXQRY LCRM SAVE *dsname* if you used INGRXQRY LCRM ON *size* and did not specify a data set name or you want to save the data to a new data set.
- Use INGRXQRY LCRM SAVE if you used INGRXQRY LCRM ON *size;dsname* to save the data to the data set that you specified.

### Notes:

1. When you offload the data space to a data set make sure that LCR(M) is still ON before you issue SAVE.
2. Only the actual data from data space is copied. Small primary and multiple secondary allocations of the data set reduce the amount of DASD space.
3. If *dsname* does *not* exist it will be automatically created with the appropriate size. If the data set *already* exists it is used as it is. You should there make sure that it is large enough.
4. If *dsname* ends with \*, for example, MY.HLQ.\*, a new data set with a unique name is created using following pattern:  
MY.HLQ.<AA|AM>.<domain>.Dyymmdd.Thhmmss
5. NetView must have RACF ALTER access to the data set. The PAM must have RACF UPDATE access to the data set. Allocation of the new data set is done automatically via the NetView ALLOCATE command. The corresponding NetView return codes are passed back to caller in case of an error.
6. The size of the data space can range from 1–2098 MB. It specifies the maximum size of the data space. Initially a smaller data space is allocated that is extended until the maximum is reached. A value of 500 is recommended and should be sufficient for all situations.
7. You must use a semicolon to separate *size* and *dsname*. No blanks are allowed.

### Examples:

```

INGRXQRY LCR ON 500;MY.AGENT.DATA.SET
INGRXQRY LCR ON 500;MY.AGENT.*
INGRXQRY LCR SAVE
INGRXQRY LCR SAVE MY.NEW.AGENT.DATA.SET
INGRXQRY LCR SAVE MY.HLQ.*

```

*Changes to HSAPRMxx and INGXINIT:* The LIFECYCLE parameter in the INGXINIT and HSAPRMxx members defines the size of the data space and the data set name for offloading the lifecycle recording data:

```
LIFECYCLE=nnnn;dsname
```

This parameter is not mandatory for using lifecycle recording. The same information can be provided later with the INGRXQRY command when lifecycle recording is actually enabled.

### Notes:

1. The LIFECYCLE parameter does not switch on lifecycle recording. This must be done with the INGRXQRY LCR(M) ON command.
2. The LIFECYCLE parameter does not allocate data sets automatically. This is done with the INGRXQRY command only.

*Performing Lifecycle Recording:* Carry out the following steps to perform lifecycle recording:

- Step 1. Allocate data sets:** If necessary, allocate sequential data sets for the primary automation manager and each automation agent to be monitored with the following attributes:

## Debugging and Tracing

- Organization PS
- Record format FB
- Record length 128
- Block size 27904

The data sets must be large enough to hold the data space. The recommendation is 500 MB. The automation agents and the PAM must have write access to their data sets.

**Step 2. Create the data space:** After the automation agent has completed initialization, enable lifecycle recording with the following commands:

- INGRXQRY LCR ON 500;*my.agent.dataset* on each automation agent that is involved in debugging the problem
- INGRXQRY LCRM ON 500;*my.pam.dataset*

If the *size* parameter is present, the old data space is deleted and a new empty data space created. The size and the data set name can also be provided with the LIFECYCLE parameter in the HSAPRMxx and INGXINIT members.

**Step 3. Check, enable or disable lifecycle recording:** You can check the status of lifecycle recording with the following commands:

- INGRXQRY LCR QRY
- INGRXQRY LCRM QRY

If the data space has already been created (through either HSAPRMxx and INGXINIT, or INGRXQRY), you can switch on lifecycle recording with the following commands, as needed:

- INGRXQRY LCR ON
- INGRXQRY LCRM ON

To switch off lifecycle recording, issue the following commands:

- INGRXQRY LCR OFF
- INGRXQRY LCRM OFF

**Step 4. Logging checkpoints:** When you have prepared and started lifecycle recording, recreate the problem situation. After the problem scenario has been duplicated, switch off lifecycle recording. Because the checkpoints are written in wraparound mode, do not unnecessarily delay switching off. This avoids important checkpoints being lost or overwritten.

**Step 5. Save data to data sets and send to SA z/OS service:** Use the following commands to copy the data space to the appropriate data sets that were previously allocated:

- INGRXQRY LCR SAVE on each automation agent that is involved in debugging the problem
- INGRXQRY LCRM SAVE

Send the data sets to SA z/OS service for evaluation and analysis.

## Tracing and Debugging for ProcOps and the BCP Internal Interface

For ProcOps SNMP connections and the BCP Internal Interface trace facilities are provided that can be used for problem determination purposes.

tracing and debugging for ProcOps and the BCP Internal Interface

## Using AOCTRACE

In order to enable the global tracing of all communications related parts of ProcOps-SNMP or BCP Internal Interface, issue the following command:

```
AOCTRACE comprname ON
```

where *comprname* is one of the following:

### INGRX810

INGHWCMD command list

### INGHWCOM

Communications module

To disable the global trace mode issue command:

```
AOCTRACE comprname OFF
```

In global trace mode all currently active connections to SEs or HMCs are traced. This may produce a large amount of data written to the Netlog, depending on the number of connections or the amount event data returned from the hardware. In such cases, use the connection specific trace.

## Tracing Specific Connections

If you want to trace the communication to a specific processor or CPC, issue one of the following service commands:

```
INGHWCMD proname TRACE ON  
INGHWCMD proname UTRACE ON
```

where *proname* is the name of the target hardware or processor that is defined in the SA z/OS customization dialog, and specifying UTRACE traces only the user session of the operator that issued the command.

Use the following service commands to disable the trace:

```
INGHWCMD proname TRACE OFF  
INGHWCMD proname UTRACE OFF
```

Although only a single processor connection is traced, the amount of data written to the Netlog may be high, especially if the processor connection is through an HMC. In this case, data from other processors serviced by this HMC is additionally shown in the trace. It is therefore recommended to limit the tracing time.

## First Failure Data Capture Support

The INGHWCMD communications module supports the First Failure Data Capture debug aid, provided by NetView for high level language command processors, written in PL/I or C. In case of an INGHWCMD abnormal end, message CNM983E, CNM998E, or CNM999E is issued.

Message CNM983E contains the HLL API user trace information that is maintained by INGHWCMD. This information allows you to determine the internal program flow at error time. If available save this debug aid information before contacting IBM for service. For more information about the diagnostic messages, refer to the *Tivoli NetView for z/OS Diagnosis Guide*.

## Using Trace Services for I/O Operations

To trace the control flow of I/O operations, SA z/OS uses the MVS Component Trace Facility with all its capabilities.

## Debugging and Tracing

The external writer is not mandatory when the trace is activated. However, when you specify the external writer on the TRACE CT,ON,... command or the WTR option of the CTIIHVZZ PARMLIB member, the writer is internally started if it is not already active.

Before the external writer can be activated the trace data set must be allocated. The recommended data set name is SYS1.HSA.*sysname*.CTRACE01. An appropriate JCL is provided in the SINGSAMP SA z/OS sample library as the member HSACTWR. In addition, the JCL of the component trace's external writer must be cataloged into SYS1.PROCLIB. For more information on customizing the component trace refer to *IBM Tivoli System Automation for z/OS Planning and Installation*.

If the trace has not been activated by the CTIIHVZZ PARMLIB member, you can activate it on a z/OS console using the command:

```
TRACE CT,ON,COMP=IHVAS
```

When the system responds with a WTOR expecting the trace options reply:

```
xx[,WTR=HSACTWR],OPTIONS=( {MIN|MORE|MAX}, ALL | set_of_options ),END
```

where *set\_of\_options* means a combination of the following:

Option	Header
AM	Switch interface
API	Programming API
AU[DIT]	Event recording
B[ASE]	Common routines
CMD	Connectivity commands
COM[MUNICATION]	VTAM, TCP/IP and internal communication
CON[FIG]	Device and CHannel Path configuration
DB	Updates of the internal database
Q[QUERY]	Display and Query commands
R[ECOVERY]	Recovery
S[CHEDULE]	Command scheduling
XM	Cross-memory transport between the I/O operations address space and other address spaces, such as IOS or TSO

You have to specify the external writer if you want analyze the trace using IPCS.

You can view the trace options that are currently active with the following command:

```
D TRACE,COMP=IHVAS
```

Unless you want to deactivate the trace automatically when I/O operations terminates, you can deactivate with the following command:

```
TRACE CT,OFF,COMP=IHVAS
```

Before the trace records can be analyzed by IPCS the trace data sets must be made available to IPCS. This is done with the operator command TRACE CT,WTRSTOP=HSACTWR. If the related message indicates that more than one data set contains trace data, the IPCS COPYTRC command must used to combine the data sets into a new one.

To analyze the trace data you need to run the IPCS CTRACE command. You can use either option 2.7.1.D or option 6 from the IPCS primary option menu. Or, you can use the IPCS batch interface (refer to “JCL Sample for Printing the Trace Records” on page 236). At least the following parameters that the command supports must be adapted:

- COMP(IHVAS)
- {FULL | SHORT | SUMMARY | TALLY}
- DSNNAME(*trace\_data\_set\_name*)

**Note:** In interactive IPCS this is only required when this is not the default data set name.

- OPTIONS(*(filter\_options)*)

where *filter\_options* is a combination of the following items, separated by a comma:

- AM
- API
- Base
- CMD
- COMMunication
- CONfig
- DB
- Query
- Recovery
- Schedule
- XM
- JOBNAME=(*jobname*[,*jobname*]...)
- ASID=(*asid*[,*asid*]...)
- TASKID=(*taskid*[,*taskid*]...)
- MODULE=(*module*[,*module*]...)
- POS=(*pos*[,*pos*]...)

The keyword options are connected by a logical AND while the values of a keyword and the remaining options are connected by a logical OR. Each keyword option supports up to 10 different values separated by a comma or a space. Imbedded spaces or commas in a value require delimiters '.' or '..'.

Available task IDs are:

```
IAEE INIT
MAIN
SCHA SCHB SCHF
UOBY UOIN
ZDQM ZSCB ZSDT ZSPL
```

The trace records are stored intermediately in a data space before they are externalized by the external writer. The data space can be regarded as a wraparound buffer. If you are interested only in the latest trace records you can use the IHVJCT job rather than the external writer and IPCS. The program allows you to print selected trace records from the data space as defined by the filter options above. The JCL can be found in the SINGSAMP SA z/OS sample library.

## Debugging and Tracing



---

## Appendix D. Hardware Interfaces: Additional Recovery Information

---

### BCP Internal Interface Recovery Information

#### Determine BCP Internal Interface Communication Task Names

The command `INGHWCMD <pname> STATCOM` can be used to determine the name of the autotask for communicating with processor `<pname>`. In the AOFA0099 report string that is returned by this command, the TASK field contains the autotask name.

If the command cannot be applied, perform the following steps:

1. Issue command: `QRYGLOBL COMMON VARS=ING.0HW*`

A list of common global variable name-value pairs is displayed. Look for the processor name value whose communication autotask name you want to know, and make a note of the two digit variable name number `<dd>`. If the processor name is not shown in the list, communication with the processor has not been established.

2. Issue command: `QRYGLOBL COMMON VARS=AOF.0HWOPER <dd>`

A common global variable name-value pair is displayed with the autotask name stored as the value.

#### BCP Internal Interface Recovery Actions

The following section explains manual recovery procedures that might be necessary in case the BCP internal interface communication module INGHWC.COM is not responding or is unavailable.

Interface report messages that indicate that module INGHWC.COM is unavailable are:

1. AOFA0000 Check Task: `<tn>_task_module_INGHWC.COM_not_running`
2. AOFA0000 Check Task: `<tn>_reached QueueLimit_<ql>`
3. AOFA0000 Check Task: `<tn>_task_is_not_available`

The recovery procedure for each of these cases is outlined below.

##### Recovery Case 1: `<tn>_task_module_INGHWC.COM_not_running`

The recovery procedure for this case is as follows:

1. Make a note of the autotask name `<tn>` and browse the netlog for messages to or from that task.
2. Search for DSI172I messages that have `<tn>` as the abending subtask and look for INGHWC.COM as the failing module. If this is the case, note down the last report message (AOFA\*) prior to the abend from that task.
3. Issue the NetView command: `EXCMD <tn> LOGOFF` (where `tn` is the autotask name that you made a note of in step 1).

This will terminate the autotask and free its task storage that was allocated by INGHWC.COM. If the autotask has been defined as an HWOPER`xx` task in the SA z/OS AUTOOPS policy entry, it will be automatically restarted and activate the communication module INGHWC.COM.

## BCP Internal Interface Recovery Actions

### Recovery Case 2: `<tn>_reached QueueLimit_<ql>`

Each Support Element connection uses a separate autotask. As soon as the "reached QueueLimit" message is returned, this indicates that the previously issued HW request has not yet sent a completion report. The normal recovery process would be to retry the HW command request at a later time.

There are, however, cases where module INGHWCOM has to wait for an indefinite period of time until the HW has completed the request. Usually, this occurs when a Support Element session has been started in SYNC processing mode while a long-running HW command such as ACTIVATE or CBU is still active

If necessary, you can interrupt and terminate the session, which will not affect any HW command currently in progress on the Support Element. If you terminate the session, you will not receive any completion information for this command.

Issue the following NetView commands:

1. EXCMD `<tn>` RESET
2. TASKUTIL `<tn>`
3. EXCMD `<tn>` LOGOFF

Command 1 terminates the communication module INGHWCOM in task `<tn>`. The session between NetView and the SE/HMC will be closed normally and the allocated task storage will be freed.

With command 2 you can control whether module INGHWCOM is no longer the active command of `<tn>` and the amount of storage still allocated by `<tn>`.

With command 3 you terminate autotask `<tn>`, which will initiate an autotask restart and the activation of INGHWCOM.

If command 2 indicates that command 1 did not end INGHWCOM, command 1 can be repeated with RESET IMMED. This will abnormally terminate INGHWCOM, but SE/HMC session termination will not be performed. Note that a 'STOP TASK' command will end the autotask, but will not invoke normal SE/HMC session termination.

### Recovery Case 3: `<tn>_task_is_not_available`

There are cases where a BCP internal interface autotask, previously defined as an HWOPER`xx` task with an SA z/OS AUTOOPS policy entry, cannot be used. In these cases, a task definition problem or a mismatch of your current ACF configuration might be the reason for the problem.

The recovery procedure for this case is as follows:

1. Verify the AUTOOPS definition statements in your PDB.
2. Verify that the corresponding DSIOFP statements have AOFPRFHW defined as the operator profile name.
3. Perform a Build operation after the policy changes.
4. Activate your policy changes, for example by performing an ACF REFRESH.

### Terminating NetView and the BCP Internal Interface

When terminating NetView, a 'CLOSE NORMAL' command should first be issued to give INGHWCOM enough time to end all its SE/HMC sessions. Then, after a predefined period of time, a 'CLOSE IMMED' command could be used to limit NetView's overall termination time.

---

## How to Switch the IP Address for Target Hardware Used by Processor Operations

This description is only valid for target hardware (defined as 'Processors' in the SA z/OS customization dialog) with a connection type of SNMP.

When defining a processor as the target hardware for Processor Operations, two IP addresses can be provided for communication with either the Support Element (SE) or the Hardware Management Console (HMC). At least one IP address for the SE or HMC must be provided; optionally, a secondary IP address can be provided for this processor. If two IP addresses are provided and both addresses are SE ones, ensure that the SE has an alternate network adapter card installed and that it is operational.

If you need to switch between these two addresses (for example, in case of a SE network adapter card failure), switching to the secondary IP address can be easily performed using Processor Operations command:

```
ISQIPSWT <target-hardware-name>
```

For a detailed command description, refer to *IBM Tivoli System Automation for z/OS Operator's Commands*.

---

## How to Terminate BCP Internal Interface Connections or Clear Hardware Control Blocks after a Connection Failure

There are occasions when it may be necessary to terminate BCP internal interface connections, for example, prior to issuing a CLOSE IMMED to terminate IBM Tivoli NetView for z/OS and IBM Tivoli System Automation for z/OS.

This can be done using the following Systems Operations command:

```
INGHWSRV TERM
```

This command performs a TERMCOM request for each CPC that is found and then attempts to disable the HSAET32 resource.

Occasionally it may be necessary to clear hardware control blocks after a connection failure with the target hardware.

This can be done using the following Systems Operations command:

```
INGHWSRV target_hardware_name FORCE
```

For a detailed command description, see *IBM Tivoli System Automation for z/OS Operator's Commands*.

See also the appendix "Controlling Access to the Processor Hardware Functions" in *IBM Tivoli System Automation for z/OS Planning and Installation*.

## BCP Internal Interface Recovery Actions

---

## Glossary

This glossary includes terms and definitions from:

- The *IBM Dictionary of Computing* New York: McGraw-Hill, 1994.
- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies can be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018. Definitions are identified by the symbol (A) after the definition.
- 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.** Automation control 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, 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.

**Address Space Workflow.** In RMF, a measure of how a job uses system resources and the speed at which the job moves through the system. A low workflow indicates that a job has few of the resources it needs and is contending with other jobs for system resources. A high workflow indicates that a job has all the resources it needs to execute.

**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.

**alert.** (1) 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. (2) 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 entry for NPDA focal-point system under *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.** (1) Automation Manager Configuration (2) The Auto Msg Classes entry type

**APF.** Authorized program facility.

**API.** Application programming interface.

**APPC.** Advanced program-to-program communications.

**application.** An z/OS subsystem or job monitored by SA z/OS.

**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.

**ApplicationGroup entry.** A construct, created with the customization dialogs, used to represent and contain policy for an application group.

**application program.** (1) A program written for or by a user that applies to the user's work, such as a program that does inventory or payroll. (2) A program used to connect and communicate with stations in a network, enabling users to perform application-oriented activities.

**ARM.** 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 (American National Standard Code for Information Interchange).** The standard code, using a coded character set consisting of 7-bit coded characters

(8-bit including parity check), for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters. (A)

**ASF.** Automation status file.

**authorized program facility (APF).** A facility that permits identification of programs that are authorized to use restricted functions.

**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.

**automated console operations (ACO).** The concept (versus a product) of using computers to perform a large subset of tasks ordinarily performed by operators, or assisting operators in performing these tasks.

**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 data set that consists of:

- the automation control file (ACF)
- the automation manager configuration file (AMC)
- the NetView automation table (AT)
- the MPFLSTSA 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 on which SA z/OS is installed. 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.

**automation manager.** 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.

**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 status file.** 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.

## B

**basic mode.** A central processor mode that does not use logical partitioning. Contrast with *logically partitioned (LPAR) mode*.

**BCP Internal Interface.** Processor function of CMOS-390, zSeries processor families. It allows the 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.

**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, timers, and 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.

**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.

**channel-attached.** (1) Attached directly by I/O channels to a host processor (for example, a channel-attached device). (2) Attached to a controlling unit by cables, rather than by telecommunication lines. Contrast with *link-attached*. Synonymous with *local*.

**CI.** Console integration.

**CICS/VS.** Customer Information Control System for Virtual Storage.

**CLIST.** 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 SYSC clone 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).** (1) 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. (2) 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 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 routine.** One of several SA z/OS programs that perform frequently used automation functions. Common routines can be used to create new automation procedures.

**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.

**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.** Central processor.

**CPC.** 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.** Command processor control block; an I/O operations internal control block that contains information about the command being processed.

**CPU.** Central processing unit. Deprecated term for *processor*.

**cross-system coupling facility (XCF).** XCF is 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.** 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 it passes through a switch.

## D

**DASD.** 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.** Double-byte character set.

**DCCF.** Disabled console communication facility.

**DCF.** 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.

**Devices.** You can see information about all devices (such as printers, tape or disk drives, displays, or

communications controllers) 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 in which the access time is effectively independent of the location of the data; for example, a disk.

**disabled console communication facility (DCCF).** A z/OS component that provides limited-function console communication during system recovery situations.

**display.** (1) To present information for viewing, usually on the screen of a workstation or on a hardcopy device. (2) Deprecated term for *panel*.

**disk operating system (DOS).** (1) An operating system for computer systems that use disks and diskettes for auxiliary storage of programs and data. (2) 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).

**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.

**domain.** (1) An access method and its application programs, communication controllers, connecting lines, modems, and attached workstations. (2) 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 by means of 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.** Extended binary-coded decimal interchange code. A coded character set consisting of 8-bit coded characters.

**ECB.** Event control block. A control block used to represent the status of an event.

**EMCS.** Extended multiple console support.

**enterprise.** An organization, such as a business or a school, that uses data processing.

**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.

**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.** Enterprise Systems Architecture.

**eServer.** Processor family group designator used by the SA z/OS customization dialogs to define a target hardware as member of the zSeries or 390-CMOS processor families.

**event.** (1) In NetView, a record indicating irregularities of operation in physical elements of a network. (2) An occurrence of significance to a task; for example, the completion of an asynchronous operation, such as an input/output operation. (3) Events are part of a trigger condition, in a way that if all events of a trigger condition have occurred, a *STARTUP* or *SHUTDOWN* of an application is performed.

**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 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.

**focus host.** A processor with the role in the context of a unified system image

**focal point system.** (1) 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. (2) **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. (3) **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*. (4) **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. (5) **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. (6) **Status focal point system.** In NetView, the system to which STATMON, VTAM and NLDM 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. (7) **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*.

## G

**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.

**generic 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.

**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.

## H

**Hardware Management Console.** A console used by the operator to monitor and control a System/390 microprocessor cluster.

**Hardware Management Console Application (HWMCA).** A direct-manipulation object-oriented graphical user interface that provides single point of control and single system image for hardware elements. HWMCA provides customer 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 customer 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 does not reflect the structure of any particular computer or operating system. For the NetView program, the high-level languages are PL/I and C.

**HLL.** High-level language.

**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.

**host (primary processor).** The processor at which you enter a command (also known as the *issuing processor*).

**HWMCA.** Hardware Management Console Application. Application for the graphic hardware management console that monitors and controls a central processor complex. It is attached to a target processor (a system 390 microprocessor cluster) as a dedicated system console. This microprocessor uses OCF to process commands.

## I

**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/VS.** 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/Virtual Storage (IMS/VS).** A database/data communication (DB/DC) system that can manage complex databases and networks. Synonymous with IMS.

**INGEIO PROC.** The I/O operations default procedure name; part of the SYS1.PROCLIB.

**initial program load (IPL).** (1) The initialization procedure that causes an operating system to commence operation. (2) The process by which a configuration image is loaded into storage at the beginning of a workday or after a system malfunction. (3) 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 file and that prerequisite applications are functional.

**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.

**interested operator list.** The list of operators who are to receive messages from a specific target system.

**internal token.** A *logical token* (LTOK); name by which the I/O resource or object is known; stored in IODF.

**IOCDs.** I/O configuration data set. The data set that describes the I/O configuration.

**I/O Ops.** I/O operations.

**IOSP.** Input/Output Support Processor.

**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 resource number.** Combination of channel path identifier (CHPID), device number, etc. See internal token.

**IPL.** Initial program load.

**ISA.** Industry Standard Architecture.

**ISPF.** Interactive System Productivity Facility.

**ISPF console.** From this 3270-type console you are logged onto ISPF to use the runtime panels for I/O operations and SA z/OS customization panels.

**issuing host.** See *primary host*; the base program at which you enter a command for processing.

## J

**JCL.** Job control language.

**JES.** Job entry subsystem.

**job.** (1) 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. (2) 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).** A facility for spooling, job queuing, and managing I/O. In SA z/OS publications, JES refers to JES2 or JES3, unless distinguished as being either one or the other.

## K

**Kanji.** An ideographic character set used in Japanese. See also *double-byte character set*.

## L

**LAN.** Local area network.

**line mode.** A form of screen presentation in which the information is presented a line at a time in the message area of the terminal screen. Contrast with *full-screen mode*.

**link.** (1) 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). (2) 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 for zSeries and S/390.** 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).** (1) A network in which a set of devices is connected for communication. They

can be connected to a larger network. See also *token ring*. (2) A network in which communications are limited to a moderately-sized geographic area such as a single office building, warehouse, or campus, and that do not generally extend across public rights-of-way.

**logical partition (LP).** A subset of the processor hardware that is defined to support an operating system. See also *logically partitioned (LPAR) 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 (PU)* and *system services control point (SSCP)*.

**logical unit (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) a peer relationship between session partners, (b) efficient use of a session for multiple transactions, (c) comprehensive end-to-end error processing, and (d) a generic application program interface (API) consisting of structured verbs that are mapped into a product implementation. Synonym for advanced program-to-program communications (APPC).

**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.** Logical partition.

**LPAR.** Logically partitioned (mode).

**LU.** Logical unit.

**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.

**LU 6.2.** 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.

## M

**MAT.** Deprecated term for NetView Automation Table.

**MCA.** Micro Channel\* architecture.

**MCS.** Multiple console support.

**member.** A specific function (one or more modules/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 on which to automate, suppress from the z/OS console display, or both, 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.** Message processing facility.

**MPFLSTSA.** The MPFLST member that is built by SA z/OS.

**Multiple Virtual Storage (MVS).** An IBM licensed program. MVS, which is the predecessor of OS/390, is an operating system that controls the running of programs on a System/390 or System/370 processor. MVS includes an appropriate level of the Data Facility Product (DFP) and Multiple Virtual Storage/Enterprise Systems Architecture System Product Version 5 (MVS/ESA SP5).

**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 z/OS images reside in one or more processors, and programs on one image can communicate with programs on the other images.

**MVS.** Multiple Virtual Storage, predecessor of z/OS.

**MVS image.** A single occurrence of the MVS/ESA operating system that has the ability to process work.

**MVS/JES2.** Multiple Virtual Storage/Job Entry System 2. A z/OS 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.

**MVS/ESA.** Multiple Virtual Storage/Enterprise Systems Architecture.

## N

**NAU.** (1) Network accessible unit. (2) Network addressable unit.

**NCCF.** Network Communications Control Facility.

**NCP.** (1) Network Control Program (IBM licensed program). Its full name is Advanced Communications Function for the Network Control Program. Synonymous with *ACF/NCP*. (2) Network control program (general term).

**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.

**network accessible unit (NAU).** 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*.

**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 (NCCF) console.** A 3270-type console for NetView commands and runtime panels for system operations and processor operations.

**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 in which NetView records events pertaining to NetView and SA z/OS activities.

**NetView message table.** See *NetView automation table*.

**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-NetView task (NNT).** The task under which a cross-domain NetView operator session runs. 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.

**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.

**network.** (1) An interconnected group of nodes. (2) In data processing, a user application network. See *SNA network*.

**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.

**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.

**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.

**NGMF.** Deprecated term for NetView Management Console.

**NGMF focal-point system.** Deprecated term for NMC focal point system.

**NIP.** 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.** 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.** Network Problem Determination Application.

**NPDA focal-point system.** See *focal-point system*.

**NTRI.** 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.

## O

**objective 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*.

**OCF.** Operations command facility.

**OCF-based processor.** A central processor complex that uses an operations command facility for interacting with human operators or external programs to perform operations management functions on the CPC.

**OPC/A.** Operations Planning and Control/Advanced.

**OPC/ESA.** Operations Planning and Control/Enterprise Systems Architecture.

**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. (T)

**operations.** The real-time control of a hardware device or software function.

**operations command facility (OCF).** A facility of the central processor complex that accepts and processes operations management commands.

**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/ESA (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.** (1) A person who keeps a system running. (2) A person or program responsible for managing activities controlled by a given piece of software such as z/OS, the NetView program, or IMS. (3) A person who operates a device. (4) In a language statement, the lexical entity that indicates the action to be performed on operands.

**operator console.** (1) A functional unit containing devices that are used for communications between a computer operator and a computer. (T) (2) 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. (3) 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.

**OS.** Operating system.

**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.

**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.

**OST.** 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.** (1) The portion of a panel that is shown on a display surface at one time. (2) To transfer instructions, data, or both between real storage and external page or auxiliary storage.

**panel.** (1) 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. (2) 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*. (3) 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.** (1) A variable that is given a constant value for a specified application and that may denote the application. (2) 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. (3) Data passed to a program or procedure by a user or another program, namely as an operand in a language statement, as an item in a menu, or as a shared data structure.

**partition.** (1) A fixed-size division of storage. (2) In VSE, a division of the virtual address area that is available for program processing. (3) 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*, *MP*, *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.** Processor controller. Also known as the “support processor” or “service processor” in some processor families.

**PDB.** Policy Database

**PDS.** 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, water, and maintenance boundaries. Contrast with *single-image (SI) configuration*.

**POI.** 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 database where the automation policy is recorded. Also known as the PDB.

**POR.** Power-on reset.

**port.** (1) System hardware to which the I/O devices are attached. (2) On 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. (3) An access point (for example, a logical unit) for data entry or exit. (4) A functional unit of a node through

which data can enter or leave a data network. (5) 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. (6) power-on reset (POR) (7) 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.

**PP.** Physically partitioned (configuration).

**PPT.** Primary POI task.

**primary host.** The base program at which you enter a command for processing.

**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 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 operations control file.** Named by your system programmer, this file contains configuration and customization information. The programmer records the name of this control file in the processor operations file generation panel ISQDPG01.

**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 *LPAR*.

**ProcOps.** 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 DB2, CICS, IMS, OPC (formerly called *features*).

**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.

**PR/SM.** Processor Resource/Systems Manager.

**PSM.** ProcOps Service Machine.

**PU.** Physical unit.

## R

**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 requester is a workstation software, which enables users to log on to a domain, that is, to the server(s) belonging to this domain, and use the resources in this domain. After the log on to a domain, users can access the shared resources and use the processing capability of the server(s). Because the bigger part of shared resources is on the server(s), users can reduce hardware investment.

**resource.** (1) 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. (2) In NetView, any hardware or software that provides function to the network. (3) 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 Monitoring Facility (RMF) Monitor III.** A program that measures and reports on the availability and activity of system hardware and software resources, such as processors, devices, storage, and address spaces. RMF can issue online reports about system performance problems as they occur.

**Resource Object Data Manager (RODM).** A data cache manager designed to support process control and automation applications. RODM provides an in-memory data 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.** SA z/OS-provided automation 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 control file.

**Restructured Extended Executor (REXX).** An interpretive language used to write command lists.

**return code.** A code returned from a program used to influence the issuing of subsequent instructions.

**REXX.** Restructured Extended Executor.

**REXX procedure.** A command list written with the Restructured Extended Executor (REXX), which is an interpretive language.

**RMF.** Resource Measurement Facility.

**RODM.** Resource Object Data Manager.

## S

**SAF.** Security Authorization Facility.

**SA IOM.** System Automation for Integrated Operations Management

**SA z/OS.** 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 an 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 display 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.** Status Display Facility.

**SDLC.** Synchronous data link control.

**SDSF.** 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.

**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 processor (SVP).** The name given to a processor controller on smaller System/370 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 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).** An IP based industry standard protocol to monitor and control resources in an IP network.

**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-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.** Service language command.

**SMP/E.** System Modification Program Extended.

**SNA.** 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.** Simple Network Management Protocol (a TCP/IP protocol). A protocol that allows network management by elements, such as gateways, routers, and hosts. This protocol provides a means of communication between network elements regarding network resources.

**solicited message.** An SA z/OS message that directly responds to a command. Contrast with *unsolicited message*.

**SSCP.** System services control point.

**SSI.** 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 at which a subsystem or application is started.

**status.** The measure of the condition or availability of the resource.

**status focal-point system.** See *focal-point system*.

**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.

**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. See cache structure, list structure, and lock structure.

**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.** Situations where the physical sysplex has been divided into subentities, for example, a test sysplex and a production sysplex. This may be done to isolate the test environment from the production environment.

**subsystem.** (1) A secondary or subordinate system, usually capable of operating independent of, or asynchronously with, a controlling system. (2) In SA z/OS, an z/OS application or subsystem defined to SA z/OS.

**subsystem interface.** 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.** Service processor.

**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.

**switch identifier.** The switch device number (swchdevn), the logical switch number (LSN) and the switch name

**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.** 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.** (1) 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. (2) 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.** (1) A console, usually having a keyboard and a display screen, that is used by an operator to control and communicate with a system. (2) 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. (3) 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.

**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.

**System/390 microprocessor cluster.** A configuration that consists of central processor complexes (CPCs) and may have one or more integrated coupling facilities.

## T

**TAF.** 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.** (1) 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. (2) 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.** (1) A basic unit of work to be accomplished by a computer. (2) 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 *communications task*, *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).** (1) 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. (2) 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 time sharing from remote stations. It is an interactive service on z/OS, MVS/ESA, and MVS/XA.

**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 command that issues a command or command processor (list of commands) at a specified time or time interval.

**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.** 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.** 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.** TSO Extensions.

## U

**UCB.** The unit control block; an MVS/ESA data area that represents a device 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. Contrast with *solicited message*.

**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. See also *Workflow*.

## 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 Storage Extended (VSE).** An IBM licensed program whose full name is Virtual Storage Extended/Advanced Function. It is an operating system that controls the execution of programs.

**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/ESA.** Virtual Machine/Enterprise Systems Architecture.

**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.

**volume.** A direct access storage device (DASD) volume or a tape volume that serves a system in an SA z/OS enterprise.

**volume entry.** A construct, created with the customization dialogs, used to represent and contain policy for a volume.

**volume group.** A named set of volumes. A volume group is part of a system definition and is used for monitoring purposes.

**volume group entry.** A construct, created with the customization dialogs, used to represent and contain policy for a volume group.

**Volume Workflow.** The SA z/OS Volume Workflow variable is derived from the RMF Resource Workflow definition, and is used to measure the performance of volumes. SA z/OS calculates Volume Workflow using:

$$\text{Volume Workflow \%} = \frac{\text{accumulated Using}}{\text{accumulated Using} + \text{accumulated Delay}} * 100$$

The definition of **Using** is the percentage of time when a job has had a request accepted by a channel for the volume, but the request is not yet complete.

The definition of **Delay** is the delay that waiting jobs experience because of contention for the volume. See also *Address Space Workflow*.

**VSE.** Virtual Storage Extended.

**VTAM.** 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*.

**workflow.** See *Address Space Workflow* and *Volume Workflow*.

**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.** Write-to-Operator.

**WTOR.** 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.** 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.** Extended recovery facility.

## Numerics

**390-CMOS.** Processor family group designator used in the SA z/OS processor operations documentation and in the online help to identify any of the following S/390 CMOS processor machine types: 9672, 9674, 2003, 3000, or 7060. SA z/OS processor operations uses the

OCF facility of these processors to perform operations management functions. See *OCF-based processor*.



---

# Index

## A

ABENDING automation status 40  
ACF 11  
ACF REFRESH 61  
ACTIVE automation status 41  
additional BCP internal interfaces 247  
advanced automation options  
  application monitor 51  
alert 201, 202  
alert adapter 209  
alerts 202  
  monitoring 114  
alternate CDS  
  allocating 132  
  spare volumes 132  
  switch to primary CDS 133  
  turn into primary CDS 133  
AMC file 61  
anchors 182, 183  
  lost&found 185  
AOCTRACE command 236, 243  
application group 11  
  nature 89  
  working with 88  
application group information 72, 77  
application monitor status 46  
application move 23, 90  
  resource aware, example 32  
  resource aware, introduced 31  
  resource aware, overview 31  
application move, resource aware 124  
applications 11  
  starting and stopping 58  
AUTODOWN automation status 41  
Automatic Restart Manager  
  statuses 47  
Automatic Restart Manager status 47  
automation agent 13, 55  
automation agent flags 33, 86  
automation agent status 40  
  changing 95  
automation agent view with  
  DISPSTAT 73  
automation configuration file  
  identifying 62  
  reloading 61  
automation control file 8, 11  
Automation flag 33  
automation flags 33  
  Automation 33  
  automation agent flags 33  
  determining which are on 86  
  displaying 87  
  global automation flag 34  
  Initstart 33  
  override settings 88  
  Recover 33  
  Restart 34  
  Shutdown  
    *See* automation flags, Terminate  
  Terminate 34

automation initialization options 219  
automation manager 13  
  diagnostic information 55  
  interaction 94  
  making it primary 54  
  operation mode 94  
  retrieve information from 17  
  starting 53  
  stopping 54  
  stopping with different options 54  
  trace services for 234  
automation manager and automation  
  agents cooperation 21  
automation manager commands  
  INGMOVE 92  
  INGSTR 138  
automation manager configuration file 8  
  identifying 63  
automation manager messages  
  log 236  
automation manager status 13  
automation operator 11  
automation policy 8, 11  
automation status 35, 37  
automation table 62  
  replacing or reloading 62  
automation, setting on or off 88  
autotasks 11  
AUTOTERM automation status 41

## B

backup  
  disruptive 26  
  non-disruptive 26  
  server battery 28  
basic mode 6  
BCP internal interface  
  communication task names,  
    determining 247  
  recovery actions 247  
  terminating after recovery  
    actions 248  
BCP Internal Interface  
  First Failure Data Capture  
    support 243  
  tracing and debugging 242  
  tracing specific connections 243  
  using AOCTRACE 243  
BCP internal interfaces  
  additional recovery information 247  
BREAKING automation status 41  
BROKEN automation status 42

## C

central processor 6  
CFRM policy  
  pending changes 130  
  switching 133

changing automation agent status 95  
changing policy for resources 90  
channel  
  types 8  
channel paths 158  
CLIST 11  
clocks  
  time of day 4  
colors  
  gateway sessions 117  
  in DISPSTAT 116  
  in SDF 116  
  WTOR 117  
command  
  dialogs 68  
  issuing 67  
Command Dialogs Main Menu 68  
command list (CLIST) 11  
command processor 11  
Command Profile Editor 67  
commands  
  AOCTRACE 236  
  DISPGW 140  
  DISPSTAT 73  
  INGLKUP 140  
  INGMOVE 92  
  INGNTFY 99  
  INGPLEX SDUMP 136  
  INGPLEX SLIP 137  
  INGPLEX SVCDUMP 136  
  INGPLEX SYTEM, WLM-related  
    data 124  
  INGRPT 77  
  INGSEND 140  
  INGSTR 138  
  INGTHRES 139  
  ISQCMON 98  
  ISQTRACE 170  
  ISQVARS 109, 110, 111  
  ISQXDST 98, 108  
  ISQXMON 98  
  issuing via NetView Management  
    Console 179  
    RMTCMD 140  
    ROUTE 140  
    SDF 115, 121, 122  
    SETSTATE 95  
    SETTIMER 94  
commands from NMC 188  
communicating  
  with other systems 140  
communications path 7  
  monitoring status 112  
communications task 7  
compound status 35, 39  
concurrent user access 233  
conflicting requests 20  
connections, tracing specific 243  
connectivity changes 158  
control units 7

- cooperation of automation manager and automation agents 21
- correlation rules 209
- couple data set (CDS)
  - alternate CDS
    - allocating 132
    - spare volumes 132
    - turn into primary CDS 133
- couple data sets 132
- coupling facilities 8
  - channels 10
- CPC activation profiles 173
- CPE 67
- critical threshold 139
- cross-domain session 142
- CTLDOWN automation status 42

## D

- DASD 8
- debugging 234
  - BCP Internal Interface 242
  - First Failure Data Capture support 243
  - ProcOps 242
- dependency information 76
- dependency relationships 20
- desired status 35, 38
- determining
  - BCP internal interface communication task names 247
- devices 8, 158
- DIAGINFO 55
- diagnostic functions 237
- diagnostic information for the automation manager 55
- dialog
  - ProcOps 165
- Discovery Library Adapter
  - configuring 164
  - discovery scope 163
  - IdML book, transferring 164
  - INGCNTL command 164
  - INGDLA command 163
  - introduction 163
  - overview 163
  - process 163
- discovery process 163
- DISPAPG 72, 77
- DISPASF 139, 140
- DISPERRS 139, 140
- DISPEVT 85
- DISPEVTS 85, 86
- DISPFLGS 87
- DISPINFO 76
- displaying
  - additional OMEGAMON session information 106
  - automation flags 87
  - flag override settings 88
  - interested operator list 98
  - NetView connection path status 112
  - OMEGAMON sessions 104
  - outstanding WTORs 97
  - statistical information with INGRPT 77
  - status with DISPSTAT 73

- displaying (*continued*)
  - status with SDF 115
  - work item statistics 238
- DISPMTR 72, 77
- DISPNTFY (Display Notification Operator Settings) 99
- DISPSCHD command dialog 88
- DISPSFLT 75
- DISPSTAT 73, 86
- DISPTREE 76
- DISPTRG 85, 154
- disruptive restoration 27
- DLA
  - See Discovery Library Adapter
- DOWN automation status 42
- duplexing
  - system-managed 131
  - user-managed 131
- dynamic switches 7
- dynamic takeover file 56

## E

- enabling
  - resource aware application move 32
- ENDED automation status 42
- ENDING automation status 42
- enterprise systems connection (ESCON) 7
- error thresholds 139
  - set or change 140
- errors
  - display occurred 140
- ESCON
  - converter 7
  - director 7
  - switch 7
- event information
  - DISPEVT 85
  - DISPEVTS 85
  - INGEVENT 85
- Event/Automation Service 201
- events (NMC) 183
- exception views 192, 196
- EXTSTART automation status 43

## F

- FALLBACK 117
- FALLBACK automation status 43
- fiber channel connection (FICON) 7
- FICON
  - converter 7
  - director 7
  - switch 7
- filters
  - DISPSFLT 75
  - INGFILT 70
- focal point monitoring 115
- focal point processor 6

## G

- gateway sessions
  - colors 117
  - determining status of 145

- gateway sessions (*continued*)
  - monitoring with SDF 115
  - overview 144
- GDG 62
- global automation flag 34
- global resource serialization 233
- goal driven automation 14
- goal, defined 18
- goals
  - creation of 14
- grouped resource objects 12
- GRS 233
- guest machines
  - processor operations support 165
- guest systems hierarchy 165
- guests summary panel ISQETRGG 168

## H

- HALFDOWN automation status 43
- HALTED automation status 43
- Handling Jobs that are not Controlled by SA z/OS 140
- hardware configuration overview 4
- hardware monitoring 100
- health performance
  - monitoring 102
- health status 35, 39
- heartbeat 184
- heartbeat function 188
- hierarchy
  - guest systems 165
  - preferences 27
- hold flag 35
- HSACTWR 234
- HSAJPTRC 236

## I

- I/O operations 4
  - connectivity changes 158
  - introduction 157
  - starting and stopping 57
  - using trace services 243
- I/O operations commands
  - using the z/OS operator console 160
- I/O operations ISPF dialog 160, 223
  - help 232
  - tutorial 232
- I/O operations ISPF Main Menu 223
- I/O resources 158
- image 7
- INGAMS 94, 238
  - diagnostic option 237
- INGAMS REFRESH 61
- INGAUTO 88
- INGDLA command 163
- INGDLG 223
- INGEIO 58
- INGEVENT 82, 85
- INGFILT 70
- INGGROUP 89
- INGHIST 83
  - display a log 236
- INGINFO 71, 149
- INGLIST 69, 147, 149

- INGLKUP 140
- INGMOVE command 92
- INGNTFY 99
- INGNTFY command 99
- INGNTFY command dialog 99
- INGPDB 233
- INGPLEX 122
- INGPLEX CDS 132
- INGPLEX CF 126
- INGPLEX command
  - SDUMP 136
  - SLIP 137
  - SVCDUMP 136
  - SYSTEM, WLM-related data 124
- INGPLEX CONsole 124
- INGPLEX Dump 135
- INGPLEX IPL 135
- INGPLEX SDUMP 136
- INGPLEX SVCDUMP 136
- INGPLEX SYStem 123
- INGPOST 183
- INGRELS 73
- INGREQ 80
- INGRPT 77
- INGRTCMD 191
- INGRXQRY command 240
- INGSCHED 84, 151
- INGSEND command 141
- INGSESS command 103
- INGSET 34, 35
- INGSTR command 138
- INGTHRES 139, 140
- INGTHRES command 139
- INGTOPO 187
- INGTRIG 85
- INGVOTE 82, 148
- Initialize automation flag 33
- initializing SA z/OS 53
- integrated system channel 10
- interested operator list 97
  - adding your ID 98
  - displaying 98
  - join or leave 98
- ISPF dialog 160, 223
- ISQ900I message 98
- ISQ901I message 98
- ISQCMON 98
- ISQEHARP
  - target hardware summary panel 166
- ISQEPNVC
  - PATH detail panel 169
- ISQESUM
  - status summary panel 166
- ISQETARG
  - target system summary panel 168
- ISQETRGG
  - guests summary panel 168
- ISQSTART 58
- ISQSTOP 58
- ISQTRACE command 170
- ISQVARS command
  - change status
    - target hardware 110
    - target system 109
    - target system attention 111
- ISQXDST 108
- ISQXDST command 98

- ISQXMON 98
- issuing commands 67

## L

- lock request 22
- locked votes and requests 22
- logical partition 6
- LookAt message retrieval tool xiv
- lost&found anchors 185
- LPAR management functions
  - CPC activation profiles 173
  - CPC and LPAR information
    - queries 173
  - enabling 174
  - enabling for BCP internal interface 174
  - enabling for processor operations 174
  - functional overview 173
  - introduced 173
  - LPAR management 173
  - usage considerations 175
- LPAR mode 6
- LU 6.2
  - routing commands 140

## M

- Main Menu 68
- major resources on the NMC 182
- management functions, LPAR 173
- mapping
  - NetView alerts to Tivoli events 209
  - SA z/OS messages and Tivoli events 203
- mapping statuses
  - automation agent to automation manager, 48
- marked and unmarked resources 197
- message classes 98
- message retrieval tool, LookAt xiv
- messages 203
  - interested operator list 97
  - monitoring 97
  - notification operators 98
  - waiting for IEA101A start message 110
  - waiting for IEA347A start message 110
  - waiting for VM start message 110
- minor resources on the NMC 182
- monitor information 72, 77
- monitor resources
  - using the NCCF panels 102
  - using the NMC 102
- monitoring
  - communications paths 112
  - hardware 100
  - health performance 102
  - messages 97
  - outstanding WTORs 97
  - processors 107
  - with SDF 114
- monitoring alerts 114
- move functions 23

- move groups and recovery mode 28
- MOVED 117
- MOVED automation status 43
- moving resources 90

## N

- names for processor operations
  - objects 194
- names for system operations objects 193
- nature of application groups
  - BASIC 89
  - MOVE 89
  - SERVER 89
- NCCF panels
  - using with monitor resources 102
- NetView
  - automation procedure 11
  - commands
    - LIST ASSIGN=GROUP 98
    - RMTCMD 140
    - ROUTE 140
  - logging on 53
  - RMTCMD routing 140
  - starting 57
  - terminating after recovery
    - actions 248
- NetView 3270 Management Console 213
- NetView alerts
  - mapping to Tivoli events 209
- NetView automation table 11
  - identifying 62
- NetView connection path 7, 112
  - displaying status 112
  - NVC 112
  - SNMP 112
- NetView Management Console 4, 67, 179
  - creating views 192
  - introduction 179
  - issuing commands 188
  - marked and unmarked resources 197
  - minor resources 197
  - monitoring 181
  - names for processor operations
    - objects 194
  - names for system operations
    - objects 193
  - problem management 196
  - processor operations 195
  - properties of objects 185
  - starting 180
- network views 192
- NMC 4, 179
  - creating views 192
  - introduction 179
  - issuing commands 188
  - marked and unmarked resources 197
  - minor resources 197
  - monitoring 181
  - multiple focal point support 180
  - names for system operations
    - objects 193, 194
  - problem management 196
  - processor operations 195
  - properties of objects 185
  - starting 180

- NMC (*continued*)
  - using with monitor resources 102
- NMC workstation 8
- non-disruptive restoration 27
- notification operators 98
  - adding 99
  - deleting 100
  - settings 99
- notify messages 99
- NVC (NetView connection) 7

## O

- objects 11
  - grouping 12
- observed status 35, 36
- observed statuses 40
- OCF (operations command facility) 6
- OCF-based processor 6
- OMEGAMON
  - INGSESS command 103
  - session management, operator interaction 103
  - sessions, displaying 104
  - sessions, displaying additional information 106
  - sessions, recovering 105
  - sessions, starting 104
  - sessions, stopping 105
- operation mode 94
- operator console 160
- order, defined 18
- OSA (open system adapter) 7
- outstanding WTORs
  - displaying 97
  - SDF colors 117
- override
  - resource 84
  - schedule 84
- override settings, displaying 88
- overrides 14
- overriding schedules 152
- overview of SA z/OS 3

## P

- PAM 18
- panels
  - INGTHRES 139
  - INGTHRES (Set Thresholds) 139
  - Path Detail 112
  - Target Hardware Summary 101
- parallel sysplex 6
- PARMLIB 32
- partitioning
  - logical 6
  - physical 7
- PATH detail panel ISQEPNVC 169
- pending policy changes 130
- physically partitioned configuration 7
- policy 8
- policy database 11
  - sharing 233
- port
  - ESCON port 7
  - FICON port 7

- preference scenarios 26
- preference values 24, 25
  - effective 24
- preferences
  - real hierarchy 27
- primary automation manager 18, 54
- priorities of subsystems 116
- priority of requests
  - FORCE 15
  - HIGH 15
  - LOW 15
- problem determination 233
- problem management on the NMC 196
- processor 6
  - monitoring status 107
  - OCF-based 6
  - target hardware summary 101
- processor control file 11
- processor operations 4
  - guest machines support 165
  - starting 58
  - stopping 58
- processor operations commands
  - ISQTRACE 170
- processor operations on the NMC 195
- processor operations status information 100
- processors 10
- ProcOps
  - First Failure Data Capture support 243
  - tracing and debugging 242
  - tracing specific connections 243
  - using AOCTRACE 243
- ProcOps dialog 165
- propagation
  - requests 19
  - requests, details 19
- propagation of conflicting requests 21
- properties of objects on NMC 185
- PSM console mode 169
- PSM control program
  - restarting 172
  - starting 170
  - stopping 171

## R

- rebuild 130
  - system-managed 127, 131
  - user-managed 127, 131
- recovering OMEGAMON sessions 105
- recovery actions
  - BCP internal interface 247
  - terminating NetView 248
  - terminating the BCP internal interface 248
- Recovery automation flag 33
- recovery mode and move groups 28
- refreshing automation policy 61
- refreshing the automation manager configuration file 62
- regular monitoring 50
- regular resources 181
- relationship
  - start 19
  - stop 19

- relationship information 73
- relationships concept 20
- reloading automation configuration file 61
- repair, takeover file 55
- request
  - defined 18
  - propagation 19
  - propagation, details 19
  - using INGVOTE 82
- requests
  - conflicting 20
  - creation of 14
  - hierarchy 15
  - locked 22
- requests and goals 19
- resource
  - depending 19
  - supporting 19
- resource aware application move 124
  - enabling 32
  - example 32
  - introduced 31
  - overview 31
- resource hierarchy with SDF 120
- resource information
  - sysplexwide 69
- resource override 84
- resource overrides 14
- resource subtype 71
- resources 11
  - start and stop 80
- Restart automation flag 34
- RESTART automation status 44
- restarting the PSM control program 172
- restoration
  - disruptive 27
  - non-disruptive 27
- Restructured Extended Executor (REXX) 11
- resynchronization 52
- REXX procedure 11
- RMTCMD command 140
- RODM (resource object data manager) 8
- ROUTE command 140
- routing
  - commands 140
- RUNNING automation status 44

## S

- SA z/OS
  - overview 3
  - starting 53
- SA z/OS components
  - I/O operations 4
  - processor operations 4
  - system operations 4
- SA z/OS event 202, 203
- SA z/OS main menu 68
- SA z/OS messages 203
  - mapping to Tivoli events 203
- SA z/OS topology manager
  - behavior 187
  - initialization 187
- SA z/OS topology manager overview 179

- SAM 18
- scenarios 147
- schedule 15, 84
- schedule override 84, 152
- schedule overrides 14
- SDF
  - monitoring application groups 121
  - monitoring with 114
  - monitoring WTORs 121
  - resource hierarchy 121
- SDF (status display facility) 12
  - customizing 115
  - how it works 118
- SDF command 115, 121, 122
- SDF hierarchy 122
- SDF, using 118
- second level systems
  - VM support 165
- secondary automation manager 18
- server management 23
- service period 84
- service period overrides 14
- session management
  - OMEGAMON, INGSSESS 103
  - OMEGAMON, operator interaction 103
- sessions
  - displaying OMEGAMON 104
  - OMEGAMON, displaying additional information 106
  - recovering OMEGAMON 105
  - starting OMEGAMON 104
  - stopping OMEGAMON 105
- SETSTATE 95
- SETSTATE command 95
- SETTIMER 94
- setting automation on or off 88
- setting up, work item lifecycle recording 239
- Shutdown automation flag
  - See* Terminate automation flag
- single point of control 141, 142
- single system image 141, 142
- SLIP TRAP 137
- specific connections, tracing 243
- SPFEDIT 233
- start resources 80
- startability status 35, 38
- STARTED automation status 44
- STARTED2 automation status 45
- starting
  - applications 58
  - automation manager 53
  - I/O operations 57
  - NetView 57
  - processor operations 58
  - PSM control program 170
  - SA z/OS 53
  - subsystems 81
  - system operations 57
- starting OMEGAMON sessions 104
- statistics, work item 238
- status
  - ABENDING 40
  - ACTIVE 41
  - alert 111
  - application monitor status 46
  - status (*continued*)
    - AUTODOWN 41
    - Automatic Restart Manager 47
    - automation agent status 40
    - AUTOTERM 41
    - BREAKING 41
    - BROKEN 42
    - clear to send 114
    - closed
      - target system 109
    - CTLDOWN 42
    - DCCF 111
    - degraded 110
    - determination 50
    - disabled wait 111
    - displaying with DISPSTAT 73
    - displaying with SDF 115
    - DOWN 42
    - ENDED 42
    - ENDING 42
    - environment alert 111
    - EXTSTART 43
    - FALLBACK 43
    - HALFDOWN 43
    - HALTED 43
    - initialized 109
    - IPL complete 109
    - IPL failed 109
    - last significant message 114
    - load failed 109
    - MOVED 43
    - NetView connection path 112
    - not active 109
    - not operating 109
    - observed status 40
    - path problem 114
    - power save 111
    - powered off 110
    - processor operations
      - attention status 111
      - path detail 112
      - path status 114
      - target system attention status 111
      - target system status 109
    - processors 107
    - RESTART 44
    - RUNNING 44
    - serious alert 111
    - service 111
    - service required 111
    - stage-1 activation complete 110
    - STARTED 44
    - STARTED2 45
    - STOPPED 45
    - STOPPING 45
    - STUCK 45
    - target hardware problem 111
    - undecidable 110
    - unknown 110
    - UP 46
    - waiting for IEA101A start message 110
    - waiting for IEA347A start message 110
    - waiting for VM start message 110
    - ZOMBIE 46
- Status Display Facility 114
- status forwarding path
  - failure 233
- status of a resource
  - automation status 35
  - compound status 35
  - desired status 35
  - observed status 35
  - startability status 35
- status of resources 35
- status panels 73
- status summary 107
- status summary panel ISQESUM 166
- statuses
  - mapping automation agent to observed 48
- statuses supplied by the automation agent
  - application monitor status 40
  - Automatic Restart Manager status 40
  - automation agent status 40
- stop resources 80
- STOPPED automation status 45
- stopping
  - applications 58
  - automation manager 54
  - I/O operations 57
  - processor operations 58
  - PSM control program 171
  - subsystems 82
- STOPPING automation status 45
- stopping OMEGAMON sessions 105
- structure
  - duplexing
    - system-managed 131
    - user-managed 131
  - pending policy changes 130
  - rebuild 130
    - system-managed 127, 131
    - user-managed 127, 131
- STUCK automation status 45
- subsystem colors and priorities 116
- subsystem priorities 116
- subsystems, stopping 82
- subtype 71
- switch 158
- switch matrix 226
- switches 7
- switching
  - alternate CDS to primary CDS 133
  - CFRM policy 133
- SYSDSN 233
- sysplex 6
  - monitor and control 122
- sysplex automation 122
- sysplex timer 8
- sysplexwide resource information 69
- system operations 4
  - starting 57
- system operations commands
  - INGMOVE 92
  - INGSTR 138
- system operations control files
  - automation control file 8
  - automation manager configuration file 8
- System Operations main menu 68
- system-managed duplexing 131

- system-managed rebuild 127, 131
- systems
  - grouping 12
  - monitoring status 108

## T

- TAF (Terminal Access Facility)
  - fullscreen sessions 143
  - panels 144
  - TAF fullscreen display 144
- takeover file repair 55
- takeover file, dynamic 56
- target hardware
  - status 110
- target hardware summary 101
- target hardware summary panel
  - ISQEHARP 166
- target processors 6
  - monitoring status 107
- target system
  - status 109
  - status panels
    - interested operator list 98
    - overall status 107
    - status summary 107
    - summary status 108
- target system summary 108
- target system summary panel
  - ISQETARG 168
- target systems
  - monitoring messages 97
  - monitoring status 108
- TEC 202
- TEC event server 202
- TEC Notification by SA z/OS 201
  - operation 211
  - starting 211
  - termination 214
- TEC rules 209
- terminal access facility fullscreen
  - sessions 143
- Terminate automation flag 34
- threshold
  - for z/OS components 139
- threshold settings 139
- thresholds 139
  - set or change 140
- timer
  - create 94
  - delete 94
  - modify 94
  - NetView 94
- TIMERS command 94
- Tivoli administrator 201
- Tivoli desktop 201
- Tivoli Enterprise Console 202
- Tivoli Enterprise Portal 8
  - using 103
- Tivoli events 202
- tokens 182, 185
- trace records, printing 236
- trace services for the automation manager 234
- tracing 234
  - automation manager 234
  - BCP Internal Interface 242

- tracing (*continued*)
  - I/O operations 243
  - ProcOps 242
- tracing specific connections 243
- trigger condition list 155
- trigger information 85
- tutorial, I/O operations ISPF dialog 232

## U

- UP automation status 46
- user IDs
  - for starting SA z/OS 53
- user-managed duplexing 131
- user-managed rebuild 127, 131
- using
  - NCCF panels with monitor resources 102
  - NMC with monitor resources 102
  - Tivoli Enterprise Portal 103

## V

- views
  - creating 192
  - exception views 192
  - network views 192
- VM
  - second level systems support 165
- vote
  - defined 18
  - using INGVOTE 82
- votes, locked 22

## W

- WLM-related data, INGPLEX command 124
- WLMQUERYINTERVAL 32
- work item lifecycle recording
  - setting up 239
  - using 239
- work item statistics, displaying 238
- working with application groups 88
- WTORS
  - displaying outstanding WTORS 97
  - monitoring with SDF 115, 121
  - SDF colors 117

## Z

- z/OS operator console 160
- ZOMBIE automation status 46

---

## Readers' Comments — We'd Like to Hear from You

System Automation for z/OS  
User's Guide  
Version 3 Release 2

Publication No. SC33-8263-05

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

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 state on this form.

Comments:

Thank you for your support.

Submit your comments using one of these channels:

- Send your comments to the address on the reverse side of this form.
- Send a fax to the following number: FAX (Germany): 07031 16-3456  
FAX (Other Countries): +49 7031 16-3456
- Send your comments via e-mail to: [s390id@de.ibm.com](mailto:s390id@de.ibm.com)

If you would like a response from IBM, please fill in the following information:

---

Name

---

Address

---

Company or Organization

---

Phone No.

---

E-mail address



Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES

# BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Deutschland Research & Development GmbH  
Department 3248  
Schoenaicher Strasse 220  
71032 Boeblingen  
Germany



Fold and Tape

Please do not staple

Fold and Tape







Program Number: 5698-SA3

Printed in USA

SC33-8263-05

