Planning and Installation

Version 3 Release 5
Note

Before using this information and the product it supports, read the information in Appendix I, "Notices," on page 263.

Edition Notes

This edition applies to IBM Tivoli System Automation for z/OS (Program Number 5698-SA3) Version 3 Release 5, an IBM licensed program, and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC34-2645-00.

IBM welcomes your comments. You may forward your comments electronically, or address your comments to:

IBM Deutschland Research & Development GmbH
Department 3282
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

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.

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

## Figures
- ix

## Tables
- xi

## Accessibility
- xiii
  - Using assistive technologies: xiii
  - Keyboard navigation of the user interface: xiii

## Dotted decimal syntax diagrams
- xv

## How to send your comments to IBM
- xvii

## About This Book
- xix
  - Who Should Use This Book: xix
  - Notes on Terminology: xix
  - Where to Find More Information: xx
    - Summary of Changes for SC34-2716-00: xx
    - New Information: xx
    - Changed Information: xx
    - Moved Information: xx
    - Deleted Information: xx

## Part 1. Planning

### Chapter 1. SA z/OS Prerequisites and Supported Equipment
- 3
  - SA z/OS Components: 3
  - Hardware Requirements: 3
  - SA z/OS Processor Operations: 3
  - SA z/OS System Operations: 3
  - Workstation Components: 4
  - Functional Prerequisites: 4
  - Software Requirements: 4
  - Mandatory Prerequisites: 4
  - Functional Prerequisites: 5
  - Supported Hardware: 5
    - Operator Terminals: 6
  - Supported Operating Systems: 6

### Chapter 2. What is New in SA z/OS 3.5
- 7
  - Configuration Assistant: 7
  - Role-based Security: 7
  - Integration with IBM Tivoli Monitoring Products: 7
  - Hardware Automation: 8
  - Application Pacing: 8
  - Customization Dialog Enhancements: 8
  - Job log monitoring for JES3: 9
  - Enterprise SDF status monitoring support: 9
  - Enhanced XCF communication: 9
  - IPL Complete Notification Support: 10
  - Relational Database Services (INGRDS) Enhancements: 10
  - GDPS Active-Active Support for Multiple Consistency Groups: 10
  - Tailorable 3270 Panels: 10
  - UP Status Delay: 10
  - Config Refresh Monitoring: 10
  - New exit AOFEXC25: 11
  - Serviceability Enhancements using NVINFO: 11

### Chapter 3. Planning to Install SA z/OS on Host Systems
- 13
  - Component Description: 13
  - System Operations: 13
  - Processor Operations: 13
  - I/O Operations: 14
  - SA z/OS and Sysplex Hardware: 14
    - Parallel Sysplex: 15
    - Coupling Facility: 15
    - Server Time Protocol (STP): 15
    - Logically Partitioned (LPAR) Mode: 16
    - The zEnterprise BladeCenter Extension (zBX): 16
    - Communications Links: 16
    - Control Units (CU): 17
    - I/O Devices: 17
    - NetView Management Console (NMC): 17
    - Tivoli Enterprise Portal Support: 17
    - Looping Address Space Suppression: 17
  - Planning the Hardware Interfaces: 18
    - Understanding the BCP Internal Interface: 18
    - Understanding the Processor Operations HTTP Interface: 18
    - Understanding the Processor Operations SNMP Interface: 19
    - Understanding supported SNMP Protocols: 19
    - Understanding the Hardware Console Automation Interface: 20
    - Understanding the TCP/IP Interface: 20
    - Deciding Which Hardware Interface to Use: 20
    - REXX Considerations: 20
      - Allocation Requirements for REXX Environments: 20
    - z/OS Considerations: 21
      - Prefixes: 21
      - Defining the XCF Group: 21
  - System Operations Considerations: 23
    - SA z/OS Hardware Interface: Important Considerations: 23
    - Automation Manager Considerations: 23
      - Storage Requirements: 24
      - OMVS Setup: 24
      - Recovery Concept for the Automation Manager: 25
      - Manager-Agent Communication and Status Backup: 25

© Copyright IBM Corp. 1996, 2014
Step 7A: Preparing the HMC (Console Workplace 2.9 and Later Versions) ........................................ 96
Step 7B: Preparing the SE (Console Workplace 2.9 and Later Versions) ........................................ 98
Step 7C: Preparing the SE (Console Workplace 2.10 and Later Versions) ...................................... 100
Step 7D: Updating Firewall Information .................. 101
Step 8: Preparing Ensemble HMC Communication 101
Step 8A: Setting up the Ensemble Hardware Management Console for use with System Automation for z/OS ........................................ 101
Step 8B: Setting up AT-TLS for the SSL socket connection ...... 101
Step 9: Preparing the VM PSM ................................ 103
Installing the PSM Code on VM .................................. 103
Configuration ........................................ 104
Customizing the PSM ..................................... 104
Step 10: Configure the Automation Manager ............... 106
Step 10A: XCF Characteristics ................................ 107
Step 10B: Configuring HSAPRmxx ................................ 107
Step 10C: ARM Instrumentation of the Automation Manager ........................................ 107
Step 10D: Security Considerations ...................... 108
Step 11: Configure the Component Trace ................ 108
Step 12: Configure the System Logger .................. 109
Step 13: Configure ISPF Dialog Panels .................. 111
Step 13A: Allocate Libraries for the Dialogs .............. 111
Step 13B: Logging Modifications to Data Set ........... 114
Step 13C: Invoking the ISPF Dialogs ................... 114
Step 13D: Reconvert I/O Operations Panels ............. 116
Step 13E: Verify the ISPF Dialog Installation .......... 116
Step 14: Verify the Number of available REXX Environments .................................................. 116
Step 15: Configure Function Packages for TSO .......... 117
Step 15A: Installation of the TSO REXX Function Package INGTXFPG ........................................ 117
Step 15B: Install SA Provided Authorized TSO Command INGPAUTH .................................... 118
Step 16: Configure Alert Notification for SA z/OS .......... 118
Enabling Alert Notification via SA IOM Peer-To-Peer Protocol ........................................ 119
Enabling Alert Notification via EIF Events. ................. 119
Enabling Alert Notification via XML .................... 120
Enabling Alert Notification via User-Defined Alert Handler ........................................ 121
Step 17: Compile SA z/OS REXX Procedures .............. 121
Step 18: Defining Automation Policy ...................... 121
Step 18A: Build the Control Files ...................... 122
Step 18B: Distribute System Operations Configuration Files .................................................. 122
Step 19: Define Host-to-Host Communications .......... 123
Step 19A: Configure VTAM Connectivity ................. 123
Step 19B: Perform VTAM Definitions for I/O Operations ........................................ 124
Step 19C: Perform TCP/IP Definitions for I/O Operations .......................................................................... 126
Step 20: Enabling SA z/OS to Restart Automatic Restart Manager Enabled Subsystems ............... 128
Step 21: Define Security ..................................... 128
Step 22: Configure the Status Display Facility (SDF) 129
Step 23: Check for Required IPL ...................... 130
Step 24: Automate System Operations Startup .......... 130
How to Automate the Automation Manager Startup ............................................................................ 131
Step 25: Verify Automatic System Operations Startup ............................................................................ 132
Step 26: Configure USS Automation ...................... 133
Step 26A: Securing USS Resources ...................... 133
Step 26B: Preparing for USS Automation .......... 133
Step 27: Enabling the System Automation WebServices Adapter ........................................ 133
Step 27A: Copy and Configure the Start Procedure ............................................................................ 134
Step 27B: Copy and Adapt the Configuration File ............................................................................ 134
Step 27C: Copy the Keystore File .......................... 135
Step 27D: Encrypt the Password of the Keystore File ........................................................................ 136
Step 27E: Update the Configuration File again .......... 137
Step 27F: Configure the WebServices Started Task ........................................................................... 137
Step 27G: Set Up Virtual IP Address for the WebServices Started Task ......................................... 137
Step 28: Install an SA z/OS Satellite ..................... 138
Step 28A: Configure the Networking NetView or Focal Point NetView Startup Procedure .................. 138
Step 28B: Configure the Networking NetView or Focal Point NetView DSIPARM Data Set .......... 138
Step 29: Installing and Customizing the NMC Focal Point .......................................................... 139
Step 29A: Preparing for NMC ................................ 139
Step 29B: Modify the NetView DSIPARM Data Set for the SA z/OS Topology Manager ................. 142
Step 29C: Configure RODM ................................ 145
Step 29D: Configure the INGTOPOF File ................. 146
Step 29E: Prepare BLDVIEWS Cards .................. 147
Step 30: Copy and Update Sample Exits ................ 147
Step 31: Install Relational Data Services (RDS) .......... 148
Step 32: Install CICS Automation in CICS ............. 148
Step 32A: SIT or Startup Overrides ...................... 148
Step 32B: Program List Table Definitions .............. 149
Step 32C: Define Consoles .................................. 149
Step 32D: Transaction and Program Definitions ....... 150
Step 32E: DFHRPL and the CICS Automation Library ........................................ 150
Step 32F: Add Libraries for NetView ................... 150
Step 32G: Installing CICSplex SM REXX API .......... 150
Step 33: Install IMS Automation in IMS ................. 151
Step 33A: Specify Required Control Region Parameters ........................................................................ 151
Step 33B: Install DFSAOE00 Exit ...................... 151
Step 33C: Add Libraries for NetView ................... 152
Step 34: Install TWS Automation in TWS .......... 152
Step 34A: Add Libraries to TWS ...................... 152
Step 34B: Add Libraries to NetView ................... 152
Step 34C: Update TWS Parameters and Exits ........ 152
Step 35: Configuring GDPS .............................. 154
Step 35A: Preparing NetView .......................... 154
Step 35B: Preparing the Automation Manager ........ 155
Step 35C: Defining the Automation Table Used by GDPS .................................................. 155
Step 36: Configuring I/O Operations .................... 156
### Chapter 11. Security and Authorization

- Authorization of the Started Procedures: 162
- Roles: 163
- Operators: 165
- Commands: 165
- Use of Commands Cross System: 166
- Use of Commands from TSO or Batch:
  - Front-end Checking: 168
  - Back-end Checking: 168
- Resources: 169
- Stylesheet Options: 171
- Other Security Options: 172
- Securing Focal Point Systems and Target Systems: 173
- Granting NetView and the STC-User Access to Data Sets: 174
  - Access to XCF Utilities: 174
  - Access to HOM Interface: 174
  - Access to IPL Information: 175
  - Access to Spare Couple Data Sets: 175
  - Access to User-Defined Couple Data Sets: 176
  - Access to Spare Local Page Data Sets: 176
  - Access to JES Spool Output Data Sets: 176
  - Access to the NetView UNIX Command Server: 176
  - Accessing authorized TSO command INGPAUTH: 177
- Restricting Access to INGPLEX and INGCF Functions: 177
- Restricting Access to Joblog Monitoring Task INGJLM: 178
- Security for IBM Tivoli Monitoring Products: 179
  - Controlling Access to IBM Tivoli Monitoring Products: 179
  - Controlling Access to OMEGAMON Monitors: 180
- Security for Ensemble HTTP Connections: 182
  - Adding SSL-Certificate to userid’s keyring: 182
  - Allowing NetView to Use the Ensemble Hardware commands: 183
- Levels of ensemble access: 183
- Password Management: 183
- Controlling Access to the Processor Hardware Functions: 184
  - Allowing NetView to Use the BCP Internal Interface: 185
  - Access to the CPCs: 185
  - Levels of CPC Access: 185
  - Defining the CPC Access Lists: 186
  - Implementing Granular Hardware Access: 186
  - Password Management for SNMPv3 HMC/SE Connections: 186
- Defining a RACF Profile for I/O Operations: 188
  - Assign RACF Authorization: 188
  - Assign Authorization by ACCESS Level: 188
  - Assign TCP/IP Port Authorization: 189
  - Access Authorization Levels: 189
  - Establishing Authorization with Network Security Program: 192
  - Security for WebServices Adapter: 193
  - Authentication of the WebServices Adapter: 193
  - Authorization of the WebServices Adapter: 194

### Chapter 12. Configuring SA z/OS Workstation Components

- Configuring the NMC Workstation: 197
  - Configuration Steps on the NMC Server: 199
  - Configuration Steps on the NMC Client: 201
  - Configuring IBM Tivoli Netcool/OMNibus: 204
  - Configuring the Triggers: 204
  - Configuring the Event View: 205
  - Configuring Tivoli Service Request Manager through Tivoli Directory Integrator: 205
  - Configuring the AssemblyLines: 206

### Part 3. Appendixes

- Appendix A. Planning for the NMC Environment: 211
  - NMC Exploitation Topology: 211
  - Planning to Install the NMC Workstation: 212
  - Running Multiple NetViews: 212
- Appendix B. Syntax for INGTOPOF File: 215
  - The SYSPLEX Statement: 215
  - The PROCOPS Statement: 216
  - The LOCATION Statement: 216
  - The ANCHOR Statement: 216
  - The BLDVIEWS Statement: 217
  - The OPTION Statement: 218
  - The TEMPLATE Statement: 219
  - Examples: 221
  - The RUNOPID Statement: 222
  - The HBDELETE Statement: 223
  - The LINKTOVIEWS Statement: 223
  - The MAPCOLOR Statement: 224
  - Sample INGTOPOF File: 224
- Appendix C. Miscellaneous Information: 227
  - Running Two NetViews on the NMC Focal Point System: 227
  - Users and RODM Authorization: 227
    - Verifying Installation of SA z/OS Satellite (Optional): 228
    - Enabling SA z/OS Support for Extended Multiple Console Support: 228
    - Setting Up EMCS: 228
    - EMCS Restrictions and Limitations: 228
# Appendix D. Using the Hardware Integrated Console of System z for External Automation with SA z/OS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI Usage in IBM Tivoli System Automation</td>
<td>231</td>
</tr>
<tr>
<td>Products</td>
<td>232</td>
</tr>
<tr>
<td>SA z/OS Processor Operations (ProcOps)</td>
<td>232</td>
</tr>
<tr>
<td>System Automation for Integrated Operations Management</td>
<td>232</td>
</tr>
<tr>
<td>Related Information</td>
<td>232</td>
</tr>
<tr>
<td>CI Protocols and Automation Interfaces</td>
<td>232</td>
</tr>
<tr>
<td>INTERNAL (BCPii Base Control Program)</td>
<td>232</td>
</tr>
<tr>
<td>Internal Interface</td>
<td>232</td>
</tr>
<tr>
<td>SNMP</td>
<td>233</td>
</tr>
<tr>
<td>System z Application Programming Interface</td>
<td>233</td>
</tr>
<tr>
<td>Related Information</td>
<td>233</td>
</tr>
<tr>
<td>CI Configuration for Remote Automation</td>
<td>233</td>
</tr>
<tr>
<td>CI Automation Basics</td>
<td>235</td>
</tr>
<tr>
<td>Related Information</td>
<td>236</td>
</tr>
<tr>
<td>CI Differences to 3270-Based Console Devices</td>
<td>236</td>
</tr>
<tr>
<td>CI Performance Factors</td>
<td>236</td>
</tr>
<tr>
<td>Network Dependencies</td>
<td>237</td>
</tr>
<tr>
<td>IP Stack Considerations</td>
<td>237</td>
</tr>
<tr>
<td>ProcOps SNMP Sessions</td>
<td>237</td>
</tr>
<tr>
<td>OS Message Format Support with ProcOps/BCPii</td>
<td>237</td>
</tr>
<tr>
<td>Automating Multi-Line z/OS Messages</td>
<td>238</td>
</tr>
<tr>
<td>Limiting the Number of z/OS IPL Messages</td>
<td>238</td>
</tr>
<tr>
<td>Displayed on CI</td>
<td>238</td>
</tr>
<tr>
<td>Recommended z/OS Console Settings for CI Usage</td>
<td>238</td>
</tr>
<tr>
<td>with SA z/OS</td>
<td>238</td>
</tr>
<tr>
<td>Using CI in a z/OS Sysplex Environment</td>
<td>239</td>
</tr>
<tr>
<td>Running with the z/OS System Console</td>
<td>239</td>
</tr>
<tr>
<td>Deactivated</td>
<td>239</td>
</tr>
<tr>
<td>z/OS Health Checker Considerations</td>
<td>239</td>
</tr>
<tr>
<td>CI Security with SA z/OS</td>
<td>240</td>
</tr>
<tr>
<td>Testing CI Performance for SNMP Connections</td>
<td>240</td>
</tr>
<tr>
<td>Summary: Managing CI Performance for SA z/OS</td>
<td>241</td>
</tr>
</tbody>
</table>

# Appendix E. Migration Information

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Migration Steps to SA z/OS 3.5</td>
<td>243</td>
</tr>
<tr>
<td>Migration Notes and Advice when Migrating to SA z/OS 3.5</td>
<td>243</td>
</tr>
<tr>
<td>Post SMP/E Steps</td>
<td>243</td>
</tr>
<tr>
<td>AT / MRT / MPF Migration Notes</td>
<td>244</td>
</tr>
<tr>
<td>File Update</td>
<td>244</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>245</td>
</tr>
<tr>
<td>Migration Notes and Advice when Migrating from SA z/OS 3.3</td>
<td>247</td>
</tr>
<tr>
<td>Coexistence of SA z/OS 3.5 with Previous Releases</td>
<td>248</td>
</tr>
</tbody>
</table>

# Appendix F. Ensemble Hardware Management Console Setup

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setting up the Hardware Management Console for use with System Automation for z/OS</td>
<td>251</td>
</tr>
<tr>
<td>Defining a user</td>
<td>251</td>
</tr>
<tr>
<td>Enable Web Services API</td>
<td>251</td>
</tr>
<tr>
<td>Getting the Hardware Management Console certificate</td>
<td>252</td>
</tr>
<tr>
<td>Firewall considerations</td>
<td>252</td>
</tr>
</tbody>
</table>

# Appendix G. Syntax for HSAPRM00

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
</table>

# Appendix H. INGDLG Command

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
</table>

# Appendix I. Notices

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming Interface Information</td>
<td>266</td>
</tr>
<tr>
<td>Trademarks</td>
<td>266</td>
</tr>
</tbody>
</table>

# Glossary

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
</table>

# Index

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
</table>
### Figures

1. Basic Hardware Configuration .......................... 14
2. Using SA z/OS Subplexes .............................. 22
3. Using Only the Takeover File for Status Backup ............ 26
4. Single Gateway Example .................................. 32
5. Example Gateways ........................................ 34
6. Alternate and Primary Focal Point System Connections from an IP Network to the Processor Hardware LAN ............. 38
7. Examples of Port Names in a Configuration ................. 45
8. Sample AT-TLS policy ..................................... 102
9. ISPF Application Selection Menu ......................... 115
10. VTAM Definition Statements ............................ 124
11. Sample of RODM Load Procedure EKGLOADP ............. 145
12. Startup JCL of I/O operations ............................ 157
13. Sample AT-TLS policy ................................... 159
14. Directory Structure of Unpacked Files ................... 199
15. Sample to Start the NMC (for WIN Environment) .......... 203
16. The SA z/OS Environment for NMC Support .............. 211
17. SA z/OS Enterprise with Networking Automation and System Automation running on the same NetView ............... 213
18. SA z/OS Enterprise Using a Networking NetView and an Automation NetView .......... 213
20. ISQ999I Test Message Pattern Example ................ 240
21. Coexistence of SA z/OS 3.5, SA z/OS 3.4, and SA z/OS 3.3 ........................................ 249
Tables

1. System Automation for z/OS Library .................................................. xix
2. Mandatory Prerequisites ......................................................... 4
3. Functional Prerequisites ......................................................... 5
4. Recovery Scenarios ................................................................. 26
5. Target Data Sets ................................................................. 51
6. USS Paths ............................................................................. 52
7. SA z/OS Host Configuration Tasks supported by the Configuration Assistant ......................................................... 55
8. Worksheet for job names ....................................................... 60
9. Configuration Tasks for SA z/OS Host Systems ................................................................. 75
10. Data Sets for Each Individual Automation Agent ......................................................... 77
11. Data Sets for I/O Operations .................................................... 78
12. Data Sets for Each Individual Automation Agent ......................................................... 78
13. Data Set for Each Sysplex ....................................................... 79
14. Data Sets for All Automation Managers in a Sysplex or Standalone System ......................................................... 79
15. Data Sets for Each Individual Automation Manager ......................................................... 80
16. Generation Data Groups for Each Individual Automation Manager ......................................................... 80
17. Data Sets for Each Individual Automation Agent ......................................................... 80
18. TSO Load Modules for INGTXFG ........................................... 117
19. Members to Start the Networking NetView ......................................................... 138
20. Use of AOFUPDAM and AOFUPDRODM to Control SA z/OS Data Storage ......................................................... 142
21. DSIPARM Members to be modified for the SA z/OS Topology Manager ......................................................... 142
22. SEQQMSG0 Data Set ............................................................... 153
23. Started Procedure Names for Functions ......................................................... 162
24. SAF-protected Resources for Functions ......................................................... 162
25. Security Roles ................................................................. 163
26. Option File variables for SAF-group name ......................................................... 164
27. Option File variables for UNIX System Services Group IDs ......................................................... 164
29. Information References for Security ......................................................... 172
30. Command Authorization Identifiers ......................................................... 181
31. I/O Operations Commands, their Availability and Access Authorization Levels ......................................................... 190
32. Access Authorization Levels Grouped by Function ......................................................... 191
33. RODM Authorization for user IDs ......................................................... 227
Accessibility

Publications for this product are offered in Adobe Portable Document Format (PDF) and should be compliant with accessibility standards. If you experience difficulties when using PDF files, please refer to the Knowledge Center at:

http://www-01.ibm.com/support/knowledgecenter/SSWRCJ_3.5.0/

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

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

Using assistive technologies

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

Keyboard navigation of the user interface

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

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

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

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

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

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

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

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

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

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

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

Notes:

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

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

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

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

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

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

Include the following information:
• Your name and address
• Your email address
• Your telephone or fax number
• The publication title and order number:
  IBM Tivoli System Automation for z/OS V3R50 Planning and Installation
  Guide
  SC34-2716-00
• The topic and page number related to your comment
• The text of your comment.

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

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

If you have a technical problem

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

This book describes IBM® Tivoli® System Automation for z/OS (SA z/OS) from a planning point of view, and how to install the product.

It also describes how to migrate to the latest release of SA z/OS.

Who Should Use This Book

This information is intended primarily for system programmers and automation programmers who plan for systems management and who install this product.

Notes on Terminology

MVS™:

References in this book to MVS refer either to the MVS/ESA product or to the MVS element of z/OS.

NetView:

The term NetView® used in this documentation stands for IBM Tivoli NetView for z/OS.

Where to Find More Information

The System Automation for z/OS Library

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

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

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

Related Product Information
For information that supports Tivoli System Automation for z/OS, visit the z/OS library in IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSLTBW/welcome).

Summary of Changes for SC34-2716-00
This document contains information previously presented in System Automation for z/OS V3.4.0 Planning and Installation, SC34-2645-00.

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

This document contains terminology, maintenance, and editorial changes.

New Information
The following information has been added:

- A description of the zEnterprise® hardware environment is added in “The zEnterprise BladeCenter® Extension (zBX)” on page 16 of Chapter 3, “Planning to Install SA z/OS on Host Systems,” on page 13.
- A new planning section for integrating SA z/OS with the IBM Tivoli Monitoring Services is added in Chapter 6, “Planning for Integration with IBM Tivoli Monitoring,” on page 41.
- A new ready-to-use automation solution works with OMEGAMON XE to query and detect looping address spaces as described in “Looping Address Space Suppression” on page 17.
- Use of a Configuration Assistant for the Installation and Configuration process is added and referenced in Part 2, “Installation and Configuration,” on page 49. The Configuration Assistant is already in use with the IBM Automation Control for z/OS product.
- “Step 2E: SA z/OS Password Store Data Set” on page 80 has been added.
- A new step for installing function packages to use authorized TSO commands under System Automation is added to “Step 15: Configure Function Packages for TSO” on page 117.
- SNMPv3 as a supported SNMP protocol is added. See “Understanding supported SNMP Protocols” on page 19, “Step 7A: Preparing the HMC (Console Workplace 2.9 and Later Versions)” on page 96 and “Step 7B: Preparing the SE (Console Workplace 2.9 and Later Versions)” on page 98. Also described is
Changed Information

The following information has been changed:

- “Step 4B: Update SCHEDxx” on page 82 is amended for z/OS 2.1.
- The GETPW command has been replaced by the SA z/OS command INGPW in Chapter 11, “Security and Authorization,” on page 161, “Controlling Access to OMEGAMON Monitors” on page 180, and “Controlling Access to the Processor Hardware Functions” on page 184.
- Former Step 1 SMP/E installation now becomes part of a new installation chapter in Part 2, “Installation and Configuration,” on page 49.
- “Step 6F: Add the REXX Function Packages to DSIRXPRM” on page 95 is changed to include NetView REXX function packages.
- “Step 19: Define Host-to-Host Communications” on page 123 is adjusted for VTAM connectivity and the required range of definitions for I/O Operations.
- The former Step 33A: Define UNIX Segments (OMVS) is retitled to “Step 26A: Securing USS Resources” on page 133.

Moved Information

The information in the section “Supported Software” has been moved to the table in “Functional Prerequisites” on page 5.

The former Step 33: Install USS Automation is moved to “Step 26: Configure USS Automation” on page 133.

The “Adding SSL-Certificate to userid’s keyring” on page 182, “Allowing NetView to Use the Ensemble Hardware commands” on page 183, “Levels of ensemble access” on page 183, and “Password Management” on page 183 sections are moved to a new section “Security for Ensemble HTTP Connections” on page 182.

Deleted Information

The appendix “CICS Automation Migration from SA z/OS” 3.2 is removed.

The section “Installing and Customizing the TEC Event Server Workstation” is removed.
Part 1. Planning

This part provides details on the following:

- Chapter 1, “SA z/OS Prerequisites and Supported Equipment,” on page 3
- Chapter 2, “What is New in SA z/OS 3.5,” on page 7
- Chapter 3, “Planning to Install SA z/OS on Host Systems,” on page 13
- Chapter 4, “Planning to Install Alert Notification by SA z/OS,” on page 27
- Chapter 5, “Planning for Automation Connectivity,” on page 31
- Chapter 7, “Naming Conventions,” on page 43
Chapter 1. SA z/OS Prerequisites and Supported Equipment

SA z/OS Components

SA z/OS consists of the following three components:

- System operations (SysOps for short)
- Processor operations (ProcOps for short)
- I/O operations (I/O-Ops for short)

I/O-Ops manages ESCON® and FICON® Directors (Switch Directors for short).

Refer to "Component Description" on page 13 for details.

SA z/OS also provides special automation facilities for the following products:

- CICS®
- DB2®
- IMS™
- TWS

Hardware Requirements

IBM has tested SA z/OS on IBM processors. SA z/OS uses the S/390® interfaces that vendors of other processors capable of running z/OS have stated that they support. Check with your vendor for details.

The target system can run in any hardware environment that supports the required software.

SA z/OS Processor Operations

The processor operations base program can run on any processor supported by Tivoli NetView for z/OS, Version 6 Release 1.

SA z/OS System Operations

The system operations base program can run on any processor supported by Tivoli NetView for z/OS, V6.1 (including OA44787) or Tivoli NetView for z/OS, V6.2.0 (including OA44787 and OA44789) or Tivoli NetView for z/OS V6.2.1 and z/OS Version 1 Release 12.

SA z/OS I/O Operations

The I/O operations base program can run on any processor supported by z/OS V1.12.
Hardware Requirements

Workstation Components

The NMC exploitation used by SA z/OS can run on all NMC topology server and NMC topology client hardware that is supported by Tivoli NetView for z/OS, V6.1.

Functional Prerequisites

The processor hardware interfaces of SA z/OS support the following processor hardware family:

- IBM System z

Following are the minimum required workplace versions of the Support Element and Hardware Management Consoles for the processor hardware interfaces of SA z/OS:

- SE: Workplace Version 2.9.2
- HMC: Workplace Version 2.9.2

Software Requirements

This section describes the environment of the target system required to install and use SA z/OS.

Notes:

1. To properly invoke the Japanese language version of SA z/OS, a Japanese language version of NetView must be installed and the Kanji support must be enabled. For Kanji workstation support a Japanese language host must be connected to a Japanese language workstation. If an English language workstation is connected to a Japanese language host some messages may be unreadable.

2. Check with IBM Service for required product service levels in addition to the base product releases. Certain service levels may be required for particular product functions.

3. SA z/OS processor operations is enabled on a focal-point system, from which it monitors and controls SA z/OS processor operations target systems. The SA z/OS processor operations target system may also have SA z/OS installed for its system operations and I/O operations but the processor operations will not be enabled. This section does not describe the SA z/OS Processor Operations target system.

Unless otherwise noted, subsequent versions or releases of products can be substituted.

Mandatory Prerequisites

A mandatory prerequisite is defined as a product that is required without exception; this product either will not install or will not function unless this requirement is met. This includes products that are specified as REQs or PREs.

Table 2. Mandatory Prerequisites

<table>
<thead>
<tr>
<th>Product Name and Minimum VRM/Service Level</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>z/OS V1.12 or later.</td>
<td></td>
</tr>
<tr>
<td>Tivoli NetView for z/OS, V6.1</td>
<td></td>
</tr>
</tbody>
</table>
Software Requirements

Functional Prerequisites

A functional prerequisite is defined as a product that is *not* required for the successful installation of this product or for the basic function of the product, but *is* needed at run time for a specific function of this product to work. This includes products that are specified as IF REQs.

Table 3. Functional Prerequisites

<table>
<thead>
<tr>
<th>Product Name and Minimum VRM/Service Level</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>z/OS base elements or optional features:</strong></td>
<td></td>
</tr>
<tr>
<td>z/OS SecureWay Security Server (including RACF® and DCE Security Server components)</td>
<td>For sysplex-based authorization and RACF-based NetView authorization</td>
</tr>
<tr>
<td><strong>Other program products:</strong></td>
<td></td>
</tr>
<tr>
<td>HTML browser</td>
<td>For customization reports</td>
</tr>
<tr>
<td>z/VM® 5.4, or later</td>
<td>For VM Second Level Systems support</td>
</tr>
<tr>
<td>PTF UA31443 on z/OS V1.7 or z/OS V1.8, or later with z/OS XML System Services</td>
<td>OMEGAMON® XE Support</td>
</tr>
</tbody>
</table>
| IBM Tivoli OMEGAMON II for MVS V5.2, or later | For the following commands:  
  • INGMTRAP  
  • INGOMX |
| IBM Tivoli OMEGAMON II for CICS V5.2, or later | |
| IBM Tivoli OMEGAMON II for IMS V5.1, or later | |
| IBM Tivoli OMEGAMON II for DB2 V5.4, or later | |
| Tivoli OMEGAMON XE for MVS on z/OS v4.2 | |
| Tivoli OMEGAMON XE for CICS on z/OS 4.2 | |
| Tivoli OMEGAMON for IMS on z/OS 4.2 | |
| Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS V4.2 | |
| IBM Tivoli Monitoring Services (ITMS, 5698-A79) V6.2.3 | SA z/OS Monitoring Agent for Tivoli Enterprise Portal support (FMID HKAH35T), see also IBM Tivoli System Automation for z/OS Monitoring Agent Configuration and User’s Guide |
| IBM CICS Transaction Server for z/OS V3.1, or later | For integrated automation of CICS address spaces and CICSPlex®-based monitoring |
| IBM DB2 for z/OS V8.1, or later | For integrated automation of DB2 address spaces |
| IBM IMS V11.1, or later | For integrated automation of IMS address spaces |
| IBM TWS for z/OS V8.2, or later | For integrated automation of TWS address spaces |

**Workstation Prerequisites:**

<table>
<thead>
<tr>
<th>Product Name and Minimum VRM/Service Level</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM Tivoli Business Service Manager for z/OS V4.2</td>
<td>For event notification</td>
</tr>
<tr>
<td>IBM Tivoli Netcool/OMNibus V7.2, or later</td>
<td>For event notification</td>
</tr>
</tbody>
</table>

Supported Hardware

SA z/OS processor operations supports monitoring and control functions for the processors of the following IBM mainframe family, including the logical partitioning of these processors:

- System z

SA z/OS processor operations supports workload optimizing attachments:

- IBM zEnterprise BladeCenter Extension (zBX) Model 003 and Model 002
Supported Hardware

SA z/OS provides a wide range of I/O configuration information and control functions for various types of hardware other than processors, though it does not require any of them. The hardware can include channels, control units and devices (both ESCON and non-ESCON), ESCON Directors (they are not required), and hardware used for sysplex coordination such as coupling facilities and External Time Reference (ETR) devices.

Operator Terminals

SA z/OS supports any display supported by ISPF V6.1 or higher. This is required for access to SA z/OS I/O operations functions and the SA z/OS customization dialogs.

The SA z/OS customization dialogs must be used with a terminal type of 3278.

Supported Operating Systems

SA z/OS processor operations monitors and controls target systems with the following operating systems:
- z/OS
- z/VM
- z/VSE
- Linux on System z
Chapter 2. What is New in SA z/OS 3.5

This information contains an overview of the major changes to SA z/OS for Version 3 Release 5. Use this information to check the impact on your user-written programming interfaces, such as automation procedures.

You should also refer to Appendix E, “Migration Information,” on page 243 for details of how to migrate to SA z/OS 3.5.

Configuration Assistant

The configuration assistant allows you to quickly configure System Automation for z/OS in a single-system and multi-system environment by using lab-proven best practices configuration settings for NetView and System Automation for z/OS.

Using a single options file, the configuration assistant generates all artifacts required to run the product, from data set allocation, to generating SYS1.PARMLIB and SYS1.PROCLIB members, populating the NetView DSIPARM up to security definitions for the z/OS Security Server to protect your automation environment. Thereby it avoids manual tasks required in the past to customize install and configure jobs and avoids duplicate user input. A verification utility furthermore can validate all prerequisites are fulfilled after the configuration is completed.

Role-based Security

System Automation for z/OS simplifies and improves the already existing security of the automation environment by consequently using roles with associated command and resource authorization levels. Users must be authorized to issue a command and to update a resource, before that resource’s status can be changed by that command.

With System Automation for z/OS also, third-party security checking is introduced for those users who access the product from TSO, from a batch job or by using OSLC web services. To simplify the definition effort, in combination with the configuration assistant, most of the z/OS Security Server profiles and resource definition commands can be generated automatically and saved in a data set member.

Integration with IBM Tivoli Monitoring Products

System Automation for z/OS improves the integration with IBM Tivoli Monitoring in multiple ways:

- The integration between System Automation for z/OS and IBM Tivoli OMEGAMON is further intensified by providing an out-of-the-box scenario for suppressing looping address spaces. Leveraging OMEGAMON for z/OS, looping address spaces are identified and policy is used to decide what course of action (for example, STOP, RESET, DIAG, WARN, ...) should be taken and under what circumstances. This support helps to increase the availability of your system, increases the productive usage of CPU capacity and hence improves the overall service level.
What is New in SA z/OS 3.5

- Captured messages can now be reported immediately as situations on the Tivoli Enterprise Portal (TEP) for selected automation resources using SA z/OS monitoring agent.
- The SA z/OS monitoring agent now supports the IBM Tivoli Monitoring self-describing-agent feature that allows to automatically install all support files on the monitoring server, the portal server and the portal itself and hence contributes to lower cost of ownership and less downtime of your monitoring components.
- Several attribute groups collected by the SA z/OS monitoring agent have been enhanced to correspond with the latest level of System Automation 3270-based commands. This includes information about run modes and pacing gates.

Hardware Automation

The following enhancements are provided:

- System Automation for z/OS tolerates the latest mainframe hardware generation, the IBM zEnterprise EC12. Furthermore, it allows the administrator to designate a target system as a zAware logical partition in the automation policy and allows the operator to activate and deactivate that logical partition like any other logical partition on the processor.
- System Automation for z/OS allows you to use SNMPv3 as the protocol between the Processor Operations focal point and the Hardware Management Console or the Support Element. With SNMPv3 the data from and to the hardware is sent in encrypted format.
- The LPAR scope feature allows you to improve the performance for initializing a particular processor. Rather than retrieving specific information for all logical partitions of a processor, only those logical partitions are processed which are defined as target systems for the processor in the automation policy. This can reduce the overall initialization time considerably.
- System Automation for z/OS can recognize priority messages, for example synchronous WTORs issued from the operating system and can also reply to such messages with the priority indication set.
- zEnterprise BladeCenter Extension support has been enhanced to allow switching the Ensemble Hardware Management Console from the primary to the alternate.

Application Pacing

Certain high CPU intensive workloads can dominate the system during start or stop times. System Automation for z/OS introduces therefore pacing gates that allow you to throttle the number of such workloads that can be started or stopped concurrently.

Customization Dialog Enhancements

The following Customization Dialog enhancements are provided:

- Concurrent edit and browse - The Customization Dialog has been enhanced to allow access to the policy in browse mode, either in the scope of the complete policy, in the scope of a particular entry type, or for one particular entry only. Browsing and editing concurrently by multiple users or in split screen is now possible as well as concurrent browse and policy built.
The command processing for NetView restart, configuration refresh, or time processing has been simplified by introducing the new start up phases REFRESHSTART and ANYSTART. This avoids in particular the duplicate definitions of commands that must be issued whenever NetView is recycled and after the application became available.

Any command can be entered in mixed case and the length of the command entry field in the Customization Dialog has been increased to allow administrators to enter longer commands directly into the policy and thus avoid writing scripts.

Further enhanced easy message management capabilities provided enhanced automation table syntax checking and allow you to control in a more granular way what is being generated for MPF, the automation table and for the message revision table.

All changes of the automation policy are now completely logged by the Customization Dialog to provide a complete audit trail and change reference.

More and enhanced best practices policies provide a less complex z/OS base configuration, include support for the Tivoli OMEGAMON manager, DB2's administrative task scheduler, IMS Connect, SAP server and JES2 network server.

Enables user-defined categories and subcategories for application definitions.

Supports flat file export and import for processors, in particular for the PROCESSOR INFO policy and also to export policies in batch.

---

**Job log monitoring for JES3**

Messages that are written to the JES job log or into another data set in the JES SPOOL are now also available for automation, regardless of what Job Entry Subsystem, JES2 or JES3, is being used.

---

**Enterprise SDF status monitoring support**

The Status Display Facility (SDF) supports now identical component names at the lowest tree level as long as this level does not immediately follow the root level to enable enterprise-wide comparisons, for example status per system or status per sysplex, on the SDF panel.

In addition, the display capabilities of SDF have been enhanced such that title line of a panel body can be placed anywhere in the panel. Finally, the sort order of the rows within a body can be changed from status priority to alphanumeric content in a cell both, in ascending and descending order.

---

**Enhanced XCF communication**

System Automation for z/OS has been enhanced to enable XCF communication within the physical sysplex, beyond the scope of the SAplex.

Commands and their responses targeted to a system outside of the current SAplex but within the physical sysplex can now be sent using XCF and do not require NetView RMTCMD.
IPL Complete Notification Support

To let installations monitor the progress of an IPL, SA z/OS allows the administrator to set a time limit and to mark those resources representative for a complete IPL.

When the system is started initially, the observed status of these resources is monitored and SA z/OS reports the status of the IPL, when all resources became available before the time limit. It would also report the status at the time limit, when some resources are still not available. And optionally, SA z/OS issues a final completion message, even after the time limit has expired, based on a new automation option.

Relational Database Services (INGRDS) Enhancements

The Relational Database Services (RDS) have been improved for better performance for insert, update and archiving of tables.

In addition, new functions have been introduced to enable mass inserts or mass updates and similar capabilities for import and export. RDS can now also be used with volatile RDS tables, if no persistent data repository is available.

GDPS Active-Active Support for Multiple Consistency Groups

System Automation for z/OS provides a new subtower GDPSSAT for the GDPS Satellite component that is targeted for GDPS Active-Active environments in support of graceful switch and multiple consistency group support.

Tailorable 3270 Panels

All tabular panels with horizontal scrolling capabilities, for example INGLIST, can now be tailored by the operators. It is possible to denote fixed and floating columns, to hide columns, to determine the order of the columns and to specify the sort order for one or multiple columns.

This together with colored status fields in INGAMS and DISPMTTR help to increase the usability of the panels.

UP Status Delay

Some applications report themselves as being fully available even though some startup activities are still going on in the background.

In order to avoid that the UP status triggers further Automation Manager activities prematurely due to relationships, System Automation for z/OS introduces a new field for applications that can be used to delay the true UP status by a user-defined time.

Config Refresh Monitoring

System Automation for z/OS has been enhanced to monitor the progress of a configuration refresh using INGAMS.

It leverages the new SDF capability that allows you to compare the same status descriptor for several systems. In the product provided sample panel, SDF
visualizes when a configuration refresh is in progress, when an error happened, or when the refresh process has successfully finished.

The sample SDF panel INGPTOP has been changed. It now includes the new status descriptor INGCFG.

**New exit AOFEXC25**

This exit allows installations to be informed about new, changed or deleted APL, APG, and MTR resources detected during a configuration refresh using INGAMS.

Administrators can use the exit to have a better understanding of the changes and may prevent the immediate activation of the changes. This is especially helpful when the automation policy can be updated anytime but configuration changes are done not very often.

**Serviceability Enhancements using NVINFO**

NetView for z/OS has introduced the command NVINFO that prints important serviceability information into the netlog and optionally produces a dump of the NetView address space.

System Automation for z/OS provides a plug-in invoked by NVINFO when the SA tower is set to include also important serviceability information from a System Automation point of view.
Chapter 3. Planning to Install SA z/OS on Host Systems

Component Description

The SA z/OS product consists of the following components:

- System operations (SysOps for short)
- Processor operations (ProcOps for short)
- I/O operations (I/O-Ops for short)

I/O-Ops manages ESCON and FICON Directors (Switch Directors for short).

System Operations

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

Enterprise monitoring is used by SA z/OS to update the NetView Management Console (NMC) resource status information which is stored in the Resource Object Data Manager (RODM).

Processor Operations

Processor operations monitors and controls processor hardware, zEnterprise BladeCenter Extensions hardware (zBX), and VM guest systems operations. It provides a connection from a focal point system to a target processor Support Element or a Hardware Management Console. With NetView on the focal point system, processor operations automates operator and system consoles for monitoring and recovering target processors and blade centers.
Component Description

Processor operations performs or automates many operator tasks, usually done using the HMC, such as activate / deactivate a logical partition, power on and off, and reset of multiple target processors. You can initiate IPLs and respond to system startup operator prompt messages, monitor status, and detect and resolve wait states. With ensemble processor operations commands you can discover, monitor and manage zBX resources, such as blades, virtual servers and workloads.

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.

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

SA z/OS and Sysplex Hardware

When SA z/OS is used in a Parallel Sysplex® environment, the hardware setup can be similar to the one illustrated in Figure 1.

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.
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.

Operators can also use SA z/OS Tivoli Enterprise Portal (TEP) support to monitor the status of automation on z/OS systems and z/OS sysplexes from a workstation that has a TEP client installed on it.

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 like 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 to perform hardware commands like LOAD or SYSTEM RESET to control the hardware and hardware images. SA z/OS processor operations can be configured to use TCP-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.

The following sections describe some relevant resources that are used by SA z/OS and its components.

**Parallel Sysplex**
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 functions are all part of a Parallel Sysplex. You can control a Parallel Sysplex by NetView-based commands or through an NMC workstation.

**Coupling Facility**
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.

**Server Time Protocol (STP)**
Server Time Protocol (STP) is a server-wide facility that is implemented in the Licensed Internal Code (LIC) of the IBM System z family processors. It provides time synchronization in sysplex or non-sysplex configuration.
Logically Partitioned (LPAR) Mode

A processor with the Processor Resource/Systems Manager™ (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 *logical* because the processor is not physically divided, but divided only by definition.) The partitions are defined, monitored, and activated separately by processor operations.

The zEnterprise™ BladeCenter® Extension (zBX)

An infrastructure component of the zEnterprise that houses and supports selected IBM blade servers and workload optimizers. zBX is the new infrastructure for extending System z® qualities of service and management capabilities across a set of integrated, fit-for-purpose POWER7® and IBM® System x® compute elements in the zEnterprise System. For more information refer to:

http://www-03.ibm.com/systems/z/hardware/zenterprise/zbx.html

Communications Links

Links that connect the focal point processor to target processors so that commands, messages, and alerts can flow. For more information refer to "Defining System Operations Connectivity" on page 31.

SNMP

SNMP may be chosen as the protocol for communications between the processor operations focal point and the SE or HMC.

See also "Understanding the Processor Operations SNMP Interface" on page 19.

BCP Internal Interface

For processor hardware automation in a sysplex environment, this link allows a z/OS system directly to communicate with its own hardware SE, as well as the SEs of other hardware which are part of a cluster of processors. This cluster must be defined to the Master HMC in a processor environment. If a sysplex processor hardware is to be automated, the processor hardware of all sysplex members must be defined to the Master HMC.

See also "Understanding the BCP Internal Interface" on page 18.

NetView RMTCMD Function

A connection that allows communication between the target and focal point system in order to pass status changes to the focal point system. This communication method is also used for other purposes.

TCP/IP

For VM second level system automation, this link allows SA z/OS ProcOps to communicate with the ProcOps Service Machine (PSM) on the VM host of the second level systems.

See also "Understanding the TCP/IP Interface" on page 20.

HTTP

HTTP may be chosen as the protocol for communications between the processor operations focal point and the ensemble HMC of a zEnterprise Ensemble supporting the zBX BladeCenters.
Control Units (CU)
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.

I/O Devices
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.

NetView Management Console (NMC)
A NetView function that consists of a graphic series of windows controlled by the NetView program and that allows you to monitor the SA z/OS enterprise interactively. The NetView Management Console consists of an NMC server and an NMC client.

The NMC client is connected to the NMC server that communicates with NetView. The NetView Management Console (NMC) can be implemented with an optional client, either on the server or separately.

Tivoli Enterprise Portal Support
SA z/OS Tivoli Enterprise Portal (TEP) support allows you to monitor the status of automation on z/OS systems and z/OS sysplexes using a TEP client. The client is the user interface for an SA z/OS monitoring agent. The monitoring agent uses Tivoli Monitoring Services infrastructure, which provides security, data transfer and storage, notification mechanisms, user interface presentation, and communication services for products in the IBM Tivoli Monitoring and OMEGAMON XE suites in an agent-server-client architecture.

The monitoring agent is installed on the systems or subsystems in the sysplex that you want to monitor and passes data to a hub Tivoli Enterprise Monitoring Server (monitoring server), which can be installed on z/OS, Windows, and some UNIX operating systems. The monitoring server communicates with the Tivoli Enterprise Portal Server (portal server), which then communicates with the portal client.

For more details, see IBM Tivoli System Automation for z/OS Monitoring Agent Configuration and User’s Guide.

Looping Address Space Suppression
This is a new automation solution ready for immediate use that queries OMEGAMON XE (through its SOAP Interface) to detect address spaces that are in long running, CPU demanding execution patterns.

Such address spaces are probably caught in CPU intensive loops and are thus undesirable.
Once the procedure identifies such an address space, it will consult automation policy to categorize it and then it will apply the pass based recovery mechanism that is specified for the category. The recovery may be passive, diagnostic, active or an escalating mixture of two or three of those elements.

For further information, refer to “Planning for Looping Address Space Suppression” on page 41.

Planning the Hardware Interfaces

This section provides additional information about the processor hardware interfaces supported by SA z/OS.

Understanding the BCP Internal Interface

In order to allow the sysplexwide activation or deactivation of the coupling facilities and to control sysplex members leaving the sysplex, SA z/OS uses the BCP (Basic Control Program) internal interface. The BCP internal interface of the following processor hardware family is supported:

• System z

Using the BCP internal interface from MVS allows you to send hardware operations commands such as SYSTEM RESET, or ACTIVATE to the Support Element attached to its own processor hardware (CPC). If the CPC is configured in LPAR mode, the operations command can be sent to all logical partitions defined on the CPC.

Furthermore, with the enhanced sysplex functions of SA z/OS, sysplex members running on other CPCs than their own image can be controlled through the BCP internal interface. This is possible by defining all CPCs of your sysplex on the master HMC of your processor hardware LAN.

At the processor hardware LAN level, the BCP internal interface uses the SNMP transport protocol. For this reason, the Support Elements need to be configured for SNMP. One HMC in the processor LAN must be configured to be the Change Management Master HMC, otherwise routing between the own SE and other SEs will not work.

Note that the MVS/HCD function uses the BCP internal interface to update IOCDS and IPL information in the Support Elements of addressed CPCs. You cannot use SA z/OS to perform these tasks, nor can HCD be used to perform the hardware operations functions of SA z/OS.

Currently, the BCP internal interface cannot be used by the processor operations focal point application. The interface can be configured and used for Parallel Sysplex automation purposes only. Exceptions to this are the processor operations common commands for LPAR management, see the chapter “Common Commands” in IBM Tivoli System Automation for z/OS Operator’s Commands for details.

Understanding the Processor Operations HTTP Interface

Using the HTTP interface of the processor operations, you can monitor and control ensemble zBX hardware from a processor operations focal point NetView in an IP network environment. With the processor operations HTTP interface, the following ensemble objects can be discovered and managed:
Planning the Hardware Interfaces

- zBX Blade Centers
- zBX Blades
- Virtualization hosts ("power-vm" and "x-hyp")
- Virtual servers ("power-vm" and "x-hyp")
- Workloads

As an extension to the BCP internal interface and SNMP, its purpose is to support the management commands (for example, ACTIVATE, DEACTIVATE) provided by the HMC Web Services API.

The Ensemble Hardware Management Console (HMC) of the ensemble you want to control must be configured for the Web Services API. Because this interface uses the SSL over IP network for communication between the processor operations focal point and the HMCs, the TCP/IP UNIX System Services stack with a running PAGENT and configured Application Transparent TLS (AT-TLS) are required to be active on the processor operations focal point system.

Understanding the Processor Operations SNMP Interface

Using the SNMP interface of processor operations, you can monitor and control local or remote processor hardware from a processor operations focal point NetView in an IP network environment. This is different to the BCP internal interface, which allows mutual hardware control among sysplex members without a system network dependency.

With the processor operations SNMP interface, the following processors can be managed:
- System z

As with the BCP internal interface, its purpose is to support the operations management commands (for example, ACTIVATE, SYSRESET) provided by the processor hardware.

The Support Elements of the CPCs you want to control must be configured for SNMP. Alternatively, you can configure a single HMC instead of multiple Support Elements in your processor LAN environment for SNMP. On this HMC the CPCs you want to control must be defined. Multiple HMCs, SEs, or both can be defined in your SA z/OS configuration.

Because this interface uses the IP network for communication between the processor operations focal point and the SEs or HMCs, the TCP/IP UNIX System Services stack is required to be active on the processor operations focal point system.

Understanding supported SNMP Protocols

Simple Network Management Protocols (SNMP) is an "Internet-standard protocol for managing devices on IP networks". There are several versions of the protocol that are available. System Automation supports SNMPv2c and SNMPv3 protocols.

SNMPv2c is Community-based Simple Network Management Protocol version 2 and is defined in RFC 1901 - RFC 1908. Authentication is done based on a Community Name.
SNMPv3 makes no change to the protocol aside from the addition of cryptographic security and user and password authentication and provides more security compared to the SNMPv2c:

- Confidentiality - Encryption of packets to prevent snooping by an unauthorized source.
- Integrity - Message integrity to ensure that a packet has not been tampered with in transit including an optional packet replay protection mechanism.
- Authentication - to verify that the message is from a valid source.

Understanding the Hardware Console Automation Interface

The System z mainframes provide a console facility that the SA z/OS hardware interfaces use to perform remotely either manual or automated operating system initialization and recovery. See Appendix D, “Using the Hardware Integrated Console of System z for External Automation with SA z/OS,” on page 231 for console definition, usage, performance, network, and basic information.

Understanding the TCP/IP Interface

Using the TCP/IP interface of Processor Operations, you can monitor and control VM guest systems from a Processor Operations focal point NetView in an IP network environment.

Processor Operations communicates with the ProcOps Service machine (PSM) using TCP/IP. The PSM can be regarded as an HMC or SE substitute for the virtual machines. The PSM itself uses the VM/CP Secondary Console Interface (SCIF) facility to communicate with the single VM second level systems.

The TCP/IP UNIX System Services stack is required to be active on the Processor Operations focal point system.

Deciding Which Hardware Interface to Use

If you want to use the Parallel Sysplex enhancements of SA z/OS and you have configured your customization to use IXC102A message automation, the BCP internal interface is required.

Note, that this interface can coexist with the supported SNMP interface on a processor operations focal point system. Because the IXC102A automation, which is part of the Parallel Sysplex XCF automation, can also be performed in SA z/OS using proxy resources together with processor operations, a decision must be made, which automation to use. It is recommended to use the XCF automation based on the BCP internal interface and to disable the IXC102A proxy resource automation based on processor operations.

REXX Considerations

Allocation Requirements for REXX Environments

Before running SA z/OS you may need to change the maximum number of REXX environments allowable.

The number of REXX environments allowable is defined in the REXX environment table. See z/OS TSO/E Customization for more information. TSO/E provides a SYS1.SAMPLIB member called IRXTSMPE, which is an SMP/E user modification
to change the maximum number of language processor environments in an address space. Define the number of allowable REXX environments on the IRXANCHR macro invocation:

IRXANCHR ENTRYNUM=xxx

For more details, see “Step 14: Verify the Number of available REXX Environments” on page 116.

Install the user modification by following the instructions in z/OS TSO/E Customization.

### z/OS Considerations

#### Prefixes

You should make sure you do not have any load modules, REXX parts or members with the following prefixes:

- AOF
- EVE
- EVI
- EVJ
- HSA
- IHV
- ING
- ISQ

#### Defining the XCF Group

To be able to communicate in certain situations, the automation manager instances and the automation agents belonging to one sysplex must be members of one and the same XCF group.

Systems with SA z/OS NetView instances that belong to the same XCF group must be defined in the Customization Dialogs in the same Group Policy Object of type sysplex. For details refer to the “Group Policy Object” chapter in IBM Tivoli System Automation for z/OS Defining Automation Policy.

#### Using SA z/OS Subplexes

You can divide your real sysplexes into several logical SA z/OS subplexes (an example is shown in Figure 2 on page 22). To do this you must define a specific XCF group suffix and a specific group policy object for each subplex. Each SA z/OS subplex must have its own automation manager. In each subplex there must also be only one shared automation manager takeover file and one shared schedule override file.

With SA z/OS subplexes you can run automation on systems of sysplexes in the same way as on single systems. This is required if you do not have shared DASDs for all your systems in the sysplex.

The group ID must be defined in an HSA parmlib member or INGXINIT for NetView.
To allow automation agents within the same physical sysplex, but to communicate via XCF rather than NetView Gateways within different SA z/OS subplexes, optionally an extended XCF Communication Group can be defined as well. For more details, see "Defining Extended XCF Communication Groups."

**Defining Extended XCF Communication Groups**

By default, an SA agent can only communicate with other agents that reside in the same SA subplex as described above. The introduction of the PLEXID parameter allows the extension of XCF communication between agents that reside in different SA subplexes.

The PLEXID parameter is a group suffix used to add the automation agent to an extended XCF communication group. This enables the automation agent to communicate via XCF with all other SA agents that were added to the same PLEXID group even though they are outside of the SA subplex. The PLEXID parameter may be defined in the member INGXINIT.

Along with the introduction of the PLEXID parameter, the TARGET parameter of all SA commands are enhanced such that they accept system name, domain id or SA subplex name of those agents that were added to the same PLEXID group. Use the command INGAMS in order to list all automation agents and automation managers that exist within the same PLEXID group.

If you want to use extended XCF communication it is strongly recommended to add all automation agents of the same SA subplex into the same PLEXID group.

If needed you may separate a specific SA subplex from the extended XCF communication group, see the figure in "Using SA z/OS Subplexes" on page 21. However, when you add all automation agents from all SA subplexes into the same PLEXID group then you have 'Single Point of Control' for all resources of all participating SA subplexes all over the physical Sysplex.

Another advantage of the enhanced XCF communication group is that you can reduce the number of gateway definitions. It helps to limit the number of NetView
System Operations Considerations

An SA z/OS instance for automation purposes is intended to run once on a system.

If you run multiple instances of SA z/OS on one system only one of those instances can perform automation tasks. This is important for GDPS environments where the automation tasks are performed by one SA z/OS instance (for example, within the GDPS PPRC environment) while another instance serves as a base for GDPS only (for example, the GDPS Global Mirror environment).

When running more than one SA z/OS instance, make sure that the underlying infrastructure is configured for the instance performing the automation tasks. This is required because certain parts of the infrastructure (MPF, MRT, PPI, LNK, LPA) exist only once per system.

Notes:

1. The MRT and PPIs reside within that SSI address space which is started first. Stopping this SSI will disable automation partly. Therefore it is recommended to keep this SSI highly available.
2. Any additional (NetView and SA z/OS) must match the releases residing in the LNKLST and LPA.

NetView ships two sample automation operators, AUTO1 and AUTO2. SA z/OS assumes that these tasks are available and have not been renamed. If they have been renamed, you must change the names in AOFMSGSY and the NetView style sheet, residing in the DSIPARM data set.

SA z/OS Hardware Interface: Important Considerations

The SA z/OS processor support commands and modules of Processor Operations and the BCP Internal Interface require a NetView task environment of CMD LOW to operate.

If you plan to use CMD HIGH task environments, be aware that ProcOps and BCPii function commands will not operate in such task environments. The ProcOps or BCPii function command will end prematurely with an error message that identifies the cause of the problem.

However you can still use NetView tasks with a CMD HIGH set for other purposes.

Automation Manager Considerations

This section presents automation manager considerations relevant to the installation process. For automation manager concepts that are of interest from an operator’s point of view, refer to IBM Tivoli System Automation for z/OS User’s Guide.

The automation manager is introduced as a separate address space. An installation requires one primary automation manager and may have one or more backups. The automation manager is loaded with a model of the sysplex when it initializes. It then communicates with the automation agents in each system, receiving...
Automation Manager Considerations

updates to the status of the resources in its model, and sending orders out to the
agents as various conditions in the model become satisfied.

A series of substeps is required to get the automation manager up and running for
your SA z/OS installation. These installation steps are described in this
documentation, but are not identified as being specific automation manager
installation steps.

Only the default installation of UNIX System Services is a prerequisite for the
automation manager. No USS file system or UNIX shell is required.

The automation manager must be defined by RACF (or an equivalent security
product) as a super user for UNIX System Services. The user that represents the
started tasks in your installation must be authorized for the OMVS segment.

Note: The system on which the automation manager should be started must be
defined as policy object System in the policy database that will be used to create
the automation manager configuration file that this automation manager uses (see
also “Step 18A: Build the Control Files” on page 122).

Storage Requirements
When the automation manager is started, it needs a constant amount of storage of
56 MB plus a variable part that depends upon the number of resources to be
automated.

The constant part consists of 40 MB for the automation manager code and 16 MB
for history information. The rule of thumb for the variable part is \( n \times 8 \text{ KB} \) where
\( n \) is the number of resources.

The sum of storage requirement according to the rule of thumb is:

\[ 40 \text{ MB} + 16 \text{ MB} + n \times 8 \text{ KB} \]

This formula covers the maximum storage requirements. However, the storage
requirements does not increase linearly with the number of automated resources.
Real measurements may be smaller than values retrieved with the rule of thumb
formula.

OMVS Setup
Because the automation manager requires OMVS, OMVS must be configured to
run without JES. (This means that OMVS should not try to initialize colony
address spaces under the JES subsystem as long as JES is not available.) Therefore
the definitions in the BPXPRMxx member must match one of the following:

- Either all FILESYSTYPE specifications with an ASNAME parameter are moved
  into a separate BPXPRM member. This can be activated via the automation
  policy by using the SETOMVS command after the message BPX10041 OMVS
  INITIALIZATION COMPLETE has been received.

- Or the parameter ‘SUB=MSTR’ is added to the ASNAME definition, for example:

  ```
  /***************************************************************************/
  /* ZFS FILESYSTEM */
  /***************************************************************************/
  FILESYSTYPE TYPE(ZFS) ENTRYPOINT(IOEFSCM)
  ASNAME(ZFS,'SUB=MSTR')
  ```

Note: In order to initialize without JES, the Automation Manager needs to be
defined as a superuser. If you use an OEM security product that does not initialize
until JES has initialized, superuser authority cannot be evaluated until JES is up and consequently JES cannot be started by SA z/OS. With z/OS version 1.10 or higher this restriction is solved and the Automation Manager can be initialized without JES and the need to be superuser. However BLOCKOMVS=YES still requires UID(0).

Recovery Concept for the Automation Manager

For sysplexwide and single-system automation, the continuous availability of the automation manager is of paramount importance.

To ensure the automation manager's functionality as automation decision server, the primary automation manager (PAM), must be backed up by additional automation manager address spaces called secondary automation managers (SAMs). Secondary automation managers are able to take over the function whenever a primary automation manager fails.

Therefore, it is recommended that you have at least one secondary automation manager running. For sysplexwide automation, the SAM should run on a different system than the PAM. It is important though that all automation managers (PAM and SAMs) run on systems which are in the same time zone.

To enable software or hardware maintenance in the sysplex, SA z/OS supports a command to force the takeover of the primary automation manager.

A takeover is only possible when the following requirements are met:

- All the automation manager instances must have access to a shared external medium (DASD) where the following is stored:
  - The configuration data (result of the ACF and AMC build process).
  - The schedule overrides VSAM file.
  - The configuration information data set — this is a mini file in which the automation manager stores the parameters with which to initialize the next time that it is started WARM or HOT.
  - The takeover file.

SA z/OS follows the concept of a floating backup because:

- The currently active automation manager has no awareness of the existence (and location) of possible backup instances.
- The location of the backup instances can change during normal processing without any interruption for the active automation manager.
- There is no communication between the primary automation manager and its backup instances during normal operation except when a SAM that is to become the new PAM informs the current PAM of that fact during a planned takeover.

This has the advantage that in normal operation, the processing is not impacted by a backup structure which can change.

Depending on the number of resources, the takeover time from a primary to a secondary automation manager is in the range of one to two minutes.

Manager-Agent Communication and Status Backup

SA z/OS provides XCF for establishing communication between the automation manager and the automation agents, and a VSAM data set (the takeover file) for keeping a backup copy of the status of the automated resources.
Automation Manager Considerations

As already pointed out, the work items and orders to the automation agents that are pending at takeover time are not stored in this implementation, so all these pending items will be lost when the PAM fails and a SAM takes over.

Figure 3 illustrates the timeline from the start of the automation manager (AM) through to its termination for the following cases:

- A planned stop and start of the automation manager
- An unexpected failure

Table 4 outlines the various recovery scenarios.

![Figure 3. Using Only the Takeover File for Status Backup](image)

**Table 4. Recovery Scenarios**

<table>
<thead>
<tr>
<th>Event</th>
<th>SA z/OS Recovery Action</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAM fails</td>
<td>SAM runs a takeover</td>
<td>The takeover file contains the state with the last successfully processed work item</td>
</tr>
<tr>
<td>PAM detects a severe error condition</td>
<td>PAM terminates and SAM runs a takeover</td>
<td>The takeover file is used to rebuild the resource object structures in case of a takeover or next hot start</td>
</tr>
<tr>
<td>System with the PAM fails</td>
<td>SAM runs a takeover</td>
<td>The takeover file is used to rebuild the resource object structures in case of a takeover or next hot start</td>
</tr>
</tbody>
</table>
Chapter 4. Planning to Install Alert Notification by SA  z/OS

Introduction of Alert Notification by SA  z/OS

SA  z/OS alert notification is triggered by the invocation of the INGALERT command. It can be used to perform one or more of the following tasks:

- Start notification escalation by IBM Tivoli System Automation for Integrated Operations Management (SA IOM)
- Display an event on a centralized operator console such as IBM Tivoli Enterprise Console (TEC) or IBM Tivoli Netcool/OMNIbus (OMNIbus)
- Create a trouble ticket in a service desk application such as IBM Tivoli Service Request Manager® (TSRM)
- Perform an arbitrary task in a user-defined alert handler

The following communication methods are available for alert notification:

- Use the peer-to-peer protocol of SA IOM to start a REXX script on the SA IOM server
- Send a Tivoli Event Integration Facility (EIF) event
- Send XML data to IBM Tivoli Directory Integrator (TDI) and from there trigger the creation of the trouble ticket
- Pass parameters to the user-defined alert handler that is called as a NetView command

Note: EIF events and the TDI interface can be used to perform a variety of tasks or to integrate other operator consoles or service desk applications. The ones listed above are provided by SA  z/OS as samples.

The behavior of INGALERT is controlled with the INGCNTL command at the system level, by a resource's Inform List at the resource level and even more granularly by CODE entries for the INGALERT entry in the MESSAGES/USER DATA policy item.

For details about the INGALERT and INGCNTL commands see IBM Tivoli System Automation for z/OS Programmer's Reference.

Alert Notification Infrastructure in SA  z/OS

When INGALERT is called in a SA  z/OS subplex the system tries to reach all specified targets by passing the request from one agent to another. If, for instance, INGALERT is called on SYS1 in order to start an SA IOM notification escalation, but SYS1 has no connection to the SA IOM server, the request is routed to SYS2 and SYS3 and so on, until the SA IOM server can be reached.

This implies that you need not have all the connectivity to your distributed products on each system in the subplex, although you should have it at least on
Introduction of Alert Notification by SA z/OS

one, of course. This is true for all of the communication methods mentioned in "Introduction of Alert Notification by SA z/OS" on page 27.

For details about the alert notification infrastructure see IBM Tivoli System Automation for z/OS Customizing and Programming.

Integration via SA IOM Peer-To-Peer Protocol

The integration of SA z/OS with SA IOM is based on the SA IOM peer-to-peer protocol. This requires that the SA IOM server must accept the system running the SA z/OS agent (or agents) as valid peers. For details about setting up SA IOM see IBM Tivoli System Automation for Integrated Operations Management User's Guide.

Through this protocol a REXX script is triggered on the SA IOM server that starts the notification escalation process asynchronously. A return code and eventually an error message are passed back to SA z/OS indicating whether the notification escalation could be started.

Note that it is not verified whether an operator can actually be notified by SA IOM.

To use integration via the SA IOM peer-to-peer protocol you must be able to set up a TCP/IP connection to the SA IOM server from at least one system that is running an SA z/OS agent.

See "Enabling Alert Notification via SA IOM Peer-To-Peer Protocol" on page 119.

Integration via EIF Events

SA z/OS can send out EIF events as the result of an INGALERT invocation. To create such an EIF event the message adapter of the IBM Tivoli Event/Automation Service (EAS) is used via the program-to-program interface (PPI).

To use integration via EIF events there must be an EAS on at least one system that is running an SA z/OS agent.

Because SA z/OS communicates only with EAS it does not matter which product receives the EIF event and which platform it is running on. There is, however, some customization required for these products.

For more details about how to set up the EAS and configure TEC and OMNIbus on Windows, see "Enabling Alert Notification via EIF Events" on page 119.

Integration via Trouble Ticket Information XML

When the creation of a trouble ticket is desired INGALERT sends XML data to a known URL (host and port). It is expected that the server sends back a response indicating success or failure and possibly an error message. It is irrelevant what kind of server this is and which platform it runs on. However, it is recommended that the server is a TDI Runtime Server. Samples are provided for this server and the customization is described in "Enabling Alert Notification via XML" on page 120.

To use integration via trouble ticket XML you must be able to set up a TCP/IP connection to a TDI server from at least one system that is running an SA z/OS agent.
Integration by User-defined Alert Handler

When INGALERT is told to inform a user-defined alert handler it calls the specified command synchronously in the NetView environment. Parameters are passed to the alert handler and a convention regarding return code and output messages must be obeyed. For details about the user-defined alert-handler see INGALERT in *IBM Tivoli System Automation for z/OS Programmer’s Reference*.

To use integration by user-defined alert handler the code must be accessible from at least one system that is running an SA z/OS agent.

For more details see "Enabling Alert Notification via User-Defined Alert Handler" on page 121.
Introduction of Alert Notification by SA z/OS
Chapter 5. Planning for Automation Connectivity

The Focal Point System and Its Target Systems 31
Defining System Operations Connectivity 31
Multiple NetViews 31
Overview of Paths and Sessions 31
Message Forwarding Path 32
Gateway Sessions 32
Automatically Initiated Terminal Access
Facility (TAF) Fullscreen Sessions 34
Using Focal Point Services 34
Defining Processor Operations Communications Links 35
Meeting Availability Requirements 35
Backup Support Element 35
Alternate Focal Point System 35
Alternate Focal Point System for HTTP Connections 35
Alternate Focal Point for SNMP connections 35
BCP internal interface considerations 35
Task Structure for Processor Operations 36
Target Control Tasks 36
Message Monitor Tasks 36
Recovery, Start, Polling and General Management Tasks 36
Planning Processor Operations Connections 37
Preparing the Processor Operations Focal Point System Connections 37
TCP/IP Firewall-Related Information 37
Preparing the Alternate Focal Point System Connections 37
Connection Example 38
Preparing the Target System Connections 38
Defining I/O Operations Communications Links 39

This chapter provides background on SA z/OS. It includes what a focal point system is and what targets are, and how to define a network of interconnected systems, known as an automation network, to SA z/OS for purposes of monitoring and controlling the systems. The procedures and examples in this chapter assume that VTAM definitions for systems in the automation network are in place and available as input.

The Focal Point System and Its Target Systems

SA z/OS allows you to centralize the customization, monitoring, and control functions of the multiple systems or images that make up your enterprise using a single, centrally located z/OS system. This controlling z/OS system is called the focal point system. The systems it controls are called target systems. These systems communicate using XCF and NetView facilities.

Defining System Operations Connectivity

This section discusses the following aspects of defining system operations connectivity:

• "Multiple NetViews"
• "Overview of Paths and Sessions"

Multiple NetViews

The number of NetViews that run in your SA z/OS complex affects how you plan for it. SA z/OS can operate with just one NetView at its focal point. It is your decision whether you want to run the Networking Automation and the System Automation on separate NetViews.

Overview of Paths and Sessions

This section provides an overview of the following:

• "Message Forwarding Path" on page 32
• "Gateway Sessions" on page 32
Message Forwarding Path

SA z/OS generates and uses messages about significant actions that it detects or takes such as a resource status change. In addition to sending these messages to operators on the same system, SA z/OS can forward them from target systems to a focal point system and can route commands and responses between systems, using a message forwarding path. This path is defined in your policy. Key components in a message forwarding path include:

- A primary focal point system
- A backup focal point system
- A target system or systems
- Gateway sessions connecting systems. Gateway sessions use inbound and outbound gateway autotasks. Communication is via the NetView RMTCMD or XCF when the focal point system and target system are in the same sysplex.

Using a message forwarding path, a focal point system can monitor several target systems.

SA z/OS uses notification messages to update the status of resources displayed on the status display facility (SDF). Routing notification messages over the message forwarding path helps consolidate monitoring operations for multiple systems on the SDF at a focal point system. See IBM Tivoli System Automation for z/OS User’s Guide for details on configuring SDF for a focal point system-target system configuration.

Gateway Sessions

Outbound and Inbound Gateway Autotasks:

Each gateway session consists of:

- Two gateway autotasks on each system:
  - One autotask for handling information outbound from a system, called the outbound gateway autotask. This establishes and maintains all connections to other systems. It sends messages, commands, and responses to one or more systems.
  - One autotask for handling information incoming from another system, called the inbound gateway autotask. A system can have one or more inbound gateway autotasks, depending on the number of systems to which it is connected.

![Figure 4](image-url) shows a single gateway between two SA z/OS agents, ING01 and ING02.
Defining System Operations Connectivity

There is one task handling all outbound data. This task is set up at SA z/OS initialization time. Normally the task has a name that begins with GAT and ends with the domain name. So for ING01, the gateway task is GATING01.

When VTAM becomes active, the gateway task (GATOPER) issues a CONNECT call to the remote system, ING02 in our example. If the GATING01 task on the remote system is not already active, it will be started automatically by NetView.

All requests initiated by system ING01 and destined for system ING02 use the task pair GATING01. Likewise all requests that originate on system ING02 and are destined for system ING01 use the pair GATING02. In other words the communication is half-duplex. There is one task pair responsible for the outbound traffic while another task pair is in charge of the inbound traffic. Each pair consists of a sender - running on the local system and receiver that runs on the remote system.

Disallowing the starting of the receiver task protects the local system from getting requests from the remote system.

The task structure is similar when using XCF as the communication vehicle. Using the "GATxxxx" task as the receiving and processing task on the remote side gives a dedicated task pair for the communication between the two systems. This task pair exists twice, once for each outbound communication. It is important to notice that the standard RPCOPER is not used for the processing of the remote procedure call.

In the automation policy for each system in an automation network, you need to define only the outbound gateway autotask (see IBM Tivoli System Automation for z/OS Defining Automation Policy). However, in the NetView DSIPARM data set member DSIOPF, you must define all gateway autotasks, both inbound to and outbound from a system, as operators.

You define the outbound gateway autotask by defining the GATOPER policy item for the Auto Operators policy object in the customization dialog. You must specify an operator ID associated with the GATOPER function in the Primary field on the Automation Operator NetView panel. See IBM Tivoli System Automation for z/OS Defining Automation Policy for more information.

For this example, the operator ID for the system CHI01 outbound gateway autotask is GATCHI01. Similarly, any operator ID for an inbound gateway autotask is the prefix GAT combined with the inbound gateway domain name.

Figure 5 on page 34 shows three systems: CHI01, ATL01, and ATL02. System CHI01 is the focal point for forwarding messages from target systems ATL01 and ATL02. In Figure 5 on page 34 gateways are designated as follows:

O Outbound gateway autotask
I Inbound gateway autotask.
How Gateway Autotasks Are Started:
Gateway autotasks establish a connection between systems when any system receives the following NetView message:

DSI112I NCCF READY FOR LOGON AND SYSTEM OPERATOR COMMANDS

When this message is received, the following steps occur:
1. The outbound gateway autotask tries to establish an outbound session with the remote system.
2. A gateway session between two systems is established when the outbound gateway autotask has established its outbound session to the remote system.

This process automatically establishes outbound and inbound connections for systems without human operator intervention.

How Gateway Sessions Are Monitored:
Optionally, gateway sessions can be monitored by a command that is executed periodically. The time interval is set in the Gateway Monitor Time field in the SYSTEM INFO policy item for the System policy object.

See IBM Tivoli System Automation for z/OS Defining Automation Policy for details. The ID of the timer created to monitor gateway sessions is AOFGATE. This timer will not be set if NONE is entered for Gateway Monitor Time.

If SA z/OS detects that any gateway session is inactive during the monitoring cycle, it tries to restart the session.

Automatically Initiated Terminal Access Facility (TAF) Fullscreen Sessions
Using the FULL SESSIONS policy item of the Network policy object, you can set up automatically-initiated terminal access facility (TAF) fullscreen sessions from within SA z/OS. IBM Tivoli System Automation for z/OS Defining Automation Policy describes how to define applications with which SA z/OS operators can establish TAF sessions automatically using the SA z/OS NetView interface.

Using Focal Point Services
Once an automation network is configured, you can use the message forwarding path to route messages, commands, and responses between systems. SA z/OS operators can display the status of gateway autotasks and TAF fullscreen sessions.
Defining System Operations Connectivity

using the SA z/OS operator commands. Details on these operator activities are in *IBM Tivoli System Automation for z/OS User’s Guide.*

Defining Processor Operations Communications Links

After determining that you plan to use the processor operations functions, you must decide the type of communication link from your focal point system to your support element. Processor operations supports the following types of communication connections:

- HTTP over TCP/IP (SSL)
- SNMP
- TCP/IP

Meeting Availability Requirements

In order to reduce the interruption time in case of processor operations communication problems, the following facilities are available:

- Backup Support Element
- Alternate focal point system

Backup Support Element

System z processors have a second Support Element (SE) installed, operating in hot-standby mode. If the primary Support Element fails, the backup SE is automatically activated as the new primary Support Element. The SE configuration information is always duplicated, so the new primary SE has the same configuration information as the failing one including the IP network addresses.

Alternate Focal Point System

An alternate focal point system can be used, in addition to the primary focal point system, to minimize the effect of a focal point system outage. If a focal point system must remain operational all the time, an alternate focal point system can be operated in a take-over mode.

Alternate Focal Point System for HTTP Connections

If you plan to use a second focal point system for your processor operations HTTP connections, make sure that the TCP/IP stack and the PAGENT are always up and that your IP network allows the SSL communication between the alternate focal point and the ensemble HMCs.

Alternate Focal Point for SNMP connections

If you plan to use a second focal point system for your processor operations SNMP connections, make sure that the TCP/IP USS stack is always up and that your IP network allows the communication between the alternate focal point and the Support Elements.

BCP internal interface considerations

If you have configured SA z/OS to use the BCP internal interface for the sysplex hardware automation, each system being a member of the sysplex has its processor hardware connection activated and can issue hardware requests to the SEs of the other sysplex members. The SA z/OS internal code routes the supported hardware commands only to a system in the sysplex with a functioning hardware interface to make sure the request can be processed successfully.
Task Structure for Processor Operations

For processor operations there is a task structure that is modular; distinct types of SA z/OS tasks handle different work assignments. The types of SA z/OS tasks are:

- Target control tasks
- Message monitor tasks (used for SNMP, TCP/IP and HTTP connections only)
- Recovery task
- Start task
- Polling task

SA z/OS allows up to 999 tasks of each of the first three types, but only one recovery task and one processor operations start task. Because SA z/OS tasks are z/OS tasks that require system services and also add to the load running in the NetView address space, you should only define as many tasks as are needed.

The following guidelines help you match the number of SA z/OS tasks to your SA z/OS configuration.

- The number of message monitoring tasks for target systems connected with an SNMP connection should be identical to the number of target control tasks in your environment.
- The number of target control tasks should be less than or equal to the number of target hardware defined. If you plan to use the processor operations group and subgroup support for the common commands, the total number of target control tasks should be equal to the number of concurrently active target hardware systems.
- In consideration of focal point performance, limit the total number of tasks to a number your system can handle.

Target Control Tasks

The number of target control tasks is automatically calculated and set.

Target control tasks process commands. A target system is assigned to a target control task when the target system is initialized. More than one target system can be assigned to the same target control task. A target control task is a NetView autotask.

Message Monitor Tasks

The number of message monitor tasks is automatically calculated and set.

Message monitor tasks receive SNMP traps from the Support Element's SNMP clients, messages from the PSMs and their associated VM second level systems and the notifications from the HMC Web Services API message broker at the focal point system. The traps, messages and notifications are broadcast to the appropriate tasks and operators.

Recovery, Start, Polling and General Management Tasks

Automation for resource control messages runs under the recovery task, which is a NetView autotask. Processor operations also uses the recovery task for processing of recovery automation commands. Normally, this task is idle. It is generated automatically when you generate NetView autotask definitions from the configuration dialogs.
The startup task, a NetView task, is used to establish the processor operations environment with the NetView program and to start the other NetView tasks needed for processor operations to function. The startup task is only active during processor operations start (ISQSTART).

The polling task, another NetView task, is used to poll the processors using NetView connections. You determine both the polling frequency and polling retries to be attempted. (These polling functions are specified using the NetView connection path definition panels in the configuration dialogs.) This task is generated automatically when you generate the NetView Autotask definitions from the customization dialogs. This NetView task enables SA z/OS to verify and update operations command facility-based processor status.

The general management task is used for message automation in case the recovery task is not available because of other workloads.

Planning Processor Operations Connections
This section describes making the hardware connections. It is divided into subsections for each set of hardware connections:

- **Preparing the Processor Operations Focal Point System Connections** and **Preparing the Alternate Focal Point System Connections** for focal point system connections
- **Preparing the Target System Connections** on page 38 for target system connections. This section also discusses complex connection configurations.

Preparing the Processor Operations Focal Point System Connections
The physical path for the focal point system consists of connections from the HMC, SE, or PSM to the focal point system. SA z/OS processor operations supports the following types of communication connections:

- HTTP over TCP/IP(SSL)
- SNMP
- TCP/IP

TCP/IP Firewall-Related Information
The TCP/IP SNMP connections of ProcOps use port number 3161. This is the port number that Support Elements or Hardware Management Consoles use to communicate with SA z/OS ProcOps or other applications using the System z API. In case you have firewalls installed between the processor LAN and the LAN that SA z/OS ProcOps belongs to, make sure port 3161 is registered to prevent SE/HMC responses from being rejected.

The TCP/IP HTTP ensemble connections of ProcOps use port numbers 6167 and 61612. These are the port numbers that the Hardware Management Consoles use to communicate with SA z/OS ProcOps or other applications using the Web Services API. In case you have firewalls installed between the processor LAN and the LAN that SA z/OS ProcOps belongs to, make sure ports 6167 and 61612 are registered to prevent HMC connections from being rejected.

Preparing the Alternate Focal Point System Connections
An alternate focal point system can be connected to your DP enterprise in addition to the primary focal point system.
Planning Processor Operations Connections

The physical connection path for the alternate focal point system is identical to that for the primary focal point system. As with the primary focal point system, SA z/OS processor operations supports the following types of communication connections:

- HTTP over TCP/IP (SSL)
- SNMP
- TCP/IP

Connection Example

Figure 6 shows an alternate focal point system as well as a primary focal point system connected from an IP network to the processor hardware LAN.

With SNMP, a connection can be established either to the Support Element of a CPC, or to an HMC. This HMC must have the CPCs defined you want to manage.

With TCP/IP, a connection can be established to a ProcOps Service Machine on a VM host (PSM).

Preparing the Target System Connections

The supported processor hardware allows you to use the attached Support Element or an HMC (SNMP connections only), connected to the processor hardware LAN for hardware operations management tasks and for operating system control. The Console Integration (CI) function of the SE or HMC is used by processor operations to send commands to an operating system and to receive messages.
Planning Processor Operations Connections

from an operating system. The operations management interfaces of the SE or HMC are used to perform tasks like SYSTEM RESET, LOAD, TEMP.CAPACITY or ACTIVATE.

The usage of CI by processor operations is intended to automate system initialization and recovery tasks. For day-to-day console operation tasks, processor operations CI usage should supplement the operating system command routing facilities of SA z/OS or the available console devices like the 2074 control units.

Defining I/O Operations Communications Links

When you use I/O-Ops on one system to make an operational change to an I/O resource, such as a shared Switch Director, it coordinates the change with other copies of I/O-Ops on other systems. This is especially important when the result of the action you are taking removes connectivity, that is, disables I/O paths, so that the systems do not lose access to critical resources. Each copy of I/O-Ops interacts with its local system image (for example, through VARY) so that the operating system has the chance to vote on the changes. If one system fails in VARYing the path of a device, I/O-Ops interprets this as a vote of no and fails the operation. This behavior is called safe-switching (see “Safe Switching” in IBM Tivoli System Automation for z/OS Operator’s Commands). The copy of I/O-Ops that you initiated the operation from then interacts with the other copies on the affected system images to back out VARYs that were successful.

The copies of I/O-Ops across your systems also use the network to share information with each other on changes to the I/O configuration and to provide displays that collect I/O information from multiple systems. To do this, the I/O-Ops functions on each system image need to intercommunicate by establishing TCP/IP or VTAM sessions between each other. All systems that share access to a given Switch Director should run I/O-Ops to provide safe-switching protection. Those copies of I/O-Ops that do share access to a Director automatically discover each other and establish sessions each time they start.

You can also use the Reset Host function of I/O-Ops to force two copies of I/O-Ops that do not share any Switch Directors to establish communications. This is useful if you want to benefit from the I/O-Ops multisystem I/O graphic displays or use its multisystem version of Remove CHP, Restore CHP, Remove Device, or Restore Device, even across system images that don’t use Switch Directors or have no reason to share them.

To plan for this function, you must review the I/O configuration across the systems that you will define as an enterprise in I/O-Ops. You should plan to include in one enterprise all system images that share a given Switch Director, in order to benefit from the I/O-Ops configuration change protection and displays.

To enable the VTAM sessions, you must create VTAM definitions as described in “Step 19B: Perform VTAM Definitions for I/O Operations” on page 124 to support communications between I/O-Ops defined as a VTAM application in each of them.

Where images do not automatically use those definitions to start sessions, because they do not share Switch Directors, you should plan local procedures to use the I/O-Ops Reset Host function to force I/O-Ops applications to start the sessions.
Chapter 6. Planning for Integration with IBM Tivoli Monitoring

Planning for the SA z/OS ITM Agent

SA z/OS ships an own IBM agent to show automation data on the Tivoli Enterprise Portal (TEP).

The SA z/OS monitoring agent is a member of the IBM Tivoli Monitoring Services family of mainframe monitoring products. It monitors the automation environment and resources it contains in systems and sysplexes. For further details about planning such an environment are provided in IBM Tivoli System Automation for z/OS Monitoring Agent and User’s Guide.

Planning for SOAP over HTTPS

The default communication for talking over the SOAP interface is by a HTTP link. As this is not secure, there is also the option of setting up an HTTPS connection.

To do this you will need to modify the parameters for your TCPIP stack.

For further information, refer to "Step 37B: Enabling SOAP over HTTPS for a TEMS" on page 158.

Planning for Looping Address Space Suppression

Prerequisites

In order to use the Looping Address Space Suppression automation you need:

- OMEGAMON XE for z/OS installed and running
- A Tivoli Enterprise Monitoring Server (TEMS) that is receiving data from the OMEGAMON XE for z/OS instance and which is running a SOAP server
- If you wish to use HTTPS communication between SA z/OS and the Tivoli Enterprise Monitoring Server, you need to plan for it to be enabled.

Data Gathering

Ask your application programmers and batch schedulers if they have any batch jobs or other programs that run for a long time in CPU intensive loops. If they can identify them now, you can enter them into your automation policy at the definition stage as known false positives which should avoid accidents later on.

Even though you should start running with automation set to LOG only, it is better to obtain a list of predicted false positives from your application experts than to simply hope that they all appear during your test phase.
Chapter 7. Naming Conventions

SA z/OS System Names .................................................. 43
Cloning on z/OS Systems .............................................. 43
Further Processor Operations Names .............................. 44
Switch Director Ports .................................................... 44
Reasons for Naming Switch Ports .................................. 44
Suggestions for Naming Switch Director Ports ............... 44
Naming CHPID Ports ..................................................... 44
Naming Control Unit Ports ............................................ 44
Methods of Naming Ports .............................................. 44
Using Port Logical Names ............................................. 45
Using Generic Logical Names ....................................... 46
Command Usage Examples with Generic Logical Names .... 46

SA z/OS System Names

The information in this section describes name requirements for z/OS systems and for processor operations functions.

All system names defined with the customization dialog in one policy database must be unique.

If your system names currently contradict this restriction, you must change the names before using SA z/OS.

System names defined in the customization dialog for z/OS, VM, TPF, or LINUX systems can have up to 20 characters and must be unique within the SA z/OS enterprise.

When you name elements of your SA z/OS processor operations, use a logical format to create names that are clear to the people using them. The following names can consist of 1 to 8 alphanumeric characters (A-Z, a-z, 0-9, #, $, @), cannot contain blanks, and must begin with an alphabetic character:

- Processor or target hardware names
- Target system names
- Focal point name

Processor or target hardware system names, target system names, group names for target systems, and subgroup names for target systems must all be different from one another. Target system names must also be different from processor operations names. For any given system, however, its system name can equal its own processor operations name.

Group and subgroup names for target systems can consist of up to 20 alphameric characters.

Sysplex group names should not be more than 8 characters in length because they are used to address the sysplex or subplex.

Cloning on z/OS Systems

The SA z/OS cloning capability allows you to specify up to 36 clone IDs to identify a system and to identify an application. These clone IDs are then used to qualify the application job name to ensure a unique job name for each system. The names given to each of these clones must be unique. The z/OS system symbolics and the NetView &domain. variable can also be used.
Further Processor Operations Names

Activation profiles (Reset, Image, Load) names are processor (CPC) related and are manually defined at the HMC or SE. When doing this with processor operations, note that these names must consist of the characters A-Z and 0-9. Image (LPAR) profile names can be up to eight characters long; Reset and Load profiles can have a length of up to sixteen characters.

Switch Director Ports

This section offers suggestions for naming Switch Director ports (dynamic switch ports) and fully utilizing these names in I/O-Ops display and connectivity commands.

Reasons for Naming Switch Ports

Assigning names to switch ports:

- Provides an indication of what is on that port. For example, CP01.SYSA.CHP38 indicates that this port is physically connected to processor CP01, on system SYSA, on CHPID 38.
- Allows you, when issuing I/O-Ops connectivity commands, to refer to ports by name. For example, BLOCK 3490.46233.CU1.E * blocks the port connected to interface E of control unit side 01, on the 3490 control unit with serial number 46233. See "Using Port Logical Names" on page 45.
- Allows you, when issuing I/O-Ops connectivity commands, to change connectivity of an entire system to a control unit. For example, PROHIBIT CP01.SYSA* 3990.35182* * removes connectivity from all ports on system SYSA of processor CP01, from all ports on the 3990 control unit with serial number 35182. See "Using Generic Logical Names" on page 46.

Suggestions for Naming Switch Director Ports

When naming ports, you should choose names that help identify what the port is connected to. This simplifies the task of entering commands when connectivity changes are required. Following are some suggestions for naming CHPID ports and control unit ports, followed by a figure displaying those ports in an actual configuration.

Naming CHPID Ports

Name the CHPID ports with three parts: the processor name, followed by the system image name, followed by the CHPID number. For example:

```
CP02.SYSC.CHP40
```

is the port name associated with CHPID 40, on system SYSC of processor CP02.

Naming Control Unit Ports

Name the control unit ports with four parts: the device type, followed by the serial number, followed by the storage cluster (or control unit side), followed by the interface letter. For example:

```
3990.35182.SC1.E
```

is the port name associated with the 3990 with serial number 35182, on storage cluster 1, interface E.
Methods of Naming Ports

You can assign names to ports using the following methods:

- The WRITE command.
  
  You can use the following command to write the name CP01.SYSB.CHP38 to port D3 on switch 100:
  ```c
  WRITE CP01.SYSB.CHP38 (D3) 100
  ```

  This command is available on the operator command line, the ISPF command line, the workstation feature command builder, and the port settings notebook.

- The matrix editor.
  
  You can use the matrix editor to enter a name next to the port number; then send the matrix to the switch. This interface is available on ISPF and the workstation.

- EXECs.
  
  You can create an EXEC to send a series of name assignments to a dynamic switch with, for example, the following commands:
  ```c
  WRITE CP01.SYSB.CHP38 (D3) 100
  WRITE 3990.35182.SC1.E (F1) 100
  ```

- The WRITE switch (WRITESWCH).
  
  You can create an EXEC to issue the WRITESWCH command, placing the new name in the WRITESWCH data block.

Using Port Logical Names

Once names are assigned to ports, you can issue a single command to change the connectivity of one or more switches. The following command blocks the port named 3490.46233.CU1.F on switch 100:
```c
BLOCK 3490.46233.CU1.F 100
```
**Naming Conventions**

The following command blocks the port named 3490.46233.CU1.F on any switch that contains that name:

```
BLOCK 3490.46233.CU1.F *
```

The following command looks for any switch that has both names, CP02.SYSC.CHP42 and 3490.46233.CU1.F:

```
PROHIBIT CP02.SYSC.CHP42 3490.46233.CU1.F *
```

If both names exist on any switch, those two ports are prohibited from each other.

The use of these commands is limited to one change per switch.

**Using Generic Logical Names**

I/O-Ops provides the ability to use an asterisk as a wild card character in commands that use port names. This allows you to make more than one change on each switch.

You can use an asterisk as a name in the DISPLAY NAME, BLOCK, UNBLOCK, ALLOW, and PROHIBIT connectivity commands. For example, suppose you issue the following command:

```
PROHIBIT CP02* 3490.46233* *
```

All switches are searched for ports with names beginning with CP02 (for example, CP02.SYSA.CHP34 and CP02.SYSB.CHP70) and ports with names beginning with 3490.46233 (for example, 3490.46233.CU1.B and 3490.46233.CU0.D). If found, those ports are prohibited from each other.

By using a single command, you can remove connectivity from a entire system to a control unit. However, for this to work properly:

- The names must be consistent across all switches.
- You must issue the connectivity commands from an I/O-Ops system that has access to all switches.

Any names that are not an exact match cause no errors. Any switches that are not affected because they were not accessed cause no errors. You only receive notification if:

- No name match is found on any one switch (warning return code).
- No name match is found on any switch (failure return code).

**Command Usage Examples with Generic Logical Names**

The following are some examples of how you can issue I/O-Ops commands using generic logical names:

- Use `DISPLAY NAME` to show information about the ports specified:

```
DISPLAY NAME CP02.SYSC* *
```

<table>
<thead>
<tr>
<th>SWCH</th>
<th>STATUS</th>
<th>PORT NAME</th>
<th>DEVN</th>
<th>LSN</th>
<th>PORT</th>
<th>H</th>
<th>B</th>
<th>C</th>
<th>P</th>
<th>DEF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>I/O</td>
<td>CP02.SYSC.CHP22</td>
<td>0400</td>
<td>02</td>
<td>C6</td>
<td>O</td>
<td>B</td>
<td>CH</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CP02.SYSC.CHP39</td>
<td>0100</td>
<td>00</td>
<td>EC</td>
<td>P</td>
<td>CHCU</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CP02.SYSC.CHP35</td>
<td>0100</td>
<td>00</td>
<td>C5</td>
<td>CH</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CP02.SYSC.CHEP0</td>
<td>0200</td>
<td>01</td>
<td>E0</td>
<td>CH</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Use `DISPLAY NAME` to show information about the ports for the 3490 with serial number 46233:
• Use BLOCK to remove access to a 3490 with serial number 46233 (four variations):
  BLOCK 3490.46233.CU0.D * (for one port on some switch)
  BLOCK 3490.46233.CU0* * (for one CU side)
  BLOCK 3490.46233* + (for one CU)
  BLOCK 3490.46233* 100 (for one CU through SW 100)

Notice that the first BLOCK command affects only one switch because there should be only one port with the name 3490.46233.CU0.D.

• Use PROHIBIT and then ALLOW to remove access from one host to one 3490 and give access to another host:
  PROHIBIT CP02.SYSC* 3490.46233* * (affects multiple paths)
  ALLOW CP01.SYSA* 3490.46233* * (affects multiple paths)

• Use PROHIBIT to remove access from one host to all 9343s to show results:
  PROHIBIT CP02.SYSA* 9343* *
  DISPLAY NAME 9343* *

In summary, you can use generic logical names to control system connectivity without being concerned about individual ports and switches.
Part 2. Installation and Configuration

This part provides instructions for:

- Chapter 8, “SMP/E Installation,” on page 51
- Chapter 9, “Base SA z/OS Configuration Using the Configuration Assistant,” on page 55
- Chapter 10, “Traditional SA z/OS Configuration,” on page 73
- Chapter 11, “Security and Authorization,” on page 161
- Chapter 12, “Configuring SA z/OS Workstation Components,” on page 197

Before SA z/OS is started on the target systems, you must install and configure the product.

Perform the SMP/E installation, as documented in Chapter 8, “SMP/E Installation,” on page 51.

After the installation is done, you need to perform configuration. There are two alternative configuration approaches:

- Basic configuration using a configuration assistant:
  - This approach provides an automated, convenient and safe way to generate installation-specific configuration members, which system programmers then can inspect, adapt (if still necessary) and use. If you intend to configure the base product using the configuration assistant, follow the documentation in Chapter 9, “Base SA z/OS Configuration Using the Configuration Assistant,” on page 55.

- Traditional manual configuration:
  - Samples are delivered with SA z/OS that manually must be adapted by system programmers as described in this section. If you intent to configure the product manually, follow the documentation in Chapter 10, “Traditional SA z/OS Configuration,” on page 73.
Chapter 8. SMP/E Installation

1. Perform the SMP/E installation on the appropriate system. Apply any available maintenance.

2. The security administrator ensures that the system programmer has ALTER access to the HLQs where they are to deploy the SMP/E target libraries to. You must use a system where the Customization Dialog is run and the systems where SA z/OS is deployed for automation.

3. If the SA z/OS installation is planned on other systems than the SMP/E system, then the system programmer must transmit the SMP/E target libraries to the system where the Customization Dialog is run and the systems where SA z/OS is deployed for automation. You can duplicate these data sets or wait until you must distribute an update to them. Updating the data sets that the product is using is not recommended.

4. On the system where the Customization Dialog is run, the system programmer must make the INGEDLG routine available to the automation programmer under ISPF. It is suggested that access to the Customization Dialog is restricted as the automation policies that are used to edit, compose part of your operation runtime data. See the IBM Tivoli System Automation for z/OS Planning and Installation Guide.

5. The security administrator must provide the following permissions for the automation programmer:
   - READ access to the SMP/E target libraries
   - ALTER access to the HLQ used for the Automation Policy Databases (PDB) and the automation control data sets.

Table 5 shows a list of target data sets as provided by the SMP/E installation process to be used for production on your system.

Table 5. Target Data Sets

<table>
<thead>
<tr>
<th>Data Set Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ING.SINGIMSG</td>
<td>ISPF messages</td>
</tr>
<tr>
<td>ING.SINGINST</td>
<td>SMP/E jobs to install the product alternatively to using SMP/E dialogs</td>
</tr>
<tr>
<td>ING.SINGIPDB</td>
<td>Policy database samples</td>
</tr>
<tr>
<td>ING.SINGIPNL</td>
<td>ISPF panels</td>
</tr>
<tr>
<td>ING.SINGIREX</td>
<td>ISPF REXX execs</td>
</tr>
<tr>
<td>ING.SINGISKL</td>
<td>ISPF skeletons</td>
</tr>
<tr>
<td>ING.SINGITBL</td>
<td>ISPF tables</td>
</tr>
<tr>
<td>ING.SINGJMSG</td>
<td>Kanji NetView messages</td>
</tr>
<tr>
<td>ING.SINGJPNL</td>
<td>Kanji NetView panels</td>
</tr>
<tr>
<td>ING.SINGMOD1</td>
<td>Different SA z/OS modules</td>
</tr>
<tr>
<td>ING.SINGMOD2</td>
<td>Different SA z/OS modules in LINKLST</td>
</tr>
<tr>
<td>ING.SINGMOD3</td>
<td>Different SA z/OS modules in LPALIB</td>
</tr>
</tbody>
</table>
Step 1: SMP/E Installation

Table 5. Target Data Sets (continued)

<table>
<thead>
<tr>
<th>Data Set Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ING.SINGNMSG</td>
<td>NetView messages</td>
</tr>
<tr>
<td>ING.SINGNPNL</td>
<td>NetView panels</td>
</tr>
<tr>
<td>ING.SINGNPRF</td>
<td>NetView profiles</td>
</tr>
<tr>
<td>ING.SINGNPRM</td>
<td>NetView DSIPARM samples</td>
</tr>
<tr>
<td>ING.SINGNREX</td>
<td>NetView REXX execs</td>
</tr>
<tr>
<td>ING.SINGTREX</td>
<td>TSO REXX execs</td>
</tr>
<tr>
<td>ING.SINGPWS1</td>
<td>NMC exploitation code</td>
</tr>
<tr>
<td>ING.SINGJPWS</td>
<td>Japanese NMC exploitation code</td>
</tr>
<tr>
<td>ING.SINGSAMP</td>
<td>General samples</td>
</tr>
<tr>
<td>ING.SINGMSGV</td>
<td>For VM second level systems support</td>
</tr>
<tr>
<td>ING.SINGOBJV</td>
<td>For VM second level systems support</td>
</tr>
<tr>
<td>ING.SINGREXV</td>
<td>For VM second level systems support</td>
</tr>
<tr>
<td>ING.SINGIMAP</td>
<td>Mapper files for Autodiscovery</td>
</tr>
<tr>
<td>ITM.TKANCUS</td>
<td>Installation CLISTs for Tivoli Enterprise Portal (TEP) support</td>
</tr>
<tr>
<td>ITM.TKANMODL</td>
<td>Load modules for TEP support</td>
</tr>
<tr>
<td>ITM.TKANDATV</td>
<td>Data files for TEP support</td>
</tr>
<tr>
<td>ITM.TKANPAR</td>
<td>Parameter files for TEP support</td>
</tr>
</tbody>
</table>

Table 6 shows a list of the USS directories that are provided by the SMP/E installation process.

Table 6. USS Paths

<table>
<thead>
<tr>
<th>USS Path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/usr/lpp/ing/adapter</td>
<td>Shell script</td>
</tr>
<tr>
<td>/usr/lpp/ing/adapter/lib</td>
<td>Executable</td>
</tr>
<tr>
<td>/usr/lpp/ing/adapter/config</td>
<td>Configuration file</td>
</tr>
<tr>
<td>/usr/lpp/ing/adapter/data</td>
<td>Customer data/empty at installation</td>
</tr>
<tr>
<td>/usr/lpp/ing/adapter/ssl</td>
<td>Customer data/empty at installation</td>
</tr>
<tr>
<td>/usr/lpp/ing/ussauto</td>
<td>USS automation executable file</td>
</tr>
<tr>
<td>/usr/lpp/ing/ussauto/lib</td>
<td>SA z/OS-related documentation</td>
</tr>
<tr>
<td>/usr/lpp/ing/doc</td>
<td>Best practice policy diagrams</td>
</tr>
<tr>
<td>/usr/lpp/ing/dist</td>
<td>For distributed connectors</td>
</tr>
<tr>
<td>/usr/lpp/ing/dist/tec</td>
<td>Tivoli Enterprise Console (TEC) related code</td>
</tr>
<tr>
<td>/usr/lpp/ing/dist/tdi</td>
<td>Tivoli Directory Integrator (TDI) related code</td>
</tr>
<tr>
<td>/usr/lpp/ing/dist/omnibus</td>
<td>Tivoli Netcool/OMNIbus-related code</td>
</tr>
<tr>
<td>/usr/lpp/ing/sap</td>
<td>SAP-related code</td>
</tr>
</tbody>
</table>

The following list helps you to grant RACF access to the appropriate users of the data sets:
Step 1: SMP/E Installation

1. Data sets of this category are related to ISPF and need to be accessed by everyone that uses the customization dialog.
2. Data sets of this category need to be accessed by the system programmer running SMP/E.
3. Data sets of this category need to be used by the NetView and automation team responsible for setting up and customizing system automation and I/O operations.
4. Data sets of this category need to be accessed by everyone that uses the SA TSO REXX environment.
5. Data sets of this category need to be accessed by anyone who will be installing the NMC component.
6. Data sets of this category are only required if you install Kanji support.
7. Data sets of this category are defined in VM setup.
8. Data sets of this category are required for the Automated Discovery function.
9. These data sets are required for Tivoli Enterprise Portal support, where &shilev is the high-level qualifier of the SMP/E target libraries used. See also IBM Tivoli System Automation for z/OS Monitoring Agent Configuration and User's Guide.
10. Files in these directories are used for USS Automation and the end-to-end automation adapter.
11. Files in the folder and subfolders of the ./doc directory under SA z/OS documentation.
12. Files in these directories are used to integrate with other products.
Step 1: SMP/E Installation
Chapter 9. Base SA z/OS Configuration Using the Configuration Assistant

The configuration of this product is supported by the Configuration Assistant.

Instead of manually adapting configuration jobs, start procedures, and initialization files to your environment, this assistant generates these files for you. The settings that are implemented are taken from the user-customized INGDOPT Configuration Options file.

The generated files are created as members within a dynamically allocated configuration data set (CONFLIB). In this data set, they are populated with the values that you define in the INGDOPT Configuration Options file.

The CONFLIB data set contains these items:
- Jobs to allocate all data sets that are required by SA z/OS during run time
- Procedures to start the components of SA z/OS to be copied to your target SYS1.PROCLIB
- Runtime configuration members for both Automation Manager and Automation Agent
- Parameter files that are ready to be copied to your target SYS1.PARMLIB
- VTAM definitions that are files ready to be copied to your target VTAMLST
- Jobs to delete data set files in case you must reconfigure or delete SA z/OS again
- A job to verify the success of the installation and configuration process.

All members within the CONFLIB data set can be inspected, if required. If you applied changes to the generated members, be aware that the CONFLIB data set is newly allocated when running the configuration assistant another time.

Note: The security administrator must give the system programmer and the automation programmer ALTER access to the HLQ for the locally allocated and active automation policy data sets. The security administrator must also authorize the user ID used by the SA z/OS started tasks for accessing the data sets as follows:
- READ for SMP/E and active automation policy
- UPDATE for the locally allocated data sets

Table 7 serves as a reference to manual SA z/OS configuration steps as documented below in Chapter 10, “Traditional SA z/OS Configuration,” on page 73. All steps are marked which are covered by the configuration assistant.

<table>
<thead>
<tr>
<th>Task</th>
<th>SysOps</th>
<th>ProcOps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1: SMP/E Installation. Refer to [Chapter 8, “SMP/E Installation,” on page 51]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 2: Allocate System-Unique Data Sets</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Step 3: Allocate Data Sets for the ISPF Dialog</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>
### Base z/OS Configuration using the Configuration Assistant

<table>
<thead>
<tr>
<th>Task</th>
<th>SysOps</th>
<th>ProcOps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 4: Configure SYS1.PARMLIB members</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Step 5: Configure SYS1.PROCLIB members</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Step 6: Configure NetView</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Step 7: Preparing the Hardware</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 8: Preparing Ensemble HMC Communication</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 9: Preparing VM PSM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 10: Configure the Automation Manager</td>
<td>✔</td>
<td>n/a</td>
</tr>
<tr>
<td>Step 11: Configure the Component Trace</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Step 12: Configure the System Logger</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Step 13: Configure ISPF Dialog Panels</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Step 14: Verify the Number of available REXX Environments</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 15: Configure Function Packages for TSO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 16: Configure Alert Notification for SA z/OS</td>
<td>✔(^1)</td>
<td></td>
</tr>
<tr>
<td>Step 17: Compile SA z/OS REXX Procedures</td>
<td>✔(^2)</td>
<td></td>
</tr>
<tr>
<td>Step 18: Define Automation Policy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 19: Define Host-To-Host Communication</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Step 20: Enabling SA z/OS to Restart Automatic Restart Manager (ARM) Enabled Systems</td>
<td>✔(^2)</td>
<td>✔</td>
</tr>
<tr>
<td>Step 21: Define Security</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Step 22: Configure the Status Display Facility (SDF)</td>
<td>✔(^2)</td>
<td>✔</td>
</tr>
<tr>
<td>Step 23: Check for Required IPL</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Step 24: Automatic system Operations Startup</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>....</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. Alert Notification through the Tivoli Event Integration Facility (EIF)
2. SDF configured for the local system

**Note:** I/O-Ops configuration is not supported by the Configuration Assistant.

---

### Preparing to Configure IBM System Automation for z/OS

Preparation consists of the following steps:

1. Allocate a data set where you can maintain working copies of the INGDOPT Configuration Options file and the Configuration Assistant job. See “Allocate a data set for work files” on page 57.
2. Create a work copy of the INGDOPT Configuration Options file and the Configuration Assistant sample job (INGDCONF). See “Create Work Copies” on page 57.
3. Edit the working copy of the INGDOPT Configuration Options file to reflect the parameters of your installation. These parameters are then used to build the necessary artifacts to complete the configuration. See “Editing the Work Copy of the INGDOPT Configuration Options File” on page 58.
4. Edit and submit the work copy of the INGDCONF sample job. This job allocates the CONFLIB data set and configures the rest of the configuration jobs. See "Editing and Submitting the Work Copy of the INGDCONF Configuration Assistant Job" on page 58.

5. Follow the instructions documented in CONFLIB in $INGREAD.

Note: The user ID under which these jobs are submitted must be authorized to read the SMP/E target libraries. Runtime-specific data sets are allocated with a high-level qualifier as is specified in the INGDOPT Configuration Options file. The user must have ALTER access to create these data sets.

Allocate a data set for work files

Allocate a data set where you can maintain working copies of the INGDOPT Configuration Options file and the Configuration Assistant job. The working copies are specific for each z/OS environment controlled by SA z/OS, also referred to as 'SAplex'.

Compose the name of that library out of a high-level qualifier (HLQ), the SAplex name (SAplex) both of your choice and the low-level qualifier (LLQ) named CONFWRK. This naming scheme must not be changed because it is used by the Configuration Assistant job. For example, if you decide to use the HLQ of 'USER' and you configure SA z/OS on z/OS systems belonging to a sysplex named SYSPLEX1, the recommended name for the work data set is USER.SYSPLEX1.CONFWRK.

The length of the data set name must not exceed 35 characters because the following data sets are allocated by other JCLs later on:

- hlq.saplex.CONFLIB.&SYSNAME.
- hlq.saplex.CONFLIB.VTAMLIB

&SYSNAME. represents a system symbol which is resolved when running these JCLs on the individual systems.

The characteristics for the data set (PDS or PDSE) are as follows:

- RECFM=FB, LRECL=80

As an initial size for the CONFWRK data set, you might allocate the following number of tracks:

- Primary Quantity . . 15
- Secondary Quantity . . 5
- Directory Blocks . . . 5
- Block Size . . . . . . 27920

Create Work Copies

The INGDOPT Configuration Options file and the INGDCONF Configuration Assistant are supplied as members in the sample data set that is part of SMP/E DDDEF name SINGSAMP.

Create a work copy of the INGDOPT and INGDCONF members in the work data set, which you allocated in the previous step. Do not change the members in the data set that belongs to SMP/E DDDEF SINGSAMP.
Editing the Work Copy of the INGDOPT Configuration Options File

You define various settings that vary from installation to installation in the INGDOPT Configuration Options file. Typical examples are data set high-level qualifiers, system name, and the NetView domain name. These settings are used to build the configuration files in the CONFLIB data set.

Next, edit the INGDOPT Configuration Options file according to the syntax rules and the documentation that you find within that file.

The INGDOPT Configuration Options file contains comprehensive documentation on the purpose of the parameters.

Editing and Submitting the Work Copy of the INGDCONF Configuration Assistant Job

This job runs the Configuration Assistant and allocates the CONFLIB partitioned data set.

The data set stores the generated JCLs, start procedures, parmlib members, and other initialization and configuration members. Follow the instructions that are given in the INGDCONF job to adapt the job statements and the JCL variables within your INGDCONF work copy. When finished, submit the job.

Follow the Instructions as Documented in $INGREAD

Documentation member $INGREAD was tailored to your installation and created in the CONFLIB data set.

Follow the instructions documented there and complete the basic configuration. When you are finished with $INGREAD, proceed with the configuration described in these sections.

Completing Member Configuration

Configure the System Logger (optional)

Configuring the System Logger allows gathering resource-related history data. Even though this configuration is not mandatory for resource automation, it is recommended for problem determination tasks.

This step must be performed on the target system, where SA z/OS is to be configured. See the configuration step "Step 12: Configure the System Logger" on page 109 in "Traditional SA z/OS Configuration" in IBM Tivoli System Automation for z/OS Planning and Installation Guide.

Note: If the system logger is not configured, the INGDVRFY verification job issues a warning message. Ignore that message if you do not want to configure the system logger for automation.

Update SMFPRMxx (optional)

If you plan to use SMF records for the availability reporting of automated resources, you must update the SMFPRMxx member.
Preparing to Configure IBM System Automation for z/OS

This step must be performed on the target system, where SA z/OS is to be configured. See the configuration step “Step 4I: Update SMFPRMxx” on page 85 in the “Traditional SA z/OS Configuration” section.

Install the TSO REXX Function Package (optional)

The function package is used for the following functions:
- Batch interface (see also member EVJSJ001 in *.SINGSAMP library)
- Relational Data Services (RDS)
- Syntax checking for automation table overrides
- Preloader function of the Automated Discovery feature

If you plan to use these functions, you must configure the TSO REXX Function Package on the target system where SA z/OS is to be installed.

See the configuration step “Step 15: Configure Function Packages for TSO” on page 117 in Chapter 10, “Traditional SA z/OS Configuration,” on page 73.

Configuration of Alert Notification for IBM SA z/OS (optional)

SA z/OS provides an alert-based notification service that alerts subject matter experts. You can escalate automation problems that require manual intervention by sending alerts, events, or trouble tickets to different kinds of notification targets.

For more information, see “Alert-Based Notification” in IBM Tivoli System Automation for z/OS Customizing and Programming Guide.

For configuration instructions, see the step “Step 12: Configure the System Logger” on page 109 in “Traditional SA z/OS Configuration” in IBM Tivoli System Automation for z/OS Planning and Installation Guide.

Verifying Your Configuration

Submit the INGDVRFY Configuration Verification job on the target system where SA z/OS was configured.

This job is in the CONFLIB library. After the job terminates, investigate the job log for INGVxxxx messages. If required, correct the configuration according to those messages.

Start SA z/OS for the First Time

Before you proceed to details about the contents of the automation policy and techniques in the Customization Dialog for resource definitions, use this section to get a jump-start with a correct Policy Database (PDB) for a plain z/OS system.

You can use the procedure to complete the initial configuration as explained previously. This procedure is expected to take less than 30 minutes.

After you validate your configuration and you have a basic policy, then you can skip the section.
Quick Planning Exercise

Quick planning exercise

The created basic policy contains a number of standard applications (started tasks) on z/OS systems. The started tasks must match the naming standards that are in place on the target system.

The following planning sheet guides you in identifying the real job names that are used in the PDB and ensures that the applications are named correctly.

Table 8. Worksheet for job names

<table>
<thead>
<tr>
<th>Application</th>
<th>Description</th>
<th>Default Job Name</th>
<th>Real Job Name</th>
<th>Default Procedure Name</th>
<th>Real Procedure Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM</td>
<td>Automation Manager</td>
<td>AM</td>
<td>AM</td>
<td>INGEAMSA</td>
<td>See Note 1</td>
</tr>
<tr>
<td>AM2</td>
<td>Spare Automation Manager</td>
<td>AM2</td>
<td>AM2</td>
<td>INGEAMSA</td>
<td>See Note 2</td>
</tr>
<tr>
<td>APPC</td>
<td>Advanced Peer-to-Peer Communication</td>
<td>APPC</td>
<td>APPC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASCH</td>
<td>APPC Scheduler</td>
<td>ASCH</td>
<td>ASCH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLSJPRMI</td>
<td>Build SNAP Tables for IPCS</td>
<td>BLSJPRMI</td>
<td>BLSJPRMI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLF</td>
<td>Data Lookaside Facility</td>
<td>DLF</td>
<td>DLF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FFST™</td>
<td>First Failure Support Technology™</td>
<td>FFST</td>
<td>FFST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HSM</td>
<td>Hierarchical Storage Manager</td>
<td>HSM</td>
<td>HSM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IRRDPTAB</td>
<td>RACF dynamic parse table loader</td>
<td>IRRDPTAB</td>
<td>IRRDPTAB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JES2</td>
<td>Job Entry Subsystem 2</td>
<td>JES</td>
<td>JES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LLA</td>
<td>Library Lookaside</td>
<td>LLA</td>
<td>LLA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OAM</td>
<td>Object Access Method</td>
<td>OAM</td>
<td>OAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OMPROUTE</td>
<td>Open MVS MultiProtocol Routing Daemon</td>
<td>OMPROUTE</td>
<td>OMPROUTE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OMVS</td>
<td>UNIX System Services subsystem</td>
<td>OMVS</td>
<td>OMVS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RACF</td>
<td>Resource Access Control Facility</td>
<td>RACF</td>
<td>RACF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RESOLVER</td>
<td>TCP/IP Name Resolver</td>
<td>RESOLVER</td>
<td>RESOLVER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMF</td>
<td>Resource Measurement Facility™</td>
<td>RMF</td>
<td>RMF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMFGAT</td>
<td>RMF Monitor III Data Gatherer</td>
<td>RMFGAT</td>
<td>RMFGAT</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 8. Worksheet for job names (continued)

<table>
<thead>
<tr>
<th>Application</th>
<th>Description</th>
<th>Default Job Name</th>
<th>Real Job Name</th>
<th>Default Procedure Name</th>
<th>Real Procedure Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>RRS</td>
<td>Resource Recovery Services</td>
<td>RRS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSVAPPL</td>
<td>Automation Application</td>
<td>&amp;JOBNAME</td>
<td>INGENVSA</td>
<td>See Note 3</td>
<td></td>
</tr>
<tr>
<td>SYSVIPLC</td>
<td>IPL Data Gatherer</td>
<td>SYSVIPLC</td>
<td>HSAPIPLC</td>
<td>See Note 4</td>
<td></td>
</tr>
<tr>
<td>SYSVSSI</td>
<td>Automation Subsystem Interface</td>
<td>SYSVSSI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCPIP</td>
<td>TCP/IP</td>
<td>TCPIP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSO</td>
<td>Time Sharing Option</td>
<td>TSO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VLF</td>
<td>Virtual Lookaside Facility</td>
<td>VLF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VTAM</td>
<td>Virtual Telecommunication Access Method</td>
<td>VTAM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZFS</td>
<td>z/OS File System</td>
<td>ZFS</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notes:
1. When you specified sa_am_start_proc in the Options File, use this value, otherwise use what is specified for sa_am_start_job.
2. When you specified sa_am_start_proc in the Options File, use this value, otherwise use what is specified for sa_am_start_job.
3. When you specified sa_saagent_start_proc in the Options File, use this value, otherwise use what is specified for sa_saagent_start_job.
4. When you specified sa_ipldata_start_proc in the Options File, use this value, otherwise use what is specified for sa_ipldata_start_job.

In all likelihood, most of the listed applications are not changed because most installations already use the default names. However, for some applications, different job names might be used and therefore the job name attribute for such applications has to be adopted in the basic policy. Also, some of the applications might not exist on the target system, so those applications can be deleted or unlinked from the basic policy. Take note of those applications that require a job name change or that can be deleted.

**Starting the Customization Dialog**

The Configuration Assistant provided you with a REXX script called INGEDLG.

1. Copy this script into a data set in your SYSPROC or SYSEXEC concatenation.
2. Start it as follows:
   - `%INGEDLG`
   - Alternatively, start it directly out of the CONFLIB with the TSO EXEC command. For example: `TSO EXEC 'MYHLQ.SYSA.CONFLIB(INGEDLG)'` After INGEDLG is started, you see a panel as follows:
Starting the Customization Dialog

Creating a basic PDB

The first thing that you must do is to create a Policy database.

1. From the System Automation for z/OS 3.5 Customization Dialog panel, enter ? in the Current Policy Database field at the bottom of the page and press Enter.

You see a panel as follows:

2. To create a PDB, type the word new on the command line and press Enter.

You now see a panel as follows:
To define a new Policy Database, specify the following information:

- Policy Database Name...
- Enterprise Name...
- Data Set Name...

Model Policy Database. *EMPTY Policy Database name or "?" for list of names

Add-on policies to be added to a standard SA model policy database:

<table>
<thead>
<tr>
<th>Action</th>
<th>Status</th>
<th>Add-on Policy</th>
<th>Customizable</th>
</tr>
</thead>
<tbody>
<tr>
<td>______</td>
<td>______</td>
<td>+BASE</td>
<td>YES</td>
</tr>
<tr>
<td>______</td>
<td>______</td>
<td>+CICS</td>
<td></td>
</tr>
<tr>
<td>______</td>
<td>______</td>
<td>+DB2</td>
<td></td>
</tr>
<tr>
<td>______</td>
<td>______</td>
<td>+E2E</td>
<td>YES</td>
</tr>
<tr>
<td>______</td>
<td>______</td>
<td>+GDPS</td>
<td></td>
</tr>
<tr>
<td>______</td>
<td>______</td>
<td>+IBMCOMP</td>
<td>YES</td>
</tr>
<tr>
<td>______</td>
<td>______</td>
<td>+IBM</td>
<td></td>
</tr>
<tr>
<td>______</td>
<td>______</td>
<td>+ITM</td>
<td>YES</td>
</tr>
<tr>
<td>______</td>
<td>______</td>
<td>+NMC</td>
<td></td>
</tr>
<tr>
<td>______</td>
<td>______</td>
<td>+PROCOPS</td>
<td></td>
</tr>
<tr>
<td>______</td>
<td>______</td>
<td>+SAPSRV</td>
<td></td>
</tr>
<tr>
<td>______</td>
<td>______</td>
<td>+TBSM</td>
<td></td>
</tr>
<tr>
<td>______</td>
<td>______</td>
<td>+TWS</td>
<td>YES</td>
</tr>
<tr>
<td>______</td>
<td>______</td>
<td>+USS</td>
<td></td>
</tr>
<tr>
<td>______</td>
<td>______</td>
<td>+WEBSPHERE</td>
<td>YES</td>
</tr>
</tbody>
</table>

*********** Bottom of data ***********

3. The **Policy Database Name** field enters the name within the Customization Dialog. The value that is entered here must be a single word but can include underscores. **TEST_PDB** is recommended.

4. The **Enterprise Name** field enters the name of your business or the section of it that you are going to define in the PDB. The value that you enter here must be a single word but may include underscores. **TEST_SYSTEMS** is recommended.

5. The **Data Set Name** field enters the name of the data set on disk that holds the policy database. A useful convention is to have the name end with a .PDB extension, and to use the same name with a .SOCNTL extension for the Automation Control File that gets built from it. If you enter a value without single quotation marks, it is taken to be relative to your TSO user ID. If you enter a value with single quotation marks, it is taken as an absolute. For example, **TEST.PDB** might result in data set **USER.TEST.PDB**, while ‘AUTO.TEST.PDB’ results in a data set ‘AUTO.TEST.PDB’. Use what you specified for **sa_automation_policy** in the **INGDOPT Configuration Assistant Options file** and put single quotation marks around it. The section at the bottom with the add-on policies adds the sample policies to your empty policy database.

6. Enter C in front of +BASE and press Enter.
Creating a basic PDB

For the basic PDB, only the Base z/OS components and one of the JES subsystems are required.

7. To deselect that component, which is not required, specify M in front and press Enter. The SELECTED status is now only shown for Base z/OS and either for JES2 or for JES3. It depends what type of JES that you use on the target system.

8. When finished, press PF3.

9. Press Enter to review the contents of the New Policy Database Dataset Information panel and press Enter once more to create the policy.

After a few messages (you must press Enter to clear them), you find yourself on the Entry Type Selection panel for your new policy database:

Adapting the System Name

You now have a basic PDB that is built from the sample add-on policy that is provided by the product. In this policy, the default systems that are being automated are called SYS1, SYS2, and SYS3. These names have to be changed to match the names of your systems.

1. Select 4 on the Option line and you see the systems that are listed as shown here:
Adapting the System Name

---
Entry Name Selection Row 1 from 3
---
Command ==> __________________________________________________ SCROLL==> CSR

Entry Type : System
PolicyDB Name : TEST_PDB
Enterprise Name : TEST_SYSTEMS

<table>
<thead>
<tr>
<th>Action</th>
<th>Entry Name</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_______</td>
<td>SYS1</td>
<td>System 1 of the SA Sample Sysplex</td>
</tr>
<tr>
<td>_______</td>
<td>SYS2</td>
<td>System 2 of the SA Sample Sysplex</td>
</tr>
<tr>
<td>_______</td>
<td>SYS3</td>
<td>System 3 of the SA Sample Sysplex</td>
</tr>
</tbody>
</table>

---

2. To rename the policy entry name of the system SYS1, enter r and press Enter. In the pop-up panel that is displayed next, enter the name of your system and press Enter again. The entry name is renamed, but one more renaming action is necessary.

3. Enter SI and press Enter. This action leads you to the SYSTEM INFORMATION policy. Here again, you have to change the field Image/System name to match the name of your system. Before the change, the panel might look like as follows:

---
System Information
---
Command ==> ___________________________________________________________________

Entry Type : System
PolicyDB Name : TEST_PDB
Entry Name : SYS1
Enterprise Name : TEST_SYSTEMS

Operating system : MVS
Image/System name : SYS1

The following specifications are for MVS systems only:
Primary JES : JES2
System monitor time : 00:59
Gateway monitor time : 00:15
Automation table(s) : TNGMTSG01

4. Rename SYS1 here to your system name and press PF3 to leave the dialog box. You now see a group of messages that flow through the panel that shows the resources that are defined for your system. The message flow reflects the contents of the basic policy.

5. As a starting point, it is sufficient to automate just a single system. So you may leave SYS2 and SYS3 untouched and add further systems later on after the first system can be automated.

6. Press PF3 twice to return to the Entry Type Selection panel.

Adapting Application Job Names

Use the notes that you took during the planning exercise to change the default job names (where necessary) to the real job names. Then, either delete or unlink those applications that are not used on the target system.

Select 6 on the Option line and press Enter. The Entry Name Selection panel for entry type Application is displayed.
Adapt Application Job Names

To change a job name for an application, enter AI next to that application and press Enter. A panel is shown as follows:

For example, if the VTAM job name is NET on the target system, change the value of the Job Name field in the panel appropriately. If you press PF3 twice, you return to the Entry Name Selection panel.

Follow the same steps if you must enter the JCL Procedure Name.

To delete an application you do not need, enter D next to it and press Enter. You see a confirmation panel and press Enter again. However, if you want to use this
adapt application job names

application in the future, unlink it. The definitions are kept in the policy but the
Customization Dialog does not create a resource for the application. You can link
such an application any time later again.

To unlink an application you do not need, enter $W$ next to it and press Enter. You
notice that the application is linked to a group called BASE_SYS. Enter $M$ (for
reMove) next to it and press PF3.

Changing System Defaults

When you create the basic PDB the first time, you have no experience yet with
customization and operations of the product. It is recommended to monitor what
is going on to further familiarize yourself with the product, and then switch on
automation. The approach protects you from stumbling into pitfalls where
unintended automation might happen by accident.

To do so, you can switch automation globally off by setting the Automation flag to
LOG in the System Defaults (SDF). No automation takes place but the commands
that the automation would run are shown in the netlog.

1. Select 35 on the Option line and press Enter. You see a single system
SYSTEM_DEFAULTS similar to what is shown here:

<table>
<thead>
<tr>
<th>Command</th>
<th>Entry Name Selection</th>
<th>Row 1 from 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>Entry Type: System Defaults</td>
<td>PolicyDB Name: TEST_PDB</td>
</tr>
<tr>
<td></td>
<td>Enterprise Name: TEST.ENTERPRISE</td>
<td></td>
</tr>
<tr>
<td>Action</td>
<td>Entry Name: SYSTEM.DEFAULTS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Short Description: System Defaults</td>
<td></td>
</tr>
</tbody>
</table>

2. Under Action, specify AF and press Enter. You enter the Automation Flag
Processing dialog that is shown here:

<table>
<thead>
<tr>
<th>Command</th>
<th>Automation Flag Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>Entry Type: System Defaults</td>
</tr>
<tr>
<td></td>
<td>PolicyDB Name: TEST_PDB</td>
</tr>
<tr>
<td></td>
<td>Enterprise Name: TEST.ENTERPRISE</td>
</tr>
<tr>
<td>Resource</td>
<td>System Defaults</td>
</tr>
<tr>
<td>Line Commands: Exi (Exits), Dis (Disable Times)</td>
<td></td>
</tr>
<tr>
<td>Automation Level: YES, NO, LOG, EXITS</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cmd Flag</th>
<th>Auto</th>
<th>Exits</th>
<th>DisableTimes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automation (A)</td>
<td>LOG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initstart (I)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start (S)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Recovery (R)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Terminate (T)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Restart (RS)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3. Change the value of Automation from YES to LOG and press PF3. No
automation happened accidentally. But do not forget to turn the flag back to
YES after you are familiar with the product.

4. Press PF3 again until you are back on the initial panel, the primary panel, of
the Customization Dialog.
Building the Configuration Files

You completed the steps to create a basic automation policy. You now create the configuration files (SOCNTL).

1. Enter option 2 from the System Automation for z/OS 3.5 Customization Dialog to start the Build dialog.

<table>
<thead>
<tr>
<th>Option</th>
<th>Build Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Build a complete enterprise</td>
</tr>
<tr>
<td>2</td>
<td>Build sysplex group or stand alone system</td>
</tr>
<tr>
<td>3</td>
<td>Build entry type or entry name</td>
</tr>
<tr>
<td>4</td>
<td>View build report</td>
</tr>
</tbody>
</table>

Build options:
- Output Data Set _____________________________
- Mode ........ ONLINE (ONLINE BATCH)
- Type .......... MODIFIED (MODIFIED ALL)
- Configuration .... NORMAL (NORMAL ALTERNATE)

Job statement information: (used for BATCH build)
//AOFBUILD JOB
//*

2. The Configuration Assistant already created a SOCNTL file for you. So, in the Output Data Set field, enter the value that you specified for sa_automation_policy in the Configuration Options file and append '.SOCNTL', surrounded by single quotation marks. For example: 'USER, POLICY, NAME, PDB, SOCNTL'.

3. Change Type from MODIFIED to ALL.

4. Select Option 1 **Build a complete enterprise** and press Enter. Messages are displayed and after a time, the build process completes successfully.

You created a SOCNTL file from your basic policy that can be used to load on the target system. For the remaining steps, you need a console to enter system commands on the target system.

Starting the Automation Manager

The Automation Manager is started with a standard MVS Start command:

Issue: S INGEMSA,JOBNAME=AM,TYPE=COLD,SUB=MSTR

**Note:** If you specified a different JCL procedure name (sa_am_start_proc) or job name (sa_am_start_job.1) for the automation manager, then use the values specified in the INGOOPT Configuration Options file.

The Automation Manager initializes and issues the following message when the initialization is complete:

HSAM1308I SA z/OS PRIMARY AUTOMATION MANAGER INITIALIZATION COMPLETE, TYPE=COLD

If this message is not displayed, see which of these actions can help you:
1. Find the messages that are displayed on the MVS console to identify the cause
2. Verify that you have the proper authority
3. Be sure that you performed correctly all steps of the configuration that are described in the chapter

Starting the Subsystem Interface Task

The Subsystem Interface Task is started with a standard MVS Start command:

Issue: $ CNMSJO10,JOBNAME=SYSVSSI,SUB=MSTR

Note: If you specified a different JCL procedure name (sa_nvssi_start_proc) or job name (sa_nvssi_start_job) for the subsystem interface task, then use the values as specified in the INGDOPT Configuration Options file.

After the task is initialized, the following message appears:

CNM541I NetView subsystem SYSV is fully functional

If this message is not displayed, see which of these actions can help you:
1. Find the messages that are displayed on the MVS console to identify the cause
2. Verify that you have the proper authority
3. Be sure that you performed correctly all steps of the configuration that are described in the chapter

Starting the Automation Agent

The Automation Agent is started with a standard MVS Start command:

Issue: $ INGENVSA,JOBNAME=SYSVAPPL,SUB=MSTR

Note: If you specified a different JCL procedure name or job name for the automation agent, then use the values (sa_saagent_start_proc or sa_saagent_start_job, respectively) as found in the INGDOPT Configuration Options file.

After the Automation Agent is initialized up to the point where logging on is possible, it responds with the following message:

*002 DS1802A ING01 REPLY WITH VALID NCCF SYSTEM OPERATOR COMMAND

If this message is not displayed, see which of these actions can help you:
1. Find the messages that are displayed on the MVS console to identify the cause
2. Verify that you have the proper authority
3. Be sure that you performed correctly all steps of the configuration that are described in the chapter

After this message is displayed, you are able to log on to the NetView 3270 console.

The Automation Manager instructs the Automation Agent to load the SOCNTL data set. When done, another message is displayed:
Starting the Automation Agent

Press Enter to close this message.

Verification

When the Automation Manager and the Automation Agent are both started successfully, log on to the NetView console.

1. To log on, enter LOGON APPLID (domain). For domain, use the value that you specified for net_netview_domain_id in the INGDOPT Configuration Options file. A panel is shown as follows:

   Enter logon information or PF3/PF15 to logoff

2. For OPERATOR ID, specify OPER1. For the PASSWORD, specify OPER1. The entries are default credentials that are set up for you to get into SA z/OS initially.

   Note: Secure the environment as soon as possible following the guidelines in Chapter 11, “Security and Authorization,” on page 161.

3. After you log on, press Enter, when you see message: =X= *** DSI662I SCREEN HELD.

4. Enter INGAMS on the command line for the operational command INGAMS. A panel as follows is then displayed:
The statuses of the Primary Automation Manager (PAM) and of the Automation Agent are READY.
Verification
Chapter 10. Traditional SA z/OS Configuration

<table>
<thead>
<tr>
<th>Overview of Configuration Tasks</th>
<th>75</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2: Allocate System-Unique Data Sets</td>
<td>77</td>
</tr>
<tr>
<td>Step 2A: Data Sets for NetView</td>
<td>77</td>
</tr>
<tr>
<td>Step 2B: Data Sets for I/O Operations</td>
<td>78</td>
</tr>
<tr>
<td>Step 2C: Data Sets for Automation Agents (Primary Automation Manager and Backups)</td>
<td>79</td>
</tr>
<tr>
<td>Step 2E: SA z/OS Password Store Data Set</td>
<td>80</td>
</tr>
<tr>
<td>Step 3: Allocate Data Sets for the ISPF Dialog</td>
<td>81</td>
</tr>
<tr>
<td>Step 4: Configure SYS1.PARMLIB Members</td>
<td>82</td>
</tr>
<tr>
<td>Step 4A: Update IEAAPFxx</td>
<td>82</td>
</tr>
<tr>
<td>Step 4B: Update SCHEDxx</td>
<td>82</td>
</tr>
<tr>
<td>Step 4C: Update MPFLSTxx</td>
<td>83</td>
</tr>
<tr>
<td>Step 4D: Update LPALSTxx</td>
<td>83</td>
</tr>
<tr>
<td>Step 4E: Update LNKLSTxx</td>
<td>84</td>
</tr>
<tr>
<td>Step 4F: Update BPXPRMxx</td>
<td>84</td>
</tr>
<tr>
<td>Step 4G: Update IEESNxx</td>
<td>84</td>
</tr>
<tr>
<td>Step 4H: Update JES3INxx</td>
<td>85</td>
</tr>
<tr>
<td>Step 4I: Update LNKLSTxx</td>
<td>85</td>
</tr>
<tr>
<td>Step 5: Configure SYS1.PROCLIB Members</td>
<td>86</td>
</tr>
<tr>
<td>Step 5A: NetView Startup Procedures</td>
<td>86</td>
</tr>
<tr>
<td>Step 5B: Startup Procedures Required for System Operations Only</td>
<td>86</td>
</tr>
<tr>
<td>Step 6: Configure NetView</td>
<td>88</td>
</tr>
<tr>
<td>Step 6A: Configure NetView Alert Information</td>
<td>88</td>
</tr>
<tr>
<td>Step 6B: Configure NetView DSIPARM Data Set</td>
<td>89</td>
</tr>
<tr>
<td>Step 6C: Modifying NetView DSIPARM Definitions for an Automation Network</td>
<td>94</td>
</tr>
<tr>
<td>Step 6D: Configure NetView for Processor Operations</td>
<td>94</td>
</tr>
<tr>
<td>Step 6E: Configure the NetView Message Translation Table</td>
<td>95</td>
</tr>
<tr>
<td>Step 6F: Add the REXX Function Packages to DSIRXPRM</td>
<td>95</td>
</tr>
<tr>
<td>Step 7: Preparing the Hardware</td>
<td>95</td>
</tr>
<tr>
<td>Step 7A: Preparing the HMC (Console Workplace 2.9 and Later Versions)</td>
<td>96</td>
</tr>
<tr>
<td>Enable the HMC API, Set the SNMP Community Names and the SNMPv3 Information</td>
<td>96</td>
</tr>
<tr>
<td>BCP Internal Interface</td>
<td>97</td>
</tr>
<tr>
<td>CPC Object Definitions on the HMC</td>
<td>98</td>
</tr>
<tr>
<td>Step 7B: Preparing the SE (Console Workplace 2.9 and Later Versions)</td>
<td>98</td>
</tr>
<tr>
<td>Enable the SE API, Set the Community Name and the SNMP Information</td>
<td>98</td>
</tr>
<tr>
<td>Step 7C: Preparing the SE (Console Workplace 2.10 and Later Versions)</td>
<td>100</td>
</tr>
<tr>
<td>Enabling Capacity Change API Requests</td>
<td>100</td>
</tr>
<tr>
<td>Step 7D: Updating Firewall Information</td>
<td>101</td>
</tr>
<tr>
<td>Connection protocol SNMP</td>
<td>101</td>
</tr>
<tr>
<td>Step 8: Preparing Ensemble HMC Communication</td>
<td>101</td>
</tr>
<tr>
<td>Step 8A: Setting up the Ensemble Hardware Management Console for use with System Automation for z/OS</td>
<td>101</td>
</tr>
<tr>
<td>Step 8B: Setting up AT-TLS for the SSL socket connection</td>
<td>101</td>
</tr>
<tr>
<td>Step 9: Preparing the VM PSM</td>
<td>103</td>
</tr>
<tr>
<td>Installing the PSM Code on VM</td>
<td>103</td>
</tr>
<tr>
<td>Configuration</td>
<td>104</td>
</tr>
<tr>
<td>Customizing the PSM</td>
<td>104</td>
</tr>
<tr>
<td>ISQADDRS DATA</td>
<td>105</td>
</tr>
<tr>
<td>ISQPARM DATA</td>
<td>105</td>
</tr>
<tr>
<td>Logger Files</td>
<td>106</td>
</tr>
<tr>
<td>Step 10: Configure the Automation Manager</td>
<td>106</td>
</tr>
<tr>
<td>Step 10A: XCF Characteristics</td>
<td>107</td>
</tr>
<tr>
<td>Step 10B: Configuring HSAPRMxx</td>
<td>107</td>
</tr>
<tr>
<td>Step 10C: ARM Instrumentation of the Automation Manager</td>
<td>107</td>
</tr>
<tr>
<td>Step 10D: Security Considerations</td>
<td>108</td>
</tr>
<tr>
<td>Step 11: Configure the Component Trace</td>
<td>108</td>
</tr>
<tr>
<td>Step 12: Configure the System Logger</td>
<td>109</td>
</tr>
<tr>
<td>Step 13: Configure ISPF Dialog Panels</td>
<td>111</td>
</tr>
<tr>
<td>Step 13A: Allocate Libraries for the Dialogs</td>
<td>111</td>
</tr>
<tr>
<td>Alternative 1: Dynamic Allocation using INGDLG</td>
<td>112</td>
</tr>
<tr>
<td>Alternative 2: Allocation of the libraries as part of the TSO Logon Procedure</td>
<td>112</td>
</tr>
<tr>
<td>Step 13B: Logging Modifications to Data Set</td>
<td>114</td>
</tr>
<tr>
<td>Step 13C: Invoking the ISPF Dialogs</td>
<td>114</td>
</tr>
<tr>
<td>Using INGDLG</td>
<td>115</td>
</tr>
<tr>
<td>Using TSO Logon or Your own Automation Procedure</td>
<td>115</td>
</tr>
<tr>
<td>Step 13D: Reconvert I/O Operations Panels</td>
<td>116</td>
</tr>
<tr>
<td>Step 13E: Verify the ISPF Dialog Installation</td>
<td>116</td>
</tr>
<tr>
<td>Step 14: Verify the Number of available REXX Environments</td>
<td>116</td>
</tr>
<tr>
<td>Step 15: Configure Function Packages for TSO</td>
<td>117</td>
</tr>
<tr>
<td>Step 15A: Installation of the TSO REXX Function Package INGTXFPG</td>
<td>117</td>
</tr>
<tr>
<td>Step 15B: Install SA Provided Authorized TSO Command INGPAUTH</td>
<td>118</td>
</tr>
<tr>
<td>Step 16: Configure Alert Notification for SA z/OS</td>
<td>118</td>
</tr>
<tr>
<td>Enabling Alert Notification via IOM Peer-To-Peer Protocol</td>
<td>119</td>
</tr>
<tr>
<td>Enabling Alert Notification via EIF Events</td>
<td>119</td>
</tr>
<tr>
<td>Starting the Event/Automation Service</td>
<td>119</td>
</tr>
<tr>
<td>Configuring the Global Initialization File</td>
<td>120</td>
</tr>
<tr>
<td>Configuring the NetView Message Adapter Service</td>
<td>120</td>
</tr>
<tr>
<td>Enabling Alert Notification via XML</td>
<td>120</td>
</tr>
<tr>
<td>Enabling Alert Notification via User-Defined Alert Handler</td>
<td>121</td>
</tr>
<tr>
<td>Step 17: Compile SA z/OS REXX Procedures</td>
<td>121</td>
</tr>
<tr>
<td>Step 18: Defining Automation Policy</td>
<td>121</td>
</tr>
<tr>
<td>Step 18A: Build the Control Files</td>
<td>122</td>
</tr>
<tr>
<td>Step 18B: Distribute System Operations Configuration Files</td>
<td>122</td>
</tr>
</tbody>
</table>
This chapter describes the tasks required to configure SA z/OS components on the SA z/OS host systems. This chapter includes information on configuring SA z/OS on both focal point and target systems. The target system configuration does not require some of the steps used for the focal point configuration. Any configuration step that does not apply to the target systems is indicated. Many of the configuration steps have corresponding planning activities and explanations in chapters 2 through 6 of this book. Chapter 12, “Configuring SA z/OS Workstation Components,” on page 197 describes installation on workstations.

In this chapter, the single installation steps are marked as either being required for all or certain SA z/OS components or as being optional. Optional denotes steps that may or may not need to be performed based on your environment, your system management procedures, and your use of the SA z/OS product. For each of these steps you need to decide whether it is required for your installation.
Each optional step explains why it is optional and describes the circumstances when you will need to perform it.

Notes:
1. The meaning of the term target system as used by SMP/E needs to be distinguished from the way the term is used in SA z/OS. As used in SMP/E and when describing the installation of z/OS products and services, a target system is the system on which a product such as SA z/OS is installed. It is the collection of program libraries that are updated during SMP/E APPLY and RESTORE processing. In this publication this meaning of target system is referred to as an “SMP/E target system”. The usual SA z/OS meaning of a “target system” is a computer system attached to a focal point system for purposes of monitoring and control.
2. In this document, data set names are shown with the high level qualifier ING. You can have a different high level qualifier for your data sets.
3. If ESCON Manager is already installed, consider that SA z/OS cannot run together with ESCON Manager on the same system. Running a mixed environment will end up with unpredictable results for example, storage overlay ABEND0C4 or ABEND0C1. See also “Step 4D: Update LPALSTxx” on page 83 and “Step 4E: Update LNKLSTxx” on page 84.

Overview of Configuration Tasks

The major tasks required for configuring SA z/OS on a focal point are listed in Table 9.

Table 9. Configuration Tasks for SA z/OS Host Systems. ✔=Required, *=Optional

<table>
<thead>
<tr>
<th>Task</th>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1: SMP/E Installation. Refer to Chapter 8, “SMP/E Installation,” on page 51</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>“Step 2: Allocate System-Unique Data Sets” on page 77</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>“Step 3: Allocate Data Sets for the ISPF Dialog” on page 81</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>“Step 4: Configure SYS1.PARMLIB Members” on page 82</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>“Step 5: Configure SYS1.PROCLIB Members” on page 86</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>“Step 6: Configure NetView” on page 88</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>“Step 7: Preparing the Hardware” on page 95</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>“Step 8: Preparing Ensemble HMC Communication” on page 101</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>“Step 9: Preparing the VM PSM” on page 103</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>“Step 10: Configure the Automation Manager” on page 108</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>“Step 11: Configure the Component Trace” on page 108</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>“Step 12: Configure the System Logger” on page 109</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>“Step 13: Configure ISPF Dialog Panels” on page 111</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Task</td>
<td>SysOps</td>
<td>ProcOps</td>
<td>I/O Ops</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>--------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>“Step 14: Verify the Number of available REXX Environments” on page 116</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>“Step 15: Configure Function Packages for TSO” on page 117</td>
<td></td>
<td>★</td>
<td></td>
</tr>
<tr>
<td>“Step 16: Configure Alert Notification for SA z/OS” on page 118</td>
<td></td>
<td>★</td>
<td></td>
</tr>
<tr>
<td>“Step 17: Compile SA z/OS REXX Procedures” on page 121</td>
<td>★</td>
<td></td>
<td></td>
</tr>
<tr>
<td>“Step 18: Defining Automation Policy” on page 121</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>“Step 19: Define Host-to-Host Communications” on page 123</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>“Step 20: Enabling SA z/OS to Restart Automatic Restart Manager Enabled Subsystems” on page 128</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>“Step 21: Define Security” on page 128</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>“Step 22: Configure the Status Display Facility (SDF)” on page 129</td>
<td>★</td>
<td>★</td>
<td></td>
</tr>
<tr>
<td>“Step 23: Check for Required IPL” on page 130</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>“Step 24: Automate System Operations Startup” on page 130</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>“Step 25: Verify Automatic System Operations Startup” on page 132</td>
<td>★</td>
<td></td>
<td></td>
</tr>
<tr>
<td>“Step 26: Configure USS Automation” on page 133</td>
<td>★</td>
<td></td>
<td></td>
</tr>
<tr>
<td>“Step 27: Enabling the System Automation Webservices Adapter” on page 133</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>“Step 28: Install an SA z/OS Satellite” on page 138</td>
<td>★</td>
<td></td>
<td></td>
</tr>
<tr>
<td>“Step 29: Installing and Customizing the NMC Focal Point” on page 139</td>
<td>★</td>
<td>★</td>
<td></td>
</tr>
<tr>
<td>“Step 30: Copy and Update Sample Exits” on page 147</td>
<td>★</td>
<td></td>
<td></td>
</tr>
<tr>
<td>“Step 31: Install Relational Data Services (RDS)” on page 148</td>
<td>★</td>
<td></td>
<td></td>
</tr>
<tr>
<td>“Step 32: Install CICS Automation in CICS” on page 148</td>
<td>★</td>
<td></td>
<td></td>
</tr>
<tr>
<td>“Step 33: Install IMS Automation in IMS” on page 151</td>
<td>★</td>
<td></td>
<td></td>
</tr>
<tr>
<td>“Step 34: Install TWS Automation in TWS” on page 152</td>
<td>★</td>
<td></td>
<td></td>
</tr>
<tr>
<td>“Step 35: Configuring GDPS” on page 154</td>
<td>★</td>
<td></td>
<td></td>
</tr>
<tr>
<td>“Step 36: Configuring I/O Operations” on page 156</td>
<td>✔️</td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>“Step 37: Installing Tivoli Enterprise Portal Support” on page 158</td>
<td>★</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Step 2: Allocate System-Unique Data Sets

<table>
<thead>
<tr>
<th></th>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Certain data sets are required several times across the focal point and target systems. This section tells you which are required on which systems or sysplexes. To allocate these data sets, sample jobs are provided in the following members of the SINGSAMP data set:

- INGALLC0
- INGALLC1
- INGALLC2
- INGALLC3
- INGALLC4
- INGALLC5
- INGALLC6

Prerequisite for running the jobs: Before you run these jobs, you need to edit them to make them runnable in your specific environment. To do so, first copy them into your private user library and then follow the instructions that are given in the comments in the jobs.

Note that the values that you fill in (such as the system name) may be different for each system where you run the jobs.

Step 2A: Data Sets for NetView

<table>
<thead>
<tr>
<th></th>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The data sets in Table 10 are required once per automation agent and cannot be shared between automation agents. They need to be referred to in the startup procedure for each automation agent’s NetView in Step 5: Configure SYS1.PROCLIB Members” on page 86.

Table 10. Data Sets for Each Individual Automation Agent

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Sample job to allocate the data set</th>
<th>Organization</th>
<th>DD name in the NetView startup procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>User-modified NetView system definitions.</td>
<td>INGALLC0</td>
<td>Partitioned</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>Stores the NetView reports, listings, files, and output from the security migration tool as well as the reports from the style sheet report generator.</td>
<td>INGALLC0</td>
<td>Library</td>
<td>DSILIST</td>
</tr>
<tr>
<td>Contains the members to be used when testing the automation table.</td>
<td>INGALLC0</td>
<td>Partitioned</td>
<td>DSIASRC</td>
</tr>
</tbody>
</table>
Step 2: Allocate System-Unique Data Sets

Table 10. Data Sets for Each Individual Automation Agent (continued)

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Sample job to allocate the data set</th>
<th>Organization</th>
<th>DD name in the NetView startup procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stores the output report produced from running tests of the automation table.</td>
<td>INGALLC0</td>
<td>Partitioned</td>
<td>DSIARPT</td>
</tr>
<tr>
<td>Contains VTAM source definitions for the sample network.</td>
<td>INGALLC0</td>
<td>Partitioned</td>
<td>DSIVTAM</td>
</tr>
<tr>
<td>NetView log data sets</td>
<td>INGALLC0</td>
<td>VSAM</td>
<td>DSILOGP, DSILOGS</td>
</tr>
<tr>
<td>NetView trace data set</td>
<td>INGALLC0</td>
<td>VSAM</td>
<td>DSITRCP, DSITRCS</td>
</tr>
<tr>
<td>DVIPA Workload Statistics</td>
<td>INGALLC0</td>
<td>Sequential</td>
<td>CNMDVIPP, CNMDVIPS</td>
</tr>
<tr>
<td>NetView save/restore data set</td>
<td>INGALLC0</td>
<td>VSAM</td>
<td>DSISVRT</td>
</tr>
</tbody>
</table>

Step 2B: Data Sets for I/O Operations

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>✔️</td>
</tr>
</tbody>
</table>

The data set in Table 11 is required once on each system where you want to have I/O operations available. It cannot be shared between systems. It needs to be referred to in the I/O operations startup procedure in “Step 5: Configure SYS1.PROCLIB Members” on page 86.

Table 11. Data Sets for I/O Operations

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Sample job to allocate the data set</th>
<th>Organization</th>
<th>DD name in the I/O operations startup procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>HCD trace file</td>
<td>INGALLC1</td>
<td>Sequential</td>
<td>HCDTRACE</td>
</tr>
</tbody>
</table>

Step 2C: Data Sets for Automation Agents

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The data sets in Table 12 are required once per automation agent and cannot be shared between automation agents. They need to be referred to in the startup procedure for each automation agent’s NetView in “Step 5: Configure SYS1.PROCLIB Members” on page 86.

Table 12. Data Sets for Each Individual Automation Agent

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Sample job to allocate the data set</th>
<th>Organization</th>
<th>DD name in the NetView startup procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automation status file</td>
<td>INGALLC2</td>
<td>VSAM</td>
<td>AOFSTAT</td>
</tr>
<tr>
<td>Dump file for diagnostic information</td>
<td>INGALLC2</td>
<td>Sequential</td>
<td>INGDUMP</td>
</tr>
</tbody>
</table>
Step 2: Allocate System-Unique Data Sets

The data set in Table 13 is required once per sysplex and cannot be shared across sysplex boundaries. It needs to be referred to in the startup procedure for each automation agent’s NetView in “Step 5: Configure SYS1.PROCLIB Members” on page 86.

Table 13. Data Set for Each Sysplex

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Sample job to allocate the data set</th>
<th>Organization</th>
<th>DD name in the NetView startup procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPL data collection</td>
<td>INGALLC4</td>
<td>VSAM</td>
<td>HSAIPL</td>
</tr>
</tbody>
</table>

Step 2D: Data Sets for Automation Managers (Primary Automation Manager and Backups)

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The data sets in Table 14 are required once per sysplex or standalone system. In the same sysplex or standalone system, they should be shared by the primary automation manager and its backups, but they cannot be shared across sysplex or standalone-system boundaries. Except for the takeover file, they need to be referred to in the automation manager startup procedure in “Step 5: Configure SYS1.PROCLIB Members” on page 86.

Each subplex requires one separate set of the following:
- The schedule override file
- The configuration information data set
- The automation manager takeover file

Table 14. Data Sets for All Automation Managers in a Sysplex or Standalone System

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Sample job to allocate the data set</th>
<th>Organization</th>
<th>DD name in the automation manager startup procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schedule override file</td>
<td>INGALLC3</td>
<td>VSAM</td>
<td>HSAOVR</td>
</tr>
<tr>
<td>Configuration information data set</td>
<td>INGALLC3</td>
<td>Sequential</td>
<td>HSACFGIN</td>
</tr>
<tr>
<td>PARMLIB</td>
<td>INGALLC3</td>
<td>Partitioned</td>
<td>HSAPLIB</td>
</tr>
<tr>
<td>Takeover file</td>
<td>INGALLC3</td>
<td>VSAM</td>
<td>—</td>
</tr>
</tbody>
</table>

Note: Use the following formula to work out the required size of the takeover file: 4000 records + n records of 4K, where n is the maximum numbers of resources.

The data sets in Table 15 on page 80 must be allocated once for each automation manager. They cannot be shared between an automation manager and its backups on the same system. Therefore, when you edit the sample job that is to allocate the data sets for a particular sysplex or standalone system, make sure that you include a fresh job step for each automation manager that you plan to have on that particular sysplex or standalone system. For more details, see the comments in the INGALLC3 sample.

Note: You can safely use the same DD names in each job step because DD names are not shared across job step boundaries.
Step 2: Allocate System-Unique Data Sets

These files also need to be referred to in the automation manager startup procedure in “Step 5: Configure SYS1.PROCLIB Members” on page 86.

Table 15. Data Sets for Each Individual Automation Manager

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Sample job to allocate the data set</th>
<th>Organization</th>
<th>DD name in the automation manager startup procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal trace files (optional)</td>
<td>INGALLC5</td>
<td>Sequential</td>
<td>TRACET0</td>
</tr>
<tr>
<td>ALLOCOUT data set</td>
<td>INGALLC5</td>
<td>Sequential</td>
<td>SYSOUT</td>
</tr>
<tr>
<td>ALOCPRRT data set</td>
<td>INGALLC5</td>
<td>Sequential</td>
<td>SYSPRINT</td>
</tr>
<tr>
<td>DUMP data set for LE environment</td>
<td>INGALLC5</td>
<td>Sequential</td>
<td>CEEDUMP</td>
</tr>
</tbody>
</table>

The generation data groups (GDGs) in Table 16 must be created once for each automation manager. They cannot be shared between an automation manager and its backups on the same system. Therefore, when you edit the sample job that is to create the GDGs for a particular sysplex or standalone system, make sure that you include a new set of GDG definitions for each automation manager that you plan to have on that particular sysplex or standalone system. For more details, see the comments in the INGALLC6 sample.

These files also need to be referred to in the automation manager startup procedure in “Step 5: Configure SYS1.PROCLIB Members” on page 86.

Table 16. Generation Data Groups for Each Individual Automation Manager

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Sample job to create the GDG</th>
<th>Organization</th>
<th>DD name in the automation manager startup procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal trace files</td>
<td>INGALLC6</td>
<td>Sequential</td>
<td>TRACET0</td>
</tr>
<tr>
<td>ALLOCOUT data set</td>
<td>INGALLC6</td>
<td>Sequential</td>
<td>SYSOUT</td>
</tr>
<tr>
<td>ALOCPRRT data set</td>
<td>INGALLC6</td>
<td>Sequential</td>
<td>SYSPRINT</td>
</tr>
<tr>
<td>DUMP data set for LE environment</td>
<td>INGALLC6</td>
<td>Sequential</td>
<td>CEEDUMP</td>
</tr>
</tbody>
</table>

Step 2E: SA z/OS Password Store Data Set

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

The data sets in Table 17 are required once for each automation agent and can be shared between automation agents. They need to be referred to in the startup procedure for each automation agent’s NetView in “Step 5: Configure SYS1.PROCLIB Members” on page 86.

Table 17. Data Sets for Each Individual Automation Agent

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Sample job to allocate the data set</th>
<th>Organization</th>
<th>DD name in the NetView startup procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password data set for INGPW</td>
<td>INGALLC2</td>
<td>VSAM</td>
<td>INGPSWD</td>
</tr>
</tbody>
</table>
Step 3: Allocate Data Sets for the ISPF Dialog

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Use the sample job INGEDLGA in SINGSAMP to allocate data sets that are required for I/O operations and the customization dialog. These data sets are normally allocated only on the focal point system where you use the customization dialog. These data sets include:

- **For system operations and processor operations:**
  - The ISPF table library data set that contains the values you enter in the customization dialog
  - The SA z/OS configuration file: this is the output data set for the customization dialog when building the SA z/OS configuration.

<table>
<thead>
<tr>
<th>Data Set Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ING.CUSTOM.AOFTABL</td>
<td>ISPF table output library for the customization dialog</td>
</tr>
<tr>
<td>ING.CUSTOM.SOCNTL</td>
<td>SA z/OS configuration files</td>
</tr>
</tbody>
</table>

- **For processor operations:**
  - The ISPF table library data set that contains the values you enter in the customization dialog
  - The SA z/OS configuration file: this is the output data set for the customization dialog when building the SA z/OS configuration.

<table>
<thead>
<tr>
<th>Data Set Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ING.CUSTOM.AOFTABL</td>
<td>ISPF customization table for customization dialog</td>
</tr>
<tr>
<td>ING.CUSTOM.POCNTL</td>
<td>Processor operations control file</td>
</tr>
</tbody>
</table>

- **For I/O operations:**
  - The I/O operations configuration file. Because you use the customization dialog to collect information and build control files, you normally need them only at the focal point. The I/O operations dialogs, however, are used to input commands and get responses from the I/O operations part of SA z/OS. Because they do not support multisystem commands for I/O operations functions, you must install them on each system, focal point or target, where you want to use them.

<table>
<thead>
<tr>
<th>Data Set Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ING.CUSTOM.IHVCONF</td>
<td>I/O operations configuration file</td>
</tr>
</tbody>
</table>

Note: Make a note of these data set names. They are used in “Step 13: Configure ISPF Dialog Panels” on page 111. If you rename the data sets, you need to adapt the corresponding names in that step.
Step 4: Configure SYS1.PARMLIB Members

The **xx** suffix on each SYS1.PARMLIB data set member can be any two characters chosen to match your IEASYS naming scheme. See z/OS MVS Initialization and Tuning Reference for information about IEASYS.

The following sections describe the SYS1.PARMLIB data set members that need to be changed and provide information about how to achieve this.

**Step 4A: Update IEAAPF**

Define authorized libraries to the authorized program facility (APF) in an IEAAPF**xx** member.

<table>
<thead>
<tr>
<th></th>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
</tbody>
</table>

Edit the IEAAPF**xx** member to add the following to the APF:

- ING.SINGMOD1, ING.SINGMOD2, ING.SINGMOD3
- SYS1.SCBDHENU (for I/O operations)

You can avoid an IPL: You can also code a PROG**xx** member to add authorized libraries to the authorized program facility (APF). If you do this, no IPL is required. For a complete description of dynamic APF and PROG**xx**, see z/OS MVS Initialization and Tuning Reference.

**Note:** Do not include SYS1.NUCLEUS.

**Step 4B: Update SCHED**

Sample: INGESCH

If you run z/OS 2.1 or higher then you can skip this step since all automation-related components are already part of the z/OS-delivered Program Property Table.

Otherwise consult the chapter "SCHED**xx**, sub-chapter "Program Property Table" in z/OS MVS Initialization and Tuning Reference to find out, which of the entries listed in sample member INGESCH are already part of the z/OS-provided PPT. Edit the SCHED**xx** member to ensure that it includes all the missing statements for INGESCH.

Compare the content of the SCHED**xx** member with the INGESCH member that resides in the SINGSAMP sample library. Edit the SCHED**xx** member so that it includes all the statements in the INGESCH member.
Step 4: Configure SYS1.PARMLIB Members

This enables the NetView subsystem interface address space, the NetView application address space (for the automation agent), the I/O operations address space and the automation manager to run without being swapped out of memory.

I/O operations exploits the MVS component trace and stores intermediate trace records in a data space. Because some trace entries are recorded outside the I/O operations address space, the data space must be common to all users. However, a common data space requires the owning address space to be non-swappable.

Step 4C: Update MPFLSTxx

<table>
<thead>
<tr>
<th></th>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
</tbody>
</table>

Sample: INGEMPF

It is recommended that you update the MPFLSTxx member after having installed the ISPF Customization Dialog (see "Step 18: Defining Automation Policy" on page 121). Using the customization dialog you can obtain a list of the messages that are involved in automation. The customization dialog also allows you to define header and trailer lines for the message list, thus building a complete MPFLSTxx member called MPFLSTSA.

In addition SA z/OS provides a sample member called INGEMPF in the SINGSAMP sample library. This contains the IDs of all of the messages that occur in the INGMSGSA NetView automation table that is delivered with SA z/OS. Thus if you concatenate both the INGEMPF member and the dynamically-created MPFLSTSA member, you obtain a list of all of the messages that are used in the INGMSGSA and INGMSG01 automation tables.

Alternatively, update the content of your MPFLSTxx member based on INGEMPF and INGMSGSA, and make sure that all of the messages that are listed there are forwarded to automation.

Step 4D: Update LPALSTxx

<table>
<thead>
<tr>
<th></th>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

Edit the LPALSTxx member to add ING.SINGMOD3 to the SA z/OS load library. There is no other choice for this library, it must be in the LPALST concatenation.

You can avoid an IPL: Because ING.SINGMOD3 contains only a few modules, you can also code a PROGxx member that enables a dynamic addition of those modules to the LPALST. If you do this, no IPL is required. For a complete description of dynamic LPA and PROGxx, see z/OS MVS Initialization and Tuning Reference.

Notes:
1. Make sure that the SA z/OS load library is cataloged in the master catalog, or copy the members in ING.SINGMOD3 to a data set that is in the master catalog.
Step 4: Configure SYS1.PARMLIB Members

2. Be sure you do not have any data sets containing load modules with prefixes of IHV, AOF, ISQ, ING, or HSA in these members.

3. If ING.SINGMOD3 is to be placed in SYS1.PARMLIB member LPALSTxx, ensure the data set organization is of type PDS.

Step 4E: Update LNKLSTxx

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

To run SA z/OS, you must ensure that program libraries can be found at startup time.

Add SINGMOD1 (recommended) and SINGMOD2 (mandatory) to the LNKLST concatenation. There is no other choice for these libraries: they must be in the LNKLST concatenation.

For the other libraries, either add them to the LNKLST concatenation or add them on STEPLIB DDs in the JCL in SYS1.PROCLIB that is used to start the products.

Adding libraries on STEPLIB DDs will involve performance degradation compared to adding them to the LNKLST concatenation and should therefore be avoided.

z/OS link list data sets no longer have to be cataloged in the master catalog. It is possible to specify a volume in the link list entry for data sets that are cataloged in user catalogs.

Edit the LNKLSTxx member to add the following to the LNKLST concatenation: ING.SINGMOD1, ING.SINGMOD2.

You can avoid an IPL: You can also code a PROGxx member to add libraries to the LNKLST concatenation. If you do this, no IPL is required. For a complete description of dynamic LSTLNK and PROGxx, see z/OS MVS Initialization and Tuning Reference.

Step 4F: Update BPXPRMxx

The zFS dataset SINGZFS contains the USS related parts of SA z/OS. It must be mounted to enable UNIX automation through SA z/OS.

Add the content of INGEBPX member, which resides in the SA z/OS sample library SINGSAMP, to your BPXPRMxx concatenated member.

You can dynamically mount the SINGZFS dataset without an IPL: Issue the MVS command D OMVS to get the current BPXPRMxx member concatenation. Use T OMVS command to activate the changed definitions of the updated BPXPRMxx members.

Step 4G: Update IEFSSNxx

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

Sample: INGESSN
Step 4: Configure SYS1.PARMLIB Members

Ensure that IEFSSNxx contains all the statements in the INGESSN sample member. If this has already been accomplished during the NetView installation there are no further updates required to this member.

Compare the contents of the IEFSSNxx member with the INGESSN member, which resides in the SA z/OS sample library. Edit the IEFSSNxx member so that it includes the subsystem records from the INGESSN member.

This defines:

- Four-character prefix used in the NetView started task names. The four-character prefix that you specify must match the four-character prefix of the NetView started task names. For example, if you specify SYSV, the names of the NetView job name must be SYSVxxxx, where xxxx are any four characters you choose. If you change this four-character prefix, you can dynamically add this entry using the z/OS command SETSSI. Otherwise you must perform an IPL of z/OS to effect the change. Please adapt the content of your IEFSSNxx member accordingly. If you run NetView 5.x then define:

  SUBSYS SUBNAME(SYSV) /* NETVIEW-SA SUBSYSTEM NAME */

If you run NetView 6.x, then define:

  SUBSYS SUBNAME(SYSV) /* NETVIEW-SA SUBSYSTEM NAME */

  INITRTN(DSI4LSIT)

- To prevent JESx from starting before SA z/OS during the IPL process, indicate that in your IEFSSNxx member accordingly.

  SUBSYS SUBNAME(JES2) /* JES2 IS THE PRIMARY SUBSYSTEM NAME */

  PRIMARY(YES) START(NO)

However if you plan to start JESx before NetView, remove the START(NO) option from your definitions in the IEFSSNxx member. For the correct syntax of your environment check the z/OS MVS Initialization and Tuning Reference.

Step 4H: Update JES3INxx

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sample: INGEJES3

If you are using JES3, compare the contents of the JES3INxx member with the INGEJES3 member which resides in the SINGSAMP sample library. You may want to review these members first to see whether there are entries in the INGEJES3 member that are already in the JES3INxx member. After merging the INGEJES3 member, be sure there are no duplicate entries in the JES3INxx member.

This includes the DUMP options and adds the JES3 parameters.

Step 4I: Update SMFPRMxx

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If you plan to use SMF records for availability reporting you must update the SMFPRMxx member in the SYS1.PARMLIB library by adding type 114 to the SYS(TYPE statement:
Step 5: Configure SYS1.PROCLIB Members

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
</table>

You need to make some changes to startup procedure members in the SYS1.PROCLIB data set. It is recommended that either you back up the startup procedure members that you are going to change or that you create new members.

Step 5A: NetView Startup Procedures

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
</table>

- **NetView Subsystem Interface Startup Procedure**
  - NetView provides a sample subsystem interface startup procedure in member CNMSJ010. Copy this member from your NetView library and adapt it to your needs:
    - Ensure that the PPIOPT parameter is set to PPI. Several SA z/OS functions use PPI communication as a base, for example, USS automation and Tivoli Enterprise Portal Support.

- **NetView Application Startup Procedure**
  - You can use the sample provided in the INGENVSA member of the SINGSAMP data set. Copy it to a member of each system's SYS1.PROCLIB data set (for the focal point system as well as for the target systems).
    - Configure each copy to your needs. In particular, do the following:
      - Make sure that the AOFSTAT, INGDUMP and HSAIPL concatenations include the data sets that you allocated in "Step 2: Allocate System-Unique Data Sets" on page 77.

**Note:** Adaptation of the JCL procedure names to meet the four-character prefix defined in the IEFSSnxx member will be done in "Step 24: Automate System Operations Startup" on page 130 when defining the jobnames for Automation NetView.

If you do not make ING01 your domain name, make a note of what your NetView domain name is. This information is needed for system operations. See also *IBM Tivoli System Automation for z/OS Defining Automation Policy* for more information on enterprise definitions.

See *Tivoli NetView for z/OS Installation: Configuring Additional Components* for further details about how to modify the NetView startup procedure.

Step 5B: Startup Procedures Required for System Operations Only

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
</table>
Step 5: Configure SYS1.PROCLIB Members

- **Automation Manager Startup Procedure**
  You can use the sample provided in the INGEAMSA member of the SINGSAMP data set. Copy it to a member of the SYS1.PROCLIB data set of all systems where System Automation will be installed and run.

  Configure that copy to your needs. **In particular, make sure that the DD concatenations mentioned in “Step 2: Allocate System-Unique Data Sets” on page 77 include the data sets that you allocated there. In addition, consider configuring the following point:**

  - If you prefer not to place the automation manager PARMLIB member in the SYS1.PARMLIB concatenation, include a HSAPLIB DD statement in the automation manager startup procedure (see also “Step 10: Configure the Automation Manager” on page 106):

    ```
    HSAPLIB DD DSN=ING.PARMLIB, DISP=SHR
    ```

    In place of ```ING.PARMLIB```, use the PARMLIB data set that you allocated in “Step 2: Allocate System-Unique Data Sets” on page 77.

- **Other System Operations Startup Procedures**
  Copy the following members from the SINGSAMP data set to members of the SYS1.PROCLIB of all systems where System Automation will be installed and run:

  **HSAPIPLC**
  This procedure gathers IPL statistics and stores the information in the IPLDATA file. Once set up, you can view sysplex-wide IPL data with the command INGPLEX IPL.

  You can give the procedure any name.

  It is recommended that you define this procedure in your automation policy as an application with the option 'START ON IPL ONLY'.

  Alternatively, you can start this procedure during every IPL. This can be accomplished by adding `COM='S HSAPIPLC, SUB=MSTR'` to a COMMANDxx parmlib member that is shared by all systems in the sysplex.

  **INGPHOM**
  This procedure is used internally by SA z/OS to process sysplex data for CF paths.

  The procedure name must not be changed.

  **INGPIPLC**
  This procedure is used internally by SA z/OS to compare IPL data.

  The procedure name must not be changed.

  **INGPIXCU**
  The procedure is used internally by SA z/OS to process sysplex data for Sysplex utilities (for example, Couple Data Set management, Coupling Facility management, and so on). Once set up, you can view and manage related Sysplex CDS and CF data with the commands INGPLEX CDS and INGPLEX CF.

  The procedure name must not be changed.

  Follow the configuration instructions that are contained in the HSAPIPLC member.
Step 5: Configure SYS1.PROCLIB Members

Note: These procedures make use of certain data sets and must have the appropriate authorizations. For details refer to "Granting NetView and the STC-User Access to Data Sets" on page 174.

Optional: Startup Procedure for the External Writer of the Component Trace
Copy member HSACTWR from SINGSAMP. At least the SYSNAME parameter must be specified before the procedure is stored in a library of the PROCLIB concatenation.

Step 5C: I/O Operations Startup Procedure

<table>
<thead>
<tr>
<th></th>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✔</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

You can use the sample provided in the INGEIO member of the SINGSAMP data set. Copy it to a member of each system’s SYS1.PROCLIB data set (for the focal point system as well as for the target systems).

Configure these copies according to your needs. In particular, do the following:

- Make sure that the HCDTRACE concatenation in the procedure includes the data set that you allocated for I/O operations in “Step 2: Allocate System-Unique Data Sets” on page 77.

If you need to specify options for HCD tracing, refer to “Defining an HCD profile” in the z/OS HCD User’s Guide for information about how to create that data set.

Step 6: Configure NetView

<table>
<thead>
<tr>
<th></th>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
</tbody>
</table>

This section discusses how to configure several aspects of NetView:

- “Step 6A: Configure NetView Alert Information”
- “Step 6B: Configure NetView DSIPARM Data Set” on page 89
- “Step 6C: Modifying NetView DSIPARM Definitions for an Automation Network” on page 94
- “Step 6D: Configure NetView for Processor Operations” on page 94
- “Step 6E: Configure the NetView Message Translation Table” on page 95
- “Step 6F: Add the REXX Function Packages to DSIRXPRM” on page 95

Step 6A: Configure NetView Alert Information

<table>
<thead>
<tr>
<th></th>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✔</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SA z/OS enterprise monitoring depends upon alert information being passed from remote systems to the focal point. Note that this is only necessary when communication is via NPDA alerts.
Step 6: Configure NetView

The NetView command SRFILTER (or SRF) establishes the conditions governing the recording of data in the hardware monitor database, the generation of messages to the authorized operator, the forwarding of alert data to a NetView focal point, and the coloring of alerts.

To ensure that the alerts required by SA z/OS for enterprise monitoring are not filtered out, the following is recommended:

- On any focal point system:
  - Issue the command: SRF AREC PASS N *
- From the remote systems:
  - Issue the command: SRF AREC PASS N *
  - Issue the command: SRF ROUTE CLEAR

These SRF commands should be included in a startup CLIST or exit because they need to be issued after every NetView startup.

If you do not want to use the SRF AREC PASS N * command to allow all alerts to pass, you should, as a minimum, allow the NTFY event type (etypes) to pass.

The NetView SRFILTER command is documented in *Tivoli NetView for z/OS Command Reference Vol. 1.*

Step 6B: Configure NetView DSIPARM Data Set

<table>
<thead>
<tr>
<th></th>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
</tbody>
</table>

Sample: INGSTGEN

A sample is provided for this step in the INGSTGEN member of the SINGSAMP library. Copy the contents of INGSTGEN to your CxxSTGEN or CxxSTUSR and configure it to match your installation. See the INGSTGEN sample for further details.

Copy any DSIPARM and SINGNPRM member that you need to configure into a data set allocated in DSIPARM before the SMP/E-maintained NetView DSIPARM and SA z/OS target libraries and edit it there.

Then change the following members in the copied NetView DSIPARM data set:

NetView Style Sheet

**Tower Statements:** The various SA z/OS components or environments are activated with the following TOWER.SA statements.

**SysOps**
- This enables application or more general resource automation.

**ProcOps**
- This enables Processor Operations.

**Satellite**
- This indicates that the SA z/OS topology manager runs on the Networking NetView for communication with RODM and the NMC.
GDPS®
This enables Geographically Dispersed Parallel Sysplex (GDPS) to run under SA z/OS. Use this definition regardless of the specific GDPS product that is running (GDPS/PPRC, GDPS/PPRC HM, GDPS/XRC or GDPS/GM).

Additionally the following GDPS subtowers are available to distinguish between the GDPS product running on the system:

- **PPRC** For GDPS/PPRC
- **HM** For GDPS/PPRC HM
- **XRC** For GDPS/XRC
- **GM** For GDPS/GM

Furthermore, code one of the following indicating whether or not this is the production versus K-system:

- **PROD** for a production system
- **KSYS** for a K-system

This information is used by SA z/OS to pick up the appropriate definition members that vary for the GDPS controlling system (K system) and the production system. For example, the K system constitutes a subplex of its own and must therefore use a different XCF group name.

GDPSAT
This statement enables the GDPS Satellite support required in a GDPS Active/Active environment.

See the INGSTGEN sample for further details about the SA tower statements.

To enable SA z/OS, make sure that the following TOWER statements are activated in the NetView style sheet (that is, uncomment them):

- **TOWER = SA**
- **TOWER.SA = SYSOPS**

**Kanji Support:** If you plan to use Kanji support make sure that you update the NetView style sheet as follows:

1. `transTbl =DSIKANJI` must be specified.
2. `transMember =CNMTRMSG` must be uncommented.

For more details, refer to the chapter “Installing the National Language Support Feature” in *Tivoli NetView for z/OS, Installation: Configuring Additional Components*.

**Automation Operator AUTO1 and AUTO2:** AUTO1 and AUTO2 refer to the NetView autotasks AUTO1 and AUTO2 supplied as samples by NetView. They are used in the initialization of SA z/OS and they should not be used by NetView for NETCONV sessions or resource discovery. To prevent the AUTO2 sample automation operator being used by NetView, do the following in the style sheet:

1. For NETCONV sessions, blank out AUTO2 in the following statement:

   ```
   function.autotask.NetConv = AUTO2
   ```

   The statement should then be:

   ```
   function.autotask.NetConv = *NONE*
   ```

2. For resource discovery, choose an autotask other than AUTO2 in the following statement:
**Step 6: Configure NetView**

```
function.autotask.autoip = AUTO2
```

**Timer Catchup Processing:** SA z/OS requires `init.TIMER=NO` for its timer catchup processing. If you do not have any timers defined in the SA z/OS policy or none of the defined timers has the `CATCHUP=YES` option, you can code `init.TIMER=YES` to cause your saved timers to be restored at NetView startup time.

Refer to the NetView documentation for details about configuring the NetView style sheet.

**AOFMSGSY (optional)**

If you have renamed any automation tasks in AOFOPFxx, you will need to make corresponding changes to the AOFMSGSY member.

Copy and edit the AOFMSGSY member that resides in ING.SINGNPRM and do the following:

1. If you want to define actions for messages that the SA z/OS NetView Automation Table does not trigger any actions for, you can use the symbol `%AOFALWAYSACTION%`.
   
   This synonym contains the action statement that is used for all messages in a Begin-End block that SA z/OS does not trigger any action for. The default, NULL, is that no action will be taken and the message does not continue to search for further matches in the same AT.
   
   See “Generic Synonyms: AOFMSGSY” in *IBM Tivoli System Automation for z/OS Customizing and Programming* for a description of these synonyms.

**NetView Automation Tables**

If you need to build NetView Automation Tables (ATs) in a way that is not supported by the customization dialog, you can use the INGMSGU1 fragment for user entries. INGMSGU1 is included before INGMSG02. You can also use the INGMSGU2 fragment for user entries. INGMSGU2 is included after INGMSG02.

If you want to have additional entries that are only valid to your environment, you can use either a separate AT (specified in the customization dialog) or use one of the user includes. The following shows the AT structure:

```
INGMSG01
   %INCLUDE AOFMSGSY
   %INCLUDE INGMSGU1
   %INCLUDE INGMSG02
   %INCLUDE INGMSGU2
```

**Message Revision Table**

During the build of the automation control file, a NetView Revision Table is being built by the customization dialog. For more information about activating the built Message Revision Table (MRT) refer to chapter ‘How to Add a Message to Automation’ in *IBM Tivoli System Automation for z/OS Customizing and Programming*.

**INGXINIT**

The communication DST initialization processing will read data that is specified in the DSIPARM member INGXINIT. Copy and edit the
Step 6: Configure NetView

INGXINIT member, which resides in ING.SINGNPRM. Uncomment the following parameters and specify your values:

**GRPID**
2-byte XCF group ID. Default is blank. The value must be the same as specified for GRPID in the corresponding member HSAPRMxx.

**PLEXID**
2-byte suffix used to build the extended XCF communication group with the name INGPX$xx. For suffix xx you can specify all alphanumeric characters except the suffix $$ (@# are acceptable, for instance). If PLEXID is not specified the automation agent is not a member of the extended XCF communication group.

**DIAGDUPMSG**
This is the number of message buffer IDs that are validated before send and after receive. This is for diagnostic purposes. A value for nnnn may be chosen between 0 (no validation) and 99999. The default is 0 and performance decreases with larger values.

**LIFECYCLE**
This parameter allows you to prepare for Life Cycle Recording in order to debug automation manager-related problems. Normally, SA z/OS Service will advise when Life Cycle Recording should be enabled.

The value of nnnn defines the size of the data space in number of megabytes (1 through 2097). A value of 500 is recommended and is sufficient in most situations.

The value of dataset specifies the fully-qualified DSN to be used when offloading the dataspace to disk.

**Note:** nnnn and dataset must be separated by a semicolon without intervening blanks. The total length of ’nnnn;dataset’ can be a maximum of 60 bytes.

**LOGSTREAM**
This parameter defines if the NetView agent should establish a connection to the system logger at initialization time. If NO is specified the agent does not establish a connection. The default is YES which causes the agent to connect to the following log streams:

- HSA.WORKITEM.HISTORY
- HSA.MESSATE.LOG

You may also specify the value GRPID. This allows you to have separate log streams per subplex. If GRPID is specified and the value of the GRPID keyword is not blank, the agent connects to the following log streams:

- HSA.GRP$xx.WORKITEM.HISTORY
- HSA.GRP$xx.MESSAGE.LOG

If the value of the GRPID keyword is blank the LOGSTREAM keyword defaults to YES.

**Note:** Both values, LOGSTREAM and GRPID, must be the same as in the PARMLIB member HSAPRM$xx that is used to start the related automation manager.
Step 6: Configure NetView

PPI  This needs to be set to YES to establish a connection to the end-to-end automation adapter.

PPIBQL  The number of elements in the PPI queue—this indicates how large the response to a request may be. It should be greater than the number of queue elements that you expect to be returned. The default is 3000.

All input requests flow into the PPI queue, so the buffer queue limit, PPIBQL, should match this. If this limit is exceeded (that is, the queue limit is too small):

• The automation adapter might not be able to send any further requests to the SA z/OS agent, and the agent issues a JNI exception with return code 1735:

  INGX9B20E JNI function ingjppi failed with return code 1735.

• The SA z/OS agent might not be able to send any responses to the automation adapter, and an AOF350E message is issued.

If you receive these error messages, increase the buffer queue limit.

Requests are lost, but the end-to-end automation operator will receive exception reports. For more details see IBM Tivoli System Automation for z/OS End-to-End Automation Adapter.

All parameter values must match with the respective parameters in the PARMLIB member HSAPRMxx of the automation manager.

You can specify a GRPID to indicate that a subset of the members of an actual z/OS sysplex is defined in a sysplex group. If specified, the ID may contain 1 or 2 characters. Valid characters are A–Z, 0–9, and the national characters ($, # and @).

The GRPID is prefixed with the string INGXSG to construct the XCF group name that is used for cross system synchronization, for example, INGXSGxy.

If you do not specify a GRPID, the default group name INGXSG is used.

Note: Syntax errors are reported by a message with error code ERRCODE=564. Any syntax errors will stop the initialization process and therefore no automation will be possible.

The following parsing syntax applies:
• Data can only be specified via key-value-pairs.
• One or more parameters may be specified on one line.
• Each record will be parsed for the keyword.
• Parsing will be stopped and any further input data will be ignored after all keywords listed above are found.
• If the same parameter is specified multiple times, the last one is used.
• For any keyword that was not specified, the default value is blank.
• No blanks between parameters and values are allowed.
• The syntax of a keyword is equal to the syntax of the parmlib member HSAPRMxx.

An example of a valid syntax is:

GRPID=XY, LIFECYCLE=500, LOGSTREAM=YES
Step 6: Configure NetView

An example of an invalid syntax is:
GRPID = 34, LIFECYCLE = 500

Step 6C: Modifying NetView DSIPARM Definitions for an Automation Network

<table>
<thead>
<tr>
<th></th>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
</tbody>
</table>

Note: The following information refers to setting up a single NetView automation network.

To support an automation network, you need to add or modify NetView definitions in the NetView DSIPARM data set member AOFOPFGW.

AOFOPFGW Modifications
In the AOFOPFGW member for each system, define the operator IDs used for both outbound and inbound gateway autotasks.

For example, in Figure 5 on page 34, the gateway autotask definitions in AOFOPFGW on domain CH101 are:
GATCH101 OPERATOR PASSWORD=GATCH101
PROFILEN AOFPRFAO
GATCH102 OPERATOR PASSWORD=GATCH102
PROFILEN AOFPRFAO
GATCH103 OPERATOR PASSWORD=GATCH103
PROFILEN AOFPRFAO

Step 6D: Configure NetView for Processor Operations

<table>
<thead>
<tr>
<th></th>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✔</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To enable SA z/OS, make sure that the following TOWER statements are activated in the NetView style sheet:
TOWER = SA
TOWER.SA = SYSOPS PROCOPS

For SNMP, BCP internal interface connections, and HTTP connections, it is mandatory to make the security definitions described in “Controlling Access to the Processor Hardware Functions” on page 184.

Processor operations uses automation table entries for its operation. The required AT entries are shipped as part of the SysOps automation table INGMSGSA and are activated if the ProcOps TOWER statement is specified.

ISQMSGU1
This empty member is supplied by processor operations and is included in the SysOps automation table ISQMSG01. By inserting your own automation entries or include statements of your own automation tables here, you can expand processor operations with your own automation routines which may utilize the processor operations supplied command API.
Step 6: Configure NetView

ProcOps user AT ISQMSGU1 is included in INGMSG01, so it runs in parallel to the ProcOps AT entries which are shipped by System Automation.

Step 6E: Configure the NetView Message Translation Table

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If you use Kanji support, the NetView Message Translation Table that was specified in the NetView style sheet with the transMember entry needs to be configured. (The NetView default for the Message Translation Table is CNMTRMSG located in library SDSIMSG1.)

Verify that in the CNMTRMSG member the INCLUDE for CNMMSJPN is uncommented:

```%INCLUDE CNMMSJPN
```

In addition add includes for the SA z/OS Kanji message members at the beginning of CNMTRMSG:

```%INCLUDE AOFJ
%INCLUDE EVEJ
%INCLUDE EVIJ
%INCLUDE EVEJ
%INCLUDE INGJ
%INCLUDE ISQJ```

Note that only the fixed text of the messages has been translated. Any variables inserted into the text cannot be translated using NetView services, even if the variable contains text strings that are in principle translatable.

Step 6F: Add the REXX Function Packages to DSIRXPRM

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
</tbody>
</table>

All NetView REXX functions of SA z/OS are packaged in module INGRXFPG. This package will be automatically loaded to the function package table in NetView at initialization time.

If you plan to use the CICSpex System Manager REXX API then specify the following line in your NetView stylesheet:

```
REXX.FUNCPKGLIST.SYS.EYU9AR00=EYU9AR00
```

Step 7: Preparing the Hardware

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
</tbody>
</table>

The steps described in this section are necessary to prepare your Hardware Management Console (HMC) and Support Elements according to the processor
Step 7: Preparing the Hardware

hardware interface you are using. For details about planning the hardware interface, refer to “Planning the Hardware Interfaces” on page 18.

In addition, refer to the publications Hardware Management Console Guide and Support Element Operations Guide for details about your HMC and SE.

Note that with workplace version 2.10 or later, the user interface (UI) style of the console can be changed from 'Classic Style' to 'Tree Style' with the User Settings task. The following preparation steps are based on the 'Classic Style' for compatibility reasons with workplace version 2.9.2.

Step 7A: Preparing the HMC (Console Workplace 2.9 and Later Versions)

Enable the HMC API, Set the SNMP Community Names and the SNMPv3 Information

In order to control a CPC using an HMC instead of the CPC's Support Element, the Hardware Management Console API function must be enabled. If you do not plan to use an HMC to control your CPCs over the TCP/IP SNMP ProcOps interface, omit this task. To complete this task:

1. For this task, you need to be logged on in Access Administrator mode on your HMC.
2. Select Console Actions and click on the Hardware Management Console Settings icon.
3. Click on the Customize API Settings icon. Make sure the Enable SNMP APIs check box is set in the Customize API Settings window.
   **Important**: The window field SNMP agent parameters must be empty. Any data in this field will prevent the console application from establishing an API session successfully.
4. For SNMP connections to the HMC, the community names must be defined. After that, you can use native SNMP commands to query and set HMC object attributes, or you can use SA z/OS ProcOps to manage CPCs defined on the HMC and to execute CPC HW commands over the SA z/OS ProcOps SNMP interface. A CPC is controlled over the SNMP interface when it is configured for connection protocol SNMP, using the Processor (CPC) entry in the SA z/OS Customization Dialog. See “Step 13: Configure ISPF Dialog Panels” on page 111 and “Step 18: Defining Automation Policy” on page 121 for further details on maintaining the SA z/OS Policy Database.
   **Note**: The Customize API Settings window must be open.
5. For a new ProcOps SNMP interface community name, select the Community Names table Add push button. In the Community Name data entry window enter the following information:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specify the name in uppercase with a maximum length of 8 characters. Record this name and use it when you go to define the processor entry for the CPC in your SA z/OS policy database with connection type SNMP.</td>
</tr>
<tr>
<td>Address</td>
<td>Use the IP address of your SA z/OS ProcOps focal point system.</td>
</tr>
</tbody>
</table>
Step 7: Preparing the Hardware

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Mask</td>
<td>Use 255.255.255.255. to make sure that only the addressed focal point can control the CPC. You may change the netmask to allow multiple focal point systems to control your CPC with the same community name. Specify 0.0.0.0 as the address and network mask if you want to allow access from any location in your network to your CPC, using the community name defined.</td>
</tr>
<tr>
<td>Access Type</td>
<td>Select the Read/Write radio button.</td>
</tr>
</tbody>
</table>

6. Select the **OK** push button to save the changed settings and close the data entry window.

7. In order to support SNMPv3 protocol, providing more security and traffic encryption, the SNMPv3 User Information must be defined. Then, you can use SA z/OS ProcOps to manage CPCs defined on the HMC and to execute CPC HW commands over SA z/OS ProcOps SNMP interface. A CPC is controlled over the SNMP interface using SNMPv3 protocol when it is configured for connection protocol SNMP and SNMPv3 is enabled at the Processor (CPC) entry in the SA z/OS Customization Dialog. See “Step 13: Configure ISPF Dialog Panels” on page 111 and “Step 18: Defining Automation Policy” on page 121 for further details on maintaining the SA z/OS Policy Database.

*Note:* The Customize API Settings window must be open.

8. For a new ProcOps SNMPv3 User, select the **SNMPv3 Users** table Add push button. In the SNMPv3 User Information data entry window, enter the following information:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| User Name     | Specify an SNMPv3 user name. The user name must be at least 8 characters in length and cannot exceed 31.  
**Note:** HMC allows specification of user names of length 32, but the SA is limited to a support maximum of 31 characters for SNMPv3 Users. |
| Password      | Specify a password for the SNMPv3 user. The password must be at least 8 characters in length and cannot exceed 31.  
**Note:** HMC allows specification of passwords length 32, but the SA is limited to support maximum of 31 characters for the SNMPv3 passwords |
| Access Type   | Select the **Read/Write** radio button.                                                                                                                                                                       |

9. Select the **OK** push button to save the changed settings and close the data entry window.

10. If you have finished the SNMP API settings, select the **Apply** push button of the Customize API Settings window to save the changes.

11. The SNMP Configuration Info window is displayed to inform you that the HMC console must be restarted to activate your configuration changes.

**BCP Internal Interface**

To prepare the master HMC, carry out the following steps:
Step 7: Preparing the Hardware

1. Log on to the HMC in your LAN that is to be used for change management operations with a user ID having SYSPROG or ACSADMIN authority. The HMC must have the CPC objects of your sysplex in its Defined CPCs Group.

2. Select Console Actions and click on the Hardware Management Console Services icon.

3. Select the Customize Console Services icon.

4. Make sure the LIC Change field in the Console Services window is set to Enabled.

5. Select the OK push button to save the change, or the Cancel push button if the LIC Change radio button was already set to Enabled.

Usually, there is one HMC in a CPC LAN environment that has LIC change permanently enabled. It will automatically be used by the BCP internal interface. Make sure that this HMC has all CPC objects of your sysplex in its Defined CPCs Group.

CPC Object Definitions on the HMC

Depending on the processor hardware interfaces, the CPCs that are to be managed must be defined to the HMC. For SA z/OS’s BCP internal interface, the master HMC, which must have the ‘LIC Change’ service enabled, is used as a router between the CPC where SA z/OS is running, and other targeted CPCs.

For SA z/OS’s ProcOps SNMP connection, the HMC serves as a single point of control. Alternatively, SA z/OS ProcOps can be configured to communicate directly with a CPC, by addressing its Support Element.

For detailed information about how to add, change, or remove CPC object definitions on a HMC, refer to the current Hardware Management Console Operations Guide (SC28-6821). Note that this manual is also available in the Books Work Area on the HMC.

Step 7B: Preparing the SE (Console Workplace 2.9 and Later Versions)

Enable the SE API, Set the Community Name and the SNMP Information

To control a CPC with the SA z/OS hardware interfaces BCPII or SNMP ProcOps directly, the CPC Support Element API function must be enabled. To complete this task:

1. For this task, you need to be logged on in Access Administrator mode on your HMC.

2. Select Console Actions and click on the Support Element Settings icon.

3. Click on the Customize API Settings icon. Make sure the Enable SNMP APIs check box is set in the Customize API Settings window.

Important: The window field SNMP agent parameters must be empty. Any data in this field will prevent the console application from establishing an API session successfully.

4. Set the Community Name for SNMP and ProcOps Connections. For SNMP connections to the SE, the community names must be defined. After that you can use native SNMP commands to query and set SE object attributes, or you can use SA z/OS ProcOps to manage the CPC and to execute CPC HW commands using the SA z/OS ProcOps SNMP interface. A CPC is controlled over the SNMP interface when it is configured with connection protocol
Step 7: Preparing the Hardware

SNMP in the Processor (CPC) entry of the SA z/OS Customization Dialog. See “Step 13: Configure ISPF Dialog Panels” on page 111 and ”Step 18: Defining Automation Policy” on page 121 for further details on maintaining the SA z/OS Policy Database.

5.

a.

Note: The Customize API Settings window must be open.
For a new ProcOps SNMP interface community name, select the Community names table Add push button. In the Community Name data entry window enter the following information:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specify the name in uppercase with a maximum length of 8 characters. Record this name and use it when you are going to specify the processor entry for the CPC in your SA z/OS policy database with connection type SNMP.</td>
</tr>
<tr>
<td>Address</td>
<td>Use the IP address of your SA z/OS ProcOps focal point system.</td>
</tr>
<tr>
<td>Network Mask</td>
<td>Use 255.255.255.255 to make sure that only the addressed focal point can control the CPC. You may change the netmask to allow multiple focal point systems to control your CPC with the same community name. Specify 0.0.0. as the address and network mask if you want to allow access from any location in your network to the SE, using the community name defined.</td>
</tr>
<tr>
<td>Access Type</td>
<td>Select the Read/Write radio button.</td>
</tr>
</tbody>
</table>

b.

Note: The Customize API Settings window must be open.
For a new BCP internal interface community name, select the Community Names table Add push button. In the Community Name data entry window enter the following information:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specify the name in uppercase with maximum length of 8 characters. Record this name and use it when you go to define the processor entry for the CPC in your SA z/OS policy database with connection type INTERNAL.</td>
</tr>
<tr>
<td>Address</td>
<td>The required address is 127.0.0.1</td>
</tr>
<tr>
<td>Network Mask</td>
<td>The required address is 255.255.255.255</td>
</tr>
<tr>
<td>Access Type</td>
<td>Select the Read/Write radio button.</td>
</tr>
</tbody>
</table>

6. Select the OK push button to save the changed settings and close the data entry window.

7. In order to support SNMPv3 protocol, providing more security and traffic encryption, the SNMPv3 User Information must be defined. Then, you can use
Step 7: Preparing the Hardware

SA z/OS ProcOps to manage CPCs defined on the HMC and to execute CPC HW commands over SA z/OS ProcOps SNMP interface. A CPC is controlled over the SNMP interface using SNMPv3 protocol when it is configured for connection protocol SNMP and SNMPv3 is enabled at the Processor (CPC) entry in the SA z/OS Customization Dialog. See "Step 13: Configure ISPF Dialog Panels" on page 111 and "Step 18: Defining Automation Policy" on page 121 for further details on maintaining the SA z/OS Policy Database.

Note: The Customize API Settings window must be open.

8. For a new ProcOps SNMPv3 User, select the SNMPv3 Users table Add push button. In the SNMPv3 User Information data entry window, enter the following information:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| User Name | Specify an SNMPv3 user name. The user name must be at least 8 characters in length and cannot exceed 31.  
**Note:** SE allows specification of user names of length 32, but the SA is limited to a support maximum of 31 characters for SNMPv3 Users. |
| Password | Specify a password for the SNMPv3 user. The password must be at least 8 characters in length and cannot exceed 31.  
**Note:** SE allows specification of passwords length 32, but the SA is limited to support maximum of 31 characters for the SNMPv3 passwords. |
| Access Type | Select the Read/Write radio button. |

9. Select the OK push button to save the changed settings and close the data entry window.

10. If you have finished the API settings, select the Apply push button of the Customize API Settings window to save the changes.

11. The SNMP Configuration Info window is displayed to inform you that the SE console must be restarted to activate your configuration changes.

Set the Cross Partition Flags:
This task is only required if you use the BCP internal interface to connect processor hardware running in LPAR mode. For this task, you need to be logged on in System Programmer mode on your CPC’s Support Element. To complete this task:

1. Click on the CPC Group and highlight the CPC icon.
2. Select the CPC Operation Customization task.
3. Click on the Change LPAR Security icon. The window displayed shows the security settings from the active IOCDS for the logical partitions defined on this CPC.
4. For each logical partition that should use the BCP internal interface to control another partition on this CPC, check the Cross Partition Authority checkbox.

Step 7C: Preparing the SE (Console Workplace 2.10 and Later Versions)

Enabling Capacity Change API Requests
To be able to perform capacity changes (for example, CBU) using the SA z/OS hardware interfaces BCPii or SNMP ProcOps, the ‘Allow Capacity Change API requests’ flag must be set:

1. For this task, you need to be logged on in Access Administrator mode on your HMC.
2. Select Console Actions and click on the Support Element Settings icon.
Step 7: Preparing the Hardware

3. Click on the Customize API Settings icon. Make sure the 'Allow Capacity Change API Requests' check box is set in the Customize API Settings window.

Step 7D: Updating Firewall Information

This step is only needed if you use ProcOps and intend to use TCP/IP based communication to your target processors.

Connection protocol SNMP

This communication protocol internally uses port number 3161. If there are firewalls installed between the LAN that the ProcOps FP belongs to and the processor LAN that the SEs or HMCs belong to, you should:

- Inform your network administrator to make sure that communication requests that come from SEs/HMCs with this port number are accepted.

Step 8: Preparing Ensemble HMC Communication

<table>
<thead>
<tr>
<th></th>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The steps described in this section are necessary to prepare your environment to communicate with the ensemble Hardware Management Console (HMC). For details about planning the hardware interface, refer to "Planning the Hardware Interfaces" on page 18. If you do not plan to manage your zEnterprise zBX Blade Centers using ProcOps interface, omit this step.

In addition, refer to the publications: System z Hardware Management Console Operations Guide Version 2.11.1 (SC28-6905-01) or later as well as to the zEnterprise System Hardware Management Console Operations Guide for Ensembles Version 2.11.1 (SC27-2615-01) or later.

Step 8A: Setting up the Ensemble Hardware Management Console for use with System Automation for z/OS

Refer to Appendix F, “Ensemble Hardware Management Console Setup,” on page 251 for further details.

Step 8B: Setting up AT-TLS for the SSL socket connection

In order to communicate to the Web Services API of the zEnterprise System Hardware Management Console (HMC), the following setup actions are required on the z/OS system where the ProcOps focal point can run.

Policy agent (PAGENT) setup.

1. Please refer to the z/OS Communication Server documentation for details. Be aware that the TCP/IP profile selected for the ensemble zBX management has to contain the statement "TCPCONFIG TTLS" to result in the activation of the processed policy definitions and the statement “AUTOLOG PAGENT ENDAUTOLOG” to result in the automatic start of the PAGENT.

AT-TLS Policy

2. Modify PAGENT environment variables to run with the AT-TLS configuration required for the SSL communication. For information on the environment variables, refer to the IP Configuration Guide.
Figure 8 is a sample AT-TLS policy with the TCPIP trace level 4. Please specify <tlsKeyring> and <ip_addr> accordingly. The minimal required cipher suite is the TLS_RSA_WITH_RC4_128_MD5.

Certificate registration in keyring

3. Upload the HMC certificate file (prepared in “Step 8A: Setting up the Ensemble Hardware Management Console for use with System Automation for z/OS” on page 101) to z/OS with ASCII to EBCDIC translation and add it to the NetView userid’s keyring.
Step 8: Preparing ensemble HMC communication

Add uploaded SSL-Certificates to the user's keyring as described in "Adding SSL-Certificate to userid's keyring" on page 182.

Step 9: Preparing the VM PSM

This step is only needed if you use ProcOps to control VM second level systems. The PSM is the communication partner for ProcOps to do this.

Installing the PSM Code on VM

The following parts are shipped as part of the Second Level Guest Support feature:
- In xxx.SINGOBJV — module ISQVMAIN (this is the PSM control program's main thread)
- In xxx.SINGREXV the following squished REXX programs:
  - ISQRGEC
  - ISQRCRSRV
  - ISQRMSRV
  - ISQRLRGR
  - ISQRCNSV
  - ISQRMHDL
- In xxx.SINGMSGV — Message definitions ISQUME

To install the VM parts perform the following steps:
1. Copy the object module ISQVMAIN to the VM file system for the PSM machine as file ISQVMAIN TEXT
2. Copy REXX programs to the VM file system for the PSM machine as files:
   - ISQRGEC REXX
   - ISQRCRSRV EXEC
   - ISQRMSRV EXEC
   - ISQRLRGR EXEC
   - ISQRCNSV EXEC
   - ISQRMHDL EXEC
3. Copy message definition ISQUME to the VM file system for the PSM machine as file ISQUME REPOS
4. Enter the following commands on the PSM machine (These may be created as an CMS EXEC if necessary). The name chosen for the operand of the GENMOD command (ISQPSM in this case) defines the name of the PSM control program. Any name may be chosen. These commands create the load module for the PSM main thread and the messages definitions for all threads.
   GENMSG ISQUME REPOS A ISQ
   SET LANG (ADD ISQ USER
   GLOBAL TXTLIB DMSAMT VMMLIB VMLIB
   LOAD ISQVMAIN
   INCLUDE ISQUME
   INCLUDE VMSTART (LIBE RESET VMSTART
   GENMOD ISQPSM
5. Create the two files ISQADDRS DATA and ISQPARAM DATA as described in "Customizing the PSM" on page 104.
Step 9: Preparing the VM PSM

If these steps are processed successfully then the PSM can be started.

**Configuration**

1. Provide TCPIP connection between the VM host system and the SA z/OS systems that are running NetView ProcOps.
2. Define a ProcOps Service Machine in each VM host. This is a regular virtual machine that IPLs a CMS when it starts. Ensure that it has a minimum of 32 MB of storage defined.
3. Use the IUCV directory control statement to authorize the PSM virtual machine to connect to the CP message service (*MSG). For more information about the IUCV statement, see the z/VM: Planning and Administration book.
4. Authorize the ProcOps Service Machine to use CP and CMS commands. The following commands are used by the PSM:
   ```
   SET SECUSER vmachine *
   SET EMSG
   TERMINAL MORE
   SET VMCONIO
   SET CPCONIO
   GLOBALV
   XAUTOLOG
   FORCE
   XMITMSG
   SEND
   SMSG
   QUERY NAMES
   QUERY vmachine
   ```
5. Optionally, ensure that the language is set automatically and that the ProcOps Service Machine starts when the PSM virtual machine starts by creating a PROFILE EXEC for virtual machine (if one does not already exist) and adding the appropriate commands to it:
   ```
   SET LANG (ADD ISQ USR
   ISQPSM
   ```
   where ISQPSM is the name of the control program in the earlier example.
6. Ensure that the ProcOps Service Machine has appropriate dispatching priority. Ideally it should have a higher dispatching priority than the guest machines that it manages.
7. Define the PSM as a Service Virtual Machine.
8. For each guest machine, ensure that the PSM virtual machine is defined as its secondary user.
9. Define SYSCONS as a NIP console and MCS console for each guest MVS machine, with appropriate routing codes.
10. It is recommended that the PSM virtual machine has read access to the minidisk that holds the TCPIP program, so that the NETSTAT command can be issued as part of problem determination procedures.

**Customizing the PSM**

The PSM uses two files to set parameters for its operation. These files are read at the time that PSM is initialized, and are not read subsequently.

The statements in them determine the various operational characteristics.
Step 9: Preparing the VM PSM

Each file is a simple sequential file that must be part of the file system available to the PSM virtual machine. Normally they are files on the A-disk. Each file must be available at PSM initialization. If any is missing, the PSM terminates.

ISQADDRS DATA
The ISQADDRS DATA file specifies those IP addresses that may enter requests to the PSM. Each ProcOps NetView that issues requests to the PSM must have its IP address specified.

Each record of the file specifies a single IP address. Any record that has an asterisk in the first position is treated as a comment. Any record that has the string "/*" in the first two positions is treated as a comment.

The IP address may be specified either in the normal dotted decimal form, or as a node name that is known to TCPIP on the PSM’s node for IPv4 connections, or in the preferred conventional form for IPv6 connections. If a node name is specified and that node name has several addresses, all addresses that are returned are used.

Note node names cannot be used to validate IPv6 connections and are ignored if the PSM is running an IPv6 environment.

An example of a valid file is as follows:

* Normal focal point NetView
  9.152.80.253
/* the backup
  9.152.80.254
* another system identified by its node name
  nv.boekey3.de.ibm.com
* a shorter, if infrequent form of IP address
  44.55
* Normal focal point Netview IPv6
  FD00:9:152:40:840:FFFF:80:253
/* the backup IPv6

The addresses are not checked for validity when they are read.

ISQPARM DATA
The ISQPARM DATA file specifies operational options for the PSM.

Each record of the file specifies a single parameter. Any record that has an asterisk in the first position is treated as a comment. Any record that has the string "/*" in the first two positions is treated as a comment.

The statements are of the form:
keyword = value

All keywords, except TCPIPNAME, PSMIPV4, and CLEANUP must be specified. If any required keywords are omitted the PSM will terminate. The keywords may be entered in upper, lower or mixed case. Values must be entered as required. If a keyword specification is entered more than once, the latest specification is used.

Valid keywords are:

MESSAGE_SERVER_PORT
The port number that will be used by the Message Server. (That is, the port on which it issues a TCPIP LISTEN request.) This is a number in the range
Step 9: Preparing the VM PSM

1-65335. Consult with your network programmer to ensure that this is a port number that is not used by any other processes.

**COMMAND_SERVER_PORT**
The port number that will be used by the Command Server.

**SECURITY**
The authorization token used to authenticate both the Message Server and Command Server. This must match the authorization token that is specified in the System Automation Customization dialogs for this PSM Target Hardware. This must have the correct (upper) case.

**TCPIPNAME**
The name of the TCPIP virtual machine that will provide the connections to ProcOps NetView. When the PSM control program starts, it checks that this virtual machine is running before issuing any TCPIP requests. The default value used, if TCPIPNAME is not specified, is TCPIP.

**MAX_MESSAGES**
The maximum number of messages that may be stored at any instant in the Message Queue. When the number of messages in the queue exceeds this number, the Message Handler thread terminates with an error message.

**TRACE_TYPE**
The trace type identifies the trace type value that is entered into log records written by the Logger thread.

**PSMIPV4**
You should set this keyword to Y to indicate that PSM should enforce IPv4 sockets in an IPv6-enabled environment. Supported values are Y or N. If PSMIPV4 is not specified, default value N is used by the PSM and IPv6 will be preferred.

**CLEANUP**
The number of days to retain logger files. The default is 0, which means that old ISQLOG files will not be removed. The valid range is 0 to 365 days. If the value is specified, all logger files older than CLEANUP days will be automatically removed from A-disk.

An example of a valid file is:

```
Message_server_port = 5556
Command_server_port = 4444
* TRACE_TYPE = 555
security = ISQHELLO
max_messages = 20
```

**Logger Files**
The PSM must also have sufficient writeable space on its A-disk to accommodate the logger files and any files that might be used by CP commands such as DUMP, if used.

---

Step 10: Configure the Automation Manager

<table>
<thead>
<tr>
<th></th>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✔</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Step 10A: XCF Characteristics

SA z/OS uses XCF characteristics with any communication method. Ensure that transport classes for CLASSLEN(956) and CLASSLEN(4028) are defined. An XCF group name should not be assigned to the transport classes.

When setting up the sysplex you need to be aware that SA z/OS has a maximum XCF message length of 3500 bytes. You can either use an existing transport class with the appropriate class length, or define a new transport class.

Step 10B: Configuring HSAPRMxx

The HSAPRMxx PARMLIB member contains information required for the initialization of the automation manager and default values for other operational parameters. The member is designed to be used in common by all automation manager instances in the automation subplex.

Alternatively you can put the automation manager PARMLIB member in any partitioned data set. Then, you need to specify the HSAPLIB DD statement in the automation manager startup procedure member.

A sample member called HSAPRM00 is provided in the SINGSAMP sample library. This sample is automatically copied into the PARMLIB of the automation manager (DD name HSAPLIB) when you allocate this data set as described in “Step 2: Allocate System-Unique Data Sets” on page 77. Refer to Appendix G, “Syntax for HSAPRM00,” on page 255 for the contents of this sample and the description of the parameters.

Step 10C: ARM Instrumentation of the Automation Manager

The automation manager can be enabled for Automatic Restart Manager (ARM). However, this is optional and not recommended if you use the *BASE best practice policy.

A job skeleton is provided in the SINGSAMP sample library as member HSADefa to define the SA z/OS specific Automatic Restart Manager policy.

You can define a policy allowing you to keep the number of automation manager instances on a certain level.

In a single system environment

With more than one automation manager active, ARM can automatically restart a failing primary instance. One of the automation managers that survived will take the primary role and the restarted instance will become a backup instance.

If there is only one automation manager active on a single system, ARM will automatically restart this instance again. It becomes the primary instance again and runs the takeover. The takeover time is extended by the time needed for the address space restart.

In a sysplex (subplex) environment

ARM will always restart the failing instance on the same system. Either there is already a backup waiting or the restarted instance will take over.

SA z/OS provides a policy sample with the following major options:

- Restart only for an address space ABEND (Option ELEMTERM). Restart in case of a system breakage is not supported.
Step 10: Configure the Automation Manager

The concept of the automation manager availability follows a 'floating' master model. It is a peer model with one or more backup instances on different systems already active and waiting to take over. Whenever a complete system goes away the failed automation managers (backup or primary) are not restarted somewhere else.

- The ARM element name is a 16 byte string concatenation HSAAM_sysnamexy with:
  
  HSAAM_
  
  is a string constant as prefix

  sysname
  
  Is the XCF member name of the automation manager which is the 8 byte MVS system name padded with '$', for example, MVS1$$

  x
  
  Is a one byte digit (one of 1, 2, ... 9) automatically determined at initialization time

  y
  
  Is a blank

- The restart command is the unchanged original start command, however the start mode is always HOT.
- There are no restart dependencies (no Waitpred processing)

Step 10D: Security Considerations

The started task that invokes the automation manager (see INGEAMSA in the sample library) must have the following access rights:

1. If the automation manager is to be started with option BLOCKOMVS=YES the started task must be defined by RACF as a superuser for UNIX System Services. For more information about BLOCKOMVS refer to Appendix G, “Syntax for HSAPRM00,” on page 255.

2. If you are not a superuser, you must have access to the OMVS segment.

3. Read access for the SYS1.PARMLIB data set.

4. Write access to the log streams.

5. Write access to the following data sets:
   - Trace data sets
   - The schedule override file
   - The configuration information file (DDname HSACFGIN)
   - The takeover file

Step 11: Configure the Component Trace

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️</td>
<td></td>
<td>✔️</td>
</tr>
</tbody>
</table>

Both the system operations component and the automation manager use the z/OS component trace for debugging purposes. The following setup must be done:

- Copy the CTIHSAZZ member from the SINGSAMP sample library to SYS1.PARMLIB. Do not change this member.
- Copy the CTIIHVZZ member from the SINGSAMP sample library to SYS1.PARMLIB. You may change this member to meet your requirements. Refer to the appendix “Problem Determination” in IBM Tivoli System Automation for z/OS User’s Guide for more information.
Step 11: Configure the Component Trace

- Copy the HSACTWR member residing in the SINGSAMP sample library into SYS1.PROCLIB.
- Allocate the trace data set used by the component trace. You can use the sample job HSAJCTWR in SINGSAMP to allocate the data set. Modify the sample job where appropriate.

Note: Make sure that the job invoking the ITTTRCWR module (see HSACTWR member in the sample library) has write access to the trace output data set.

Step 12: Configure the System Logger

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Although this step is optional, it is, however, recommended. The automation manager writes history information to the z/OS system logger and the automation agents read from it.

If you do not perform this step, users will not get any output from the INGHIST commands.

Notes:
1. The LOGSTREAM parameter in the HSAPRMxx parmlib member is set to YES by default. The automation manager connects to the logger address space at initialization.
2. If you set the LOGSTREAM parameter to NO, no access is established to the system logger. "Step 12: Configure the System Logger" is then unnecessary.
3. If you set the LOGSTREAM parameter to GRPID, the automation manager connects to the logger address space at initialization time. However, the log streams to which the automation manager connects depend on the value of the GRPID parameter. For more information, see Appendix G, “Syntax for HSAPRM00,” on page 255.

To exploit the system logger, the following must be fulfilled:
- Systems in a sysplex must run in XCF mode and the following must be defined in SYS1.PARMLIB(IEASYSxx):
  
  PLEXCFG=MULTISYSTEM
- For standalone systems the following must be defined in SYS1.PARMLIB(IEASYSxx):
  
  PLEXCFG=MONOPLEX

Next, the LOGR couple data sets must be formatted, if this has not already been done. For this task you can use the sample JCL provided in the HSAJFCDS member of the sample library.

Use the following sample JCLs to define the log stream in different environments:
- For a single system environment, use the sample JCL provided in member HSAJDLGM (for the automation manager)
- For a sysplex, use the sample JCL provided in member HSAJDLGS (for the automation manager)
Step 12: Configure the System Logger

In both cases you may want to adapt the HLQ parameter in the LOGR policy according to your environment. The default is IXGLOGR. Use the corresponding HSAJDxxx members as input and make the changes accordingly.

**Note:** Do not change the provided MAXBUFFSIZE values in the HSAJDxxx job. The provided values match the size of the expected data.

For a sysplex environment, you must additionally add the log structures to the CFRM policy:

```plaintext
STRUCTURE NAME(HSA_LOG)
  SIZE(9216)
  FULLTHRESHOLD(0)
  PREFLIST(cfname,cfname)
```

In this CFRM policy, you have to adapt the PREFLIST for structure HSA_LOG if you are setting up the system logger. Also adapt the SIZE parameter to a recommended minimum of 8 megabytes (8M). Since System Logger manages the space of the structure there is no need for additional monitoring. The parameter FULLTHRESHOLD(0) disables XES monitoring and potential IXC588E messages.

If you are running on z/OS 1.9 or above you will need to increase the size of for the CFRM policy to a minimum of 9216K. The minimum size for z/OS 1.9 CF level 16 is 9216K. You will also need to modify the sample HSAJDLGS to increase the size as well. You may see message IXL015I STRUCTURE ALLOCATION INFORMATION indicating the size specified was not large enough.

The system logger must be authorized. If it is not yet assigned either privileged or trusted RACF status, or both, refer to chapter “Planning for System Logger Applications” in z/OS MVS Setting Up a Sysplex for more information about how to define authorization to system logger resources. The names of the system logger resources used by SA z/OS are HSA.MESSAGE.LOG and HSA.WORKITEM.HISTORY.

The address spaces of the NetView agents and automation manager need to be authorized to access the log streams. They need update access for the following:

```plaintext
RESOURCE(logstream_name)
CLASS(LOGSTRM)
```

Where `logstream_name` stands for HSA.MESSAGE.LOG and HSA.WORKITEM.HISTORY.

For further information see section “Define Authorization to System Logger Resources” in z/OS MVS Setting Up a Sysplex.

Now activate the couple data sets via the console commands:

```plaintext
SETXCF COUPLE,TYPE=LOGR,PCOUPLE=(primary_couple_data_set)
SETXCF COUPLE,TYPE=LOGR,ACOUPLE=(alternate_couple_data_set)
```

For a sysplex, after defining the new structure in the CFRM policy, activate the CFRM policy via:

```plaintext
SETXCF START,POLICY,TYPE=CFRM,POLNAME=policy_name
```
Step 13: Configure ISPF Dialog Panels

SA z/OS ships two types of ISPF dialogs:

- For defining automation policy: The customization dialog is used to create system operations and processor operations configuration and automation definitions.
- For I/O operations: The I/O operations command panels are used for I/O operations functions.

Both of these ISPF dialogs are invoked using the INGDLG exec. This exec provides parameters for selection of the appropriate dialogs. In addition, this exec can optionally be used to allocate the required dialog libraries. INGDLG should be invoked from an ISPF menu or from a user-defined TSO REXX exec. See Appendix H, “INGDLG Command,” on page 263 for more details.

Because you use the customization dialog to collect information and build control files, you normally need them only at the focal point. However, as the customization dialog allows editing of specific entry types by multiple users, you also need to observe the instructions given in the appendix “Problem Determination” in IBM Tivoli System Automation for z/OS User’s Guide.

The I/O operations dialogs, however, are used to input commands and get responses from the I/O operations part of SA z/OS. Because they do not support multisystem commands for I/O operations functions, you must install them on each system, focal point or target, where you want to use them.

Step 13A: Allocate Libraries for the Dialogs

To set up the dialogs, you must allocate the REXX load libraries and customization dialog load libraries. This section describes the two alternative options available:

- **Alternative 1:** Dynamic allocation of the libraries using the INGDLG exec
- **Alternative 2:** Allocation of the libraries as part of the TSO logon procedure

The recommended way to start the customization dialog is Alternative 1. SA z/OS provides a sample INGEDLG in the SINGSAMP library for this.

Ensure that the ISPF table output library ISPTABL is allocated. The table output data set must also be in the sequence of data sets allocated to ISPTLIB. Furthermore it is recommended that the first data set allocated to ISPTLIB is user-specific. This is guaranteed if INGDLG is called with the default of ALLOCATE(YES). Then the user’s ISPPROF data set is automatically defined as the first data set, and the table output data set is allocated as well. If the first data set allocated to ISPTLIB is not-user specific, multiple users may experience enqueue problems if working with the same PDB concurrently. The reason is that when
Step 13: Configure ISPF Dialog Panels

ISPF opens a table, it requests an enqueue for a resource name that consists of a table name and the first data set allocated to ISPTLIB. For more information, see ISPF User’s Guide.

Remember: Throughout this step use the names of the data sets that you created in “Step 3: Allocate Data Sets for the ISPF Dialog” on page 81.

Alternative 1: Dynamic Allocation using INGDLG
This exec performs allocations prior to starting the dialogs. In order to invoke the exec, you need to be in ISPF. The INGDLG command parameters describe where the data sets are found. See Appendix H, “INGDLG Command,” on page 263 for the use of INGDLG to allocate libraries.

Alternative 2: Allocation of the libraries as part of the TSO Logon Procedure
Create a new TSO logon procedure that has the SA z/OS data sets in the appropriate concatenations.

To create a TSO logon procedure, take an existing one and modify its DD statements to include the following:

```
//ISPLLIB DD ...
DD DSN=ING.SINGIPNL,DISP=SHR
DD ...

//ISPLLIB DD ...
DD DSN=ING.SINGIMSG,DISP=SHR
DD ...

//ISPSLIB DD ...
DD DSN=ING.SINGISKL,DISP=SHR
DD ...

//ISPTLIB DD ...
DD DSN=ING.CUSTOM.AOFTABL,DISP=SHR
DD DSN=ING.SINGITBL,DISP=SHR
DD ...

//ISPLLIB DD ...
DD DSN=ING.SINGMOD1,DISP=SHR
DD ...

//SYSPROC DD ...
DD DSN=ING.SINGIREX,DISP=SHR
DD ...

//AOFTABL DD DSN=ING.CUSTOM.AOFTABL,DISP=SHR 1
//AOFPRINT DD SYSOUT=... 2
//AOFIPDB DD DSN=ING.SINGIPDB,DISP=SHR 3
//IHVCONF DD DSN=ING.CUSTOM.IHVCONF,DISP=SHR 4
```

Notes:
1. Ensure that your ISPF temporary data sets have been allocated with enough space.
   • When a build of the automation control file is performed, each file is written to the temporary data sets before it is copied into the target data set. This can lead to a temporary data set many thousands of lines long. For an enterprise with many applications, there may be several hundred thousand lines written to the temporary data set. These are in the ISPWRK data sets.
Step 13: Configure ISPF Dialog Panels

See z/OS ISPF Planning and Customizing for more information, where it is recommended that you pre-allocate to VIO however, because it reduces overhead and eliminates potential problems from insufficient space.

- The ISPCTL1 temporary data set is used by SA z/OS to temporarily hold file tailoring output and to hold the JCL for batch jobs. See for more information on the ISPCTL1 data set.

2. The ellipses (...) in the DD statements indicate the presence of more information in the JCL: for example, other data sets in a concatenation.

3. User-specific data sets should be placed before the SA z/OS data sets. Generally speaking you need to take care that the concatenation of the SA z/OS data sets does not interfere with the concatenation with data sets from other products.

4. The AOFFTABL DD statement (1) is required to store ISPF tables created when you use the customization dialog. Such tables are used, for example, during pdb import or when the administrator modifies the SA z/OS policy definitions from the SA z/OS customization dialog. This data set is also used to hold the data set definitions for batch processing. This data set was allocated by you in the sample INGEDLGA (see “Step 3: Allocate Data Sets for the ISPF Dialog” on page 81).

5. The AOFFPRINT DD statement (2) is used in place of SYSPRINT for IEBUPDTE, which is invoked when a user of the customization dialog creates a policy database using an SA z/OS-supplied sample as a model. If this DD statement is not allocated, SA z/OS allocates the DD as SYSOUT=H.

   If the IEBUPDTE invocation is successful and SA z/OS dynamically allocated the AOFFPRINT file as SYSOUT=H, the output is purged. If the invocation fails, the output is saved for use in diagnosis of the problem.

   When specifying AOFFPRINT(SYSOUT(Cls)), the output of the dynamically called IEBUPDATE utility is placed in the JES output class Cls. This output is not purged.

6. The AOFIPDB DD statement (3) points to the SA z/OS sample library.

   The AOFIPDB DD statement is required for using best practice policies and for building system operations configuration files.

7. IHVCONF (4), is required for I/O operations. If you are not using I/O operations this DD statement is optional.

8. You should not use any DD names starting with AOF in your logon procedure except those specified in the example above. This is because the SA z/OS customization dialog may dynamically generate AOFxxxxx DD names. Specifically, SA z/OS generates AOFIN and AOFUT2 DD names.

9. I/O operations ISPF dialogs use REXX execs that invoke I/O operations commands and ISPF services. These execs must be made available to the users who want to use the ISPF dialogs. Note that the default record format of the I/O operations REXX target library (whose name is SINGIREX) is FB. The data sets in your SYSPROC concatenation might not be FB. If this is the case, the ALLOCATE command can be used, but you are not able to execute the differently formatted or sized execs. You can do one of the following to correct this:

   a. Copy the contents of the SINGIREX exec library to another data set that is already in your SYSPROC concatenation.

   b. Copy the contents of the SINGIREX exec library to a new data set that has the same characteristics as the other data sets in your SYSPROC concatenation.
Step 13: Configure ISPF Dialog Panels

If you already use a CLIST to allocate your data sets for ISPF, modify it to include the SA z/OS data sets in the appropriate concatenations for users of the customization dialog. If you want to create a CLIST to allocate your data sets you should find out your current allocations for the DD names that need SA z/OS data sets allocated to them. This can be done with the LISTALC STATUS command.

Step 13B: Logging Modifications to Data Set

During APAR apply, a log of the modifications is created and it is written to that data set. If the data set does not exist a dynamic allocation is attempted using a default name. If this name does not fit the installation's naming conventions, or a data set allocation is not allowed at all, this data set should be pre-allocated. Besides the APAR apply, this data set is needed by the report functions which are invoked by the "Report Selection Menu".

**Hint:** The Report Output Data Set is required for APAR apply.

Step 13C: Invoking the ISPF Dialogs

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

The ISPF dialogs are invoked with the INGDLG command. Parameters of this command determine which set of dialogs is invoked (that is, system operations, processor operations, or I/O operations).

Add the command dialogs selections to an ISPF menu panel, such as the ISPF Master Application Menu panel (ISP@MSTR) or the ISPF Primary Menu panel (ISP@PRIM).

**Note:** If you use a customized, non-standard ISPF primary menu panel, modify the definition for that panel instead of ISP@MSTR or ISP@PRIM.

See z/OS ISPF Planning and Customizing for information about customizing ISPF panels. The modified panel should be placed in a data set so that it is used by all users who have the dialog data sets in their concatenation, but it is not used by anyone who does not. You may want to copy it into an enterprise-specific panel data set that you allocate in front of your normal ISPF panel data sets. Figure 9 on page 115 is an example of what a modified panel might look like.
Step 13: Configure ISPF Dialog Panels

The options for the customization dialog and the I/O operations command dialogs must also be added to the panel processing section of the ISPF Application Selection Menu panel as follows. The lines you add are written in italics in the example. You can select the character used to specify the dialogs on your menu.

There are two alternatives to invoke the ISPF dialog:

- "Using INGDLG." This is the recommended method.
- "Using TSO Logon or Your own Automation Procedure."

**Using INGDLG**

If you let INGDLG, described in Appendix H, “INGDLG Command,” on page 263, allocate the data sets dynamically prior to starting the dialogs, the following is a sample definition to be added to the ISPF processing section:

```plaintext
C,'CMD( EXEC 'ING.SINGIREX(INGDLG)' +
   'HLQ(MYHLQ) +
   AOFTABL(ING.CUSTOM.AOFTABL) +
   SELECT(ADMIN)' )'
I,'CMD( EXEC 'ING.SINGIREX(INGDLG)' +
   'HLQ(MYHLQ) +
   IHVCONF(ING.CUSTOM.IHVCONF) +
   SELECT(IOCONNECT)' )'
```

Alternatively, you can invoke the dialogs using TSO REXX execs:

```plaintext
/* REXX ADMIN */
ADDRESS ISPEXEC "SELECT CMD( EXEC 'ING.SINGIREX(INGDLG)' ,
   'HLQ(ING)' ,
   "AOFTABL(ING.CUSTOM.AOFTABL) " ,
   "SELECT(ADMIN) '""
/* REXX IOCONNECT */
ADDRESS ISPEXEC "SELECT CMD( EXEC 'ING.SINGIREX(INGDLG)' ,
   'HLQ(ING)' ,
   "IHVCONF(ING.CUSTOM.IHVCONF) " ,
   "SELECT(IOCONNECT) '""
```

A sample member called INGEDLG is provided in SINGSAMP sample library for invocation of INGDLG with data set allocation done by INGDLG.

**Using TSO Logon or Your own Automation Procedure**

This is the example to be followed if you allocated the data sets using the TSO logon procedure or an automation procedure of your own:
Step 13: Configure ISPF Dialog Panels

)PROC
  &ZQ = &Z
  IF (&ZCMD ^= ' ')
  &ZQ = TRUNC(&ZCMD,'.'
  IF (&ZQ = ' ')
    .MSG = ISRU000
  &ZSEL = TRANS(&ZQ
  0,'PANEL(ISPOPTA)'
  ... 
  C,'CMD(INGDLG SELECT(ADMIN) ALLOCATE(NO))'
  I,'CMD(INGDLG SELECT(IOCONNECT) ALLOCATE(NO))'
  T,'PGM(ISPTUTOR) PARM(ISR00000)'
  ... 
  X,'EXIT'
*,'?' )
&ZTRAIL = .TRAIL
)END

Step 13D: Reconvert I/O Operations Panels

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The I/O operations dialog panels are defined using Dialog Tag Language (DTL) for ISPF. Both the source panels and converted panels are provided in the product libraries. If you choose to update the panels, the source panels must then be reconverted.

Step 13E: Verify the ISPF Dialog Installation

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

Logon to TSO using your modified logon procedure or running your data set allocation CLIST.

Access the customization dialog from the ISPF main menu that you defined. On the Customization Dialog Primary Menu that will appear, verify the release in the panel header.

If you run the REXX exec IOCONNECT shown on step section “Using INGDLG” on page 115, the I/O Operations ISPF Main Menu is displayed. You can use the information shown to verify your SA z/OS installation.

Step 14: Verify the Number of available REXX Environments

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

Change the value of the maximum number of available REXX environments to at least 400. The variables to do this are in the sample assembly and linkedit job in SYS1.SAMPLIB(IRXTSMPE). Change the value of the ENTRYNUM= parameter to at least 400. The sample is a user exit, so follow your SMP/E process for handling user exits. See also “Allocation Requirements for REXX Environments” on page 20.
Step 15: Install Function Packages for NetView and TSO

Step 15: Configure Function Packages for TSO

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This step is only required when you intend to use one of the following features:
- the general purpose command receiver
- the syntax checking for automation table overrides
- command INGRCRDX
- the Preloader function of Automated Discovery

Step 15A: Installation of the TSO REXX Function Package INGTXFPG

Add INGTXFPG to the function package table in the appropriate TSO module below. TSO/E provides the following samples in SYS1.SAMPLIB that you can use to code your load modules:

**Table 18. TSO Load Modules for INGTXFPG**

<table>
<thead>
<tr>
<th>Sample name</th>
<th>Load module name</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRXREXX1</td>
<td>( IRXPARMS for MVS)</td>
</tr>
<tr>
<td>IRXREXX2</td>
<td>( IRXTSPRM for TSO/E)</td>
</tr>
<tr>
<td>IRXREXX3</td>
<td>( IRXISPRM for ISPF)</td>
</tr>
</tbody>
</table>

There are various considerations for providing your own parameters modules. For further details, see the chapter “Function Package” of the TSO REXX Reference. The different considerations are based on whether you want to change a parameter value for an environment(s) initialized:
- for ISPF
- for both TSO/E and ISPF sessions
- in a non-TSO/E address space

Select the appropriate sample parameters modules, for example **IRXREXX2 for TSO/E and batch PGM=IKJEFT01** and make the highlighted and underlined changes similar to the example both:

```assembly
PACKTB_SYSTEM_FIRST DC A(PACKTB_ENTRIES) /* Address of the first*/
PACKTB_SYSTEM_TOTAL DC F'3' /* Total number of */
PACKTB_SYSTEM_USED DC F'3' /* Number of System */
PACKTB_LENGTH DC F'8' /* entries in use */
PACKTB_FFFF DC X'FFFFFFFFFFFFF' /* Length of each PACKTB entry */
PACKTB_ENTRIES EQU * /* System Package Table entries */
PACKTB_ENTRY_MVS EQU * /* The MVS-PACKTB */
PACKTB_NAME_MVS DC CL8 'IRXEFMVS' /* 1. Set function package name */
PACKTB_NAME_MVS DS 0C /* Point to the next entry */
PACKTB_ENTRY_TSO EQU * /* The TSO PACKTB entry */
PACKTB_NAME_TSO DC CL8 'IRXEFPC' /* 2. Set function package name */
PACKTB_NEXT_TSO DS 0C /* Point to the next entry */
PACKTB_ENTRY_SAM EQU * /* The SAM PACKTB entry */
PACKTB_NAME_SAM DC CL8 'INGTXFPG' /* 3. Set SA function package */
PACKTB_NEXT_SAM DS 0C /* Point to next entry */
```
Step 15: Install Function Packages for NetView and TSO

1. Link-edit the REXX default parameters module with the corresponding names. For example, the load module for the sample IRXREXX2 must have the name IRXTSPRM.
2. Place the resultant REXX default parameter module in the LPALST.
3. Make sure that the function package INGTXFPG resides in the LinkList.

Step 15B: Install SA Provided Authorized TSO Command INGPAUTH

System Automation delivers the authorized TSO command INGPAUTH. The Relational Data Services require that the TSO command INGPAUTH must be defined as an authorized command in TSO. This can be achieved by adding the command name to the PARMLIB member IKJTSOxx in SYS1.PARMLIB under AUTHCMD.

Use the TSO/E command PARMLIB UPDATE(xx), or MVS command SET IKJTSO=xx, to activate the new settings. Be sure that INGPAUTH is concatenated in the LINKLIST.

Refer to "Accessing authorized TSO command INGPAUTH" on page 177 and complete further SAF relevant actions that secure the infrastructure appropriately.

Step 16: Configure Alert Notification for SA z/OS

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This section describes the configuration steps that are required for alert notification by SA z/OS.

In order to use alert notification the following must apply to the affected resource in your automation policy:

1. The inform list of the resource must contain at least one of the following communication methods (it can also be defaulted or inherited):
   - IOM: via the IBM Tivoli System Automation for Integrated Operations Management (SA IOM) peer-to-peer protocol
   - EIF: via a Tivoli Event Integration Facility (EIF) event
   - TTT: via XML
   - USR: via a user-defined alert handler
2. Codes must be present on the reserved message ID, INGALERT, that are suitable for the chosen communication methods.

For full details about the installation of related workstation components, refer to Chapter 12, "Configuring SA z/OS Workstation Components," on page 197.

Additionally for further information, see IBM Tivoli System Automation for z/OS Defining Automation Policy and IBM Tivoli System Automation for z/OS Customizing and Programming.

Furthermore, for each system that is able to trigger an alert (that is, to issue an INGALERT command), the ALERTMODE parameter must be set to the chosen communication methods with the INGCNTL command, for example:

INGCNTL SET ALERTMODE='IOM EIF TTT USR'
Step 16: Configure Alert Notification for SA z/OS

You can also use the following command to set alerting for all available communication methods:

```
INGCNTL SET ALERTMODE=ON
```

The available communication methods are:

- IOM: via the SA IOM peer-to-peer protocol
- EIF: via EIF events
- TTT: via XML
- USR: via a user-defined alert handler

For more details about INGCNTL, see IBM Tivoli System Automation for z/OS Programmer’s Reference.

Depending on the chosen communication methods, additional customization is required. This is described in the following sections. Note that you can combine the INGCNTL calls shown in this section in one single invocation.

**Enabling Alert Notification via SA IOM Peer-To-Peer Protocol**

On each system that can connect to an SA IOM server you must set the host name and port number with INGCNTL, for example:

```
INGCNTL SET ALERTHOST=IOMSRV1:1040
```

For more details about INGCNTL, see IBM Tivoli System Automation for z/OS Programmer’s Reference.

**Enabling Alert Notification via EIF Events**

Alert notification uses the message adapter service of the event/automation service (EAS) component of NetView to create EIF events and to integrate SA z/OS and products such as:

- IBM Tivoli Enterprise Console (TEC)
- IBM Tivoli Netcool® OMNibus (OMNibus)

On each system that is able to send EIF events you must set the PPI receiver name of the EAS with INGCNTL, for example:

```
INGCNTL SET EIFPP1=INGEVOMN
```

For more details about INGCNTL, see IBM Tivoli System Automation for z/OS Programmer’s Reference.

**Starting the Event/Automation Service**

The E/AS and the steps to enable it are described in the chapter, “Setting Up UNIX System Services for the NetView Program” in IBM Tivoli NetView for z/OS Installation: Configuring Additional Components. The following section only provides additional information about how to enable the NetView message adapter service of EAS for alert notification.

The E/AS can be started either with a job from an MVS system console, or from a UNIX System Service command shell. In either case, startup parameters must be provided in the form of the following initialization files:

- Global initialization file (Default: IHSAINIT)
- Message adapter configuration file (Default: IHSAMCFG)

A sample for starting E/AS as a job is located in NETVIEW.SCNMUXMS for NetView V5R3 and below or NETVIEW.CNMSAMP for NetView V5R4 and above.
as the member IHSAEVNT. The initialization files are assumed to be located in a
data set that is allocated to the DD name IHSSMP3. Perform the following updates
to the sample to meet the requirements of your installation:
1. If you do not use the default name IHSAINIT for the global initialization file,
   pass the name of your file via the parameter INITFILE.
2. If you do not use the default name IHSAMCFG for the message adapter
   configuration file, pass the name of your file via the parameter MSGCFG.
3. In the DD statement, specify the data set names of your installation.

**Configuring the Global Initialization File**

1. Make sure that the NetView message adapter service is also started when you
   start the EAS. This is done by commenting out the following statement:

   ```
   NOSTART TASK=MESSAGEA
   ```

   The other services are not needed by alert notification, so prevent them from
   starting.

2. Specify INGEVOMN, or any other name of the PPI receiver ID, in the following
   statement:

   ```
   PPI=INGEVOMN
   ```

   You can also pass the PPI receiver ID as a parameter when starting EAS. Make
   sure that you define the same name you specified with INGCNTL.

**Configuring the NetView Message Adapter Service**

Configuration of the NetView message adapter service is done in the message
adapter configuration file, as follows:

1. Provide the IP address or host name and, optionally, the port address of the
   event receiver. This can be virtually any kind of server that can handle EIF
   events but SA z/OS supplies integration with:

   - IBM Tivoli Enterprise Console (TEC)
   - IBM Tivoli Netcool OMNIbus (OMNIbus)

2. Specify the name of the NetView message adapter format file. The version of
   this file that is to be used by alert notification is delivered in
   ING.SINGSAMP(INGMFMTO). If this is in its own data set, copy it to a data
   set that is concatenated to IHSSMP3.

**Enabling Alert Notification via XML**

Alert notification can help with creating trouble tickets automatically. Thus
SA z/OS collects details about the failed resource and stores it in a details data set.
It also creates XML data with overview information.

You must use the INGCNTL command to set the host name and port number to
send the XML data to on each system that is able to create trouble ticket
information, for example:

```
INGCNTL SET TTTHOST=TDISRV1:8000
```

You must also specify allocation data for the details data set, for example:

```
INGCNTL SET TTTDATA='ING.TTT.DATA 1 1'
```

For more details about INGCNTL, see *IBM Tivoli System Automation for z/OS
Programmer’s Reference*. 
**Enabling Alert Notification via User-Defined Alert Handler**

SA z/OS allows you to handle an alert in any other way that you choose.

On each system that is able to run a user-defined alert handler you must specify the command to be executed with INGCNTL, for example:

`INGCNTL SET USRHANDLER=MYHANDLER`

For more details about INGCNTL, see *IBM Tivoli System Automation for z/OS Programmer’s Reference*.

For details about the parameters that are passed and the return codes, see the sample handler delivered in ING.SINGSAMP(AOFEXALT).

### Step 17: Compile SA z/OS REXX Procedures

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

You should perform this step to gain considerable performance improvement for system operations startup.

You can optionally compile the SA z/OS automation procedures, which are written in REXX. The decision to compile the SA z/OS automation procedures implies an added responsibility for recompiling whenever ING.SINGNREX members are affected by SMP/E maintenance. To compile and execute these automation procedures, the IBM Compiler and Library for REXX/370 must be installed on your system along with their prerequisite products.

The JCL job INGEREXR and related routine INGEREXC are provided in the SA z/OS sample library to help you compile the ING.SINGNREX members. Modify the data set names and jobcard in INGEREXR as necessary and submit the job. The ING.SINGNREX.CREXX library can be modelled on ING.SINGNREX, and ING.SINGNREX.LIST should be a VBA LRECL 125 PDS library. If necessary add to the SYSEXEC DD statement the library where the REXXC program can be found. Finally, specify the name of the resulting compiled REXX data set in your NetView application startup procedure.

Consult the *REXX/370 User’s Guide and Reference R3* (SH19-8160) for the compiler options that apply to your installation. If necessary, change the INGEREXC routine accordingly.

**Notes:**
1. A compiler return code of 4 can be expected and is acceptable.
2. SA z/OS has not been tested to run with the REXX Alternate Library. Officially, this is not a supported environment.
3. The NOTESTHALT compiler option should not be used when compiling System Automation REXX.

### Step 18: Defining Automation Policy

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️</td>
<td>✔️</td>
<td></td>
</tr>
</tbody>
</table>
Step 18: Defining Automation Policy

Before you can start using automation, you need to define your automation policy using the customization dialog.

If you start from scratch:
1. Use the IBM best practice policies that are delivered with SA z/OS, *BASE and any others as required, and create your new policy database. Read the information in the section “Creating a New Policy Database” in IBM Tivoli System Automation for z/OS Defining Automation Policy. There is also an application automated discovery function that generates a simple automation policy based on a snapshot of all applications that were active at the time of the discovery. Using that tool in combination with the best practice policies may help you to get your policy customized faster. Refer to the chapter “Automated System Resource Discovery” in IBM Tivoli System Automation for z/OS Customizing and Programming.

2. Next adjust and extend your automation policy. Start by working with the following policy objects:
   - Applications
   - Application groups
   - Monitor Resources
   - Processors
   - Systems
   - A group for each sysplex

You can find detailed information about how to perform these steps in IBM Tivoli System Automation for z/OS Defining Automation Policy, which provides information on using the customization dialog for the required definitions.

If you already have a policy database, make a copy or backup, then complete the following steps.

Step 18A: Build the Control Files

IBM recommends that you use the SA z/OS best practice sample policies to define your SA z/OS components. When you have defined the policies for the SA z/OS components, use the BUILD command to create the configuration files. The BUILD command is available from various panels of the customization dialog. For more information about how to perform this step, refer to IBM Tivoli System Automation for z/OS Defining Automation Policy. You can use the sample job INGEBBLD in the SINGSAMP sample library to create the configuration files in batch.

Note: It is mandatory to use the SA z/OS customization dialog to create policy objects for the resources you want to automate. Do not edit the automation configuration files manually. A manually edited automation control file may damage your automation.

Step 18B: Distribute System Operations Configuration Files

You need to make the configuration files available to the automation agents and automation managers on the target systems. All automation managers and automation agents in the same sysplex must have access to the same system operations control files or a copy of them. You must send the files to the target sysplexes and make the data available to the automation agents and the automation managers.
For the automation managers it can either be placed in the automation managers’ current configuration data set or the automation managers can be told to use a new configuration data set.

**Step 19: Define Host-to-Host Communications**

<table>
<thead>
<tr>
<th></th>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

VTAM definitions are required for both host-to-host communications and host-to-workstation communications. This section of the installation addresses the host-to-host communications.

Verify that your NetView APPL member is consistent with the steps that follow.

The host-to-host communications require:
- Defining each host as a CDRM
- Defining the host ACB

**Step 19A: Configure VTAM Connectivity**

<table>
<thead>
<tr>
<th></th>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The configuration of VTAM bases on Mode table and Major Nodes.

Consult the description of member INGEMTAB that resides in the SINGSAMP sample library. INGEMTAB generates appropriate VTAM mode tables for the NetView application. Incorporate the resulting mode tables into your SYS1.VTAMLIB concatenation of your active VTAM startup procedure.

SA z/OS provides a sample major node member INGENET that resides in the SINGSAMP sample library. Adapt and rename INGENET according to your needs and incorporate it into your SYS1.VTAMLST concatenation of your active VTAM startup procedure.

**Notes:**

1. SA z/OS uses the NetView BGNSESS command with the parameter SRCLU=* to create terminal access facility (TAF) fullscreen sessions for communication with OMEGAMON monitors, if requested. It is expected that OMEGAMON is installed and has been configured for VTAM. See *Tivoli NetView for z/OS Installation: Configuring Additional Components* and *z/OS Communications Server: SNA Network Implementation Guide* for more details.

2. The NetView primary program operator interface task (PPT) is defined as AUTH=(NVSPACE,SPO). This causes unsolicited VTAM messages to be broadcast on the SSI and thus available to NetView. If however you have another NetView, defined as a primary program operator application program (PPO), it receives unsolicited first and messages do not reach the NetView that is defined as a secondary program operator application program (SPO). See *Tivoli NetView for z/OS Installation and Administration* for information on PPO and SPO definitions.
Step 19: Define Host-to-Host Communications

Step 19B: Perform VTAM Definitions for I/O Operations

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

Note: This applies to I/O operations host-to-host communications only. If you have configured a prior level of ESCON Manager or I/O operations, these definitions remain the same.

To use VTAM for I/O operations, there are some definitions that VTAM requires. These definitions are in addition to those needed for the installation and running of VTAM. If you already have VTAM installed, some of these definitions may already exist.

The I/O operations program in each host that carries on this communication must be defined as a VTAM application in each host. The I/O operations program that it communicates with in another host must be defined as a cross domain resource unless you use APPN. I/O operations uses the LU 0 protocol for the communication between hosts.

Note: Refer to the COMM setting in "Step 36B: Prepare I/O Operations Startup" on page 157 for the restriction of the VTAM applid.

Because the means of the I/O operations program may be a channel-to-channel adapter, this connection has to be defined to VTAM via VTAM definition statements.

If the alternate path used is via a network communications program (NCP), the NCP must be defined to VTAM.

In order for VTAM to choose what routes to use for this communication and what priorities to assign, PATH statements and CLASS OF SERVICE must be defined.

An example of some of these VTAM definition statements is shown in Figure 10.

![Figure 10. VTAM Definition Statements](image)

In this example, there are two hosts running I/O operations. One application is named IHVAPPL1 and is in subarea 10. The second application is named IHVAPPL2 and is in subarea 20. Each host has its own set of VTAM definition statements.

Cross-domain definitions

<table>
<thead>
<tr>
<th>V10M</th>
<th>V20M</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTAMA VBUILD TYPE=CDRM</td>
<td>VTAMB VBUILD TYPE=CDRM</td>
</tr>
<tr>
<td>V10M CDRM SUBAREA=10</td>
<td>V10M CDRM SUBAREA=10</td>
</tr>
<tr>
<td>V20M CDRM SUBAREA=20</td>
<td>V20M CDRM SUBAREA=20</td>
</tr>
</tbody>
</table>
Step 19: Define Host-to-Host Communications

The appropriate definitions are needed for each host that will be communicating via I/O operations. Each host will be defined as a CDRM.

If a communication path between the hosts is a channel-to-channel adapter, this has to be defined to VTAM.

Note: Change each “x” to the appropriate value.

```vtam
CTCV20 VBUILD TYPE=CA
label1 GROUP LNCTL=CTCA,
   DELAY=x,  (cause link to INOP if SIO timeout occurs)
   MH=., (channel unit address of channel to channel adapter)
   MAXFRU=x, (channel unit address of channel to channel adapter)
   MIH=x,  (channel unit address of channel to channel adapter)
   REPLYTO=x (tells VTAM how long to wait for completion after
   channel program started)
label2 LINE ADDRESS=x, (channel unit address of channel to channel adapter)
  MAXBFRU=x (channel unit address of channel to channel adapter)
label3 PU PUTYPE=4,
  TG=1
```

Each I/O operations program must be defined via an application statement in each host. The user-specified names must be unique in the network. These are the names that the other I/O operations hosts will know each I/O operations by.

The ACBNAME parameter is required for I/O operations. This name must be IHVISC, and must be reserved for this use only.

The parameters SONSCIP=YES and AUTH=ACQ must also be specified.

For I/O operations it is strongly recommended that the DLOGMOD and MODETAB parameters given in the example below, or equivalent definitions, should be used. Note that an RUSIZE of ‘zero’ is used with this LU TYPE 0 protocol.

```vtam
VBUILD TYPE=APPL
IHVAPPL1 APPL ACBNAME=IHVISC, AUTH=ACQ,
   DLOGMOD=INTERACT, SONSCIP=YES, MODETAB=ISTINCLM
IHVAPPL2 APPL ACBNAME=IHVISC, AUTH=ACQ,
   DLOGMOD=INTERACT, SONSCIP=YES, MODETAB=ISTINCLM
```

Using the above VTAM definitions the LOGMODE table entry would be:

```vtam
IBM3767 MODEENT LOGMODE=INTERACT,
   FM PROF=X'03',
   TS PROF=X'03',
   PRIPRO T=X'B1',
   SECPRO T=X'A0',
   COMPRO T=X'3040'
```

Each host must have a cross-domain definition for the other I/O operations host applications. They are defined as cross domain resources, as follows:

```vtam
VBUILD TYPE=CDRSC
IHVAPPL2 CDRSC CDRM=V20M
IHVAPPL1 CDRSC CDRM=V10M
```

The communication paths between the I/O operations hosts must be defined, as follows:

```vtam
PATH DESTSA=20,
   ER0=(20,1),
   ER1=(20,1),
   VR0=1,
   VR1=0
PATH DESTSA=10,
   ER0=(10,1),
   ER1=(10,1),
   VR0=1,
   VR1=0
```
Step 19: Define Host-to-Host Communications

The class of service (COS) definition is:

\[
\begin{align*}
\text{ISTSDCOS COSTAB} & : \\
\text{IHVAPPL1 COS VR=((0,2),(1,2))} & : \\
\text{COSEND} & :
\end{align*}
\]

\[
\begin{align*}
\text{ISTSDCOS COSTAB} & : \\
\text{IHVAPPL2 COS VR=((0,2),(1,2))} & : \\
\text{COSEND} & :
\end{align*}
\]

### APPN Definitions

Assuming that both hosts reside in the same domain, XYZ, the equivalent APPN definitions are:

<table>
<thead>
<tr>
<th>VTAM A</th>
<th>VTAM B</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>APPN Transport Resource List</strong></td>
<td><strong>APPN Transport Resource List</strong></td>
</tr>
<tr>
<td>VBUILD TYPE=LOCAL</td>
<td>VBUILD TYPE=LOCAL</td>
</tr>
<tr>
<td>XYZANTRL PU TRLE=XYZATRLN, CONNTYPE=APPN, CPCP=YES</td>
<td>XYZBNTRL PU TRLE=XYZBTRLN, CONNTYPE=APPN, CPCP=YES</td>
</tr>
<tr>
<td><strong>Channel to Channel Adapter</strong></td>
<td><strong>Channel to Channel Adapter</strong></td>
</tr>
<tr>
<td>VBUILD TYPE=TRL</td>
<td>VBUILD TYPE=TRL</td>
</tr>
<tr>
<td>XYZATRLN TRLE LNCTL=MPC, READ=(cua), WRITE=(cua), MPCLEVEL=NOHPDT</td>
<td>XYZBTRLN TRLE LNCTL=MPC, READ=(cua), WRITE=(cua), MPCLEVEL=NOHPDT</td>
</tr>
<tr>
<td><strong>I/O operations / VTAM</strong></td>
<td><strong>I/O operations / VTAM</strong></td>
</tr>
<tr>
<td>VBUILD TYPE=APPL</td>
<td>VBUILD TYPE=APPL</td>
</tr>
<tr>
<td>IHVAPPL1 APPL ACBNAME=IHVISC, AUTH=ACQ, DLOGMOD=INTERACT, SONSCIP=YES, MODETAB=ISTINCLM</td>
<td>IHVAPPL2 APPL ACBNAME=IHVISC, AUTH=ACQ, DLOGMOD=INTERACT, SONSCIP=YES, MODETAB=ISTINCLM</td>
</tr>
</tbody>
</table>

For details of the channel unit address (CUA) refer to the section “Operand descriptions” in the chapter “Transport resource list major node” in *z/OS Communications Server: SNA Resource Definition Reference*.

The APPL-specific definitions are identical to those described in “Cross-domain definitions” on page 124.

### Step 19C: Perform TCP/IP Definitions for I/O Operations

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>✔️</td>
</tr>
</tbody>
</table>

**Note:** This applies to I/O operations host-to-host communications only.

In order to use TCP/IP for I/O operations, there are some definitions that I/O operations requires. These definitions are in addition to those needed for the installation and running of VTAM.

I/O operations prefers to communicate with other hosts using the TCP protocol when the remote host is running a release level that is the same or higher. I/O operations requires a TCP/IP host name or an alias name that is no more than 8 characters long. This is because this name must be stored in the switch host data.
Step 19: Define Host-to-Host Communications

buffer, which has a limited size. If you have defined longer host names you must
define an alias of up to 8 characters for each host name that exceeds the limit. You
can define alias host names in the local host tables which can be either
HOSTS.LOCAL or IPNODES. I/O operations uses z/OS Communications Server
resolver API services to obtain the host name.

Note: Refer to the COMM setting in "Step 36B: Prepare I/O Operations Startup" on page 157 for the restriction of the TCP/IP host name.

1. Refer to the section “Search orders used in the native MVS environment” in the
z/OS Communications Server IP Configuration Guide for information about how
the host name value is determined and how the base resolver configuration
files, local host tables, and services information are used.

2. For configuring locally defined host names, refer to the section “Configuring
the local host table (optional)” in the z/OS Communications Server IP
Configuration Guide.

3. If you are using locally defined host names, check the TCPIP.DATA LOOKUP
statement of each system that you have defined an alias for. Verify that either
LOCAL is specified or LOCAL precedes DNS, because the DNS server does not
return an alias name. Refer to the section "LOOKUP" in the chapter
"TCPIP.DATA configuration statements” of for more information.

Because each I/O operations program acts as a server as well as a client, it requires
port definitions in the '/etc/services' respectively 'ETC.SERVICES' service
information data set:

IHVsrvr portnumber/tcp
IHVCnt portnumber/tcp

The first entry is mandatory for reestablishing a connection after an interrupt. The
second entry is optional. You have to specify the second entry if you want to
restrict particular ports for the use by I/O operations. For details of controlling
access to ports refer to the section "Port access control” in the chapter “Security” of
z/OS Communications Server: IP Configuration Guide.

Note that the service names require the component code (IHV) to be defined in
uppercase and the remaining characters in lowercase. For details on how the
service names are obtained by TCPIP, see the Services information in the section
"Search orders used in the native MVS environment” in z/OS Communications
Server: IP Configuration Guide. For information on defining service names, see the
section "'/etc/services and ETC.SERVICES port assignments' in z/OS
Communications Server: IP Configuration Reference.

Note: If you omit the definition of the server port, I/O operations suppresses the
TCP/IP communication for its lifetime and falls back to VTAM communication.

Check the MAXSOCKETS parameter in the BPXPRMxx parmlib member's
NETWORK statement that corresponds to the addressing family. This value
determines how many sockets for a particular addressing family can be opened in
the entire system. I/O operations requires twice the number of possible TCP
connections plus one.

The number of sockets that an application can open is also limited by the UNIX
System Services parameter MAXFILEPROC in the BPXPRMxx parmlib member.
This parameter determines the number of sockets each address space can have
open. The same rules apply to this parameter, that is, I/O operations requires twice
the number of possible TCP connections plus one.
**Step 19: Define Host-to-Host Communications**

Check the SOMAXCONN value that defines the maximum number of connection requests queued for the listening socket. I/O operations cannot have more than 256 connection (VTAM and TCP/IP) at a time. However it is very unlikely that all connection requests occur at the same time because it is dependent on the time that each I/O operations is started. The default value of 10 is probably large enough.

**Step 20: Enabling SA z/OS to Restart Automatic Restart Manager**

**Enabled Subsystems**

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If you intend to use the z/OS Automatic Restart Manager and you want to coordinate its actions with those of SA z/OS, you must ensure the following:

- The SA z/OS-supplied element restart exit (ERE) must be available to z/OS. The exit, AOFPERRE, is in the ING.SINGMOD2 data set. No customization is required.
- The AOFARCAT autotask must be created. The autotask name is included in the AOFOPF member and is created automatically by NetView if you install SA z/OS without changing AOFOPF.
- The NetView Subsystem Interface (SSI) must be active for the coordination of SA z/OS and z/OS automatic restart management to occur.
- As part of its Automatic Restart Manager support, SA z/OS claims all PPI receiver IDs starting with AOF. If you have any other PPI receivers named AOFxxxx, results are unpredictable.

For further information on the relationship between SA z/OS and Automatic Restart Manager, see *IBM Tivoli System Automation for z/OS Defining Automation Policy.*

**Step 21: Define Security**

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

Perform this step to ensure that only authorized staff can manage the resources in your environment.

Your operations staff and automation facilities at SA z/OS-controlled systems need to be authorized to manage the resources in their environment. You can control human and automation operator authority through the password security provided by either by NetView or an SAF-based security product, such as RACF.

Refer to the [Chapter 11, “Security and Authorization,” on page 161](#) and complete all SAF relevant actions that secure the infrastructure appropriately.

For a basic security setup consult the following subchapters:

- “Authorization of the Started Procedures” on page 162
- “Roles” on page 163
Step 21: Define Security

- “Operators” on page 165
- “Commands” on page 165
- “Use of Commands Cross System” on page 166
- “Use of Commands from TSO or Batch” on page 167
- “Resources” on page 169

See also the following sections in Chapter 11, “Security and Authorization,” on page 161 for other security options:
- “Other Security Options” on page 172
- “Securing Focal Point Systems and Target Systems” on page 173
- “Granting NetView and the STC-User Access to Data Sets” on page 174
- “Restricting Access to INGPLEX and INGCF Functions” on page 177
- “Restricting Access to Joblog Monitoring Task INGJLM” on page 178
- “Security for IBM Tivoli Monitoring Products” on page 179 (OMEGAMON)
- “Controlling Access to the Processor Hardware Functions” on page 184
- “Defining a RACF Profile for I/O Operations” on page 188
- “Establishing Authorization with Network Security Program” on page 192

Note: To plan your RMTCMD-based INGSEND security, see the discussion of RMTCMD security features in the NetView documentation.

Step 22: Configure the Status Display Facility (SDF)

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>

If you decide to use SDF as the SA z/OS fullscreen operator interface for monitoring automated resource statuses at the NetView 3270 console, configuring SDF involves defining the following:

- SDF initialization parameters. These are defined in the AOFINIT member of a NetView DSIPARM data set.
- Copy and configure member INGPTOP in the ING.SINGNPRM library concatenate it in the DSIPARM data set before the SA z/OS libraries. Configure it the system and sysplex names.
- Define and configure the following variable in the NetView style sheet:

  COMMON.AOF_AAO_SDFROOT_LIST = SYS1 SYS2 SYS3 (deprecated)
  COMMON.AOF_AAO_SDFROOT.0 = 5
  COMMON.AOF_AAO_SDFROOT.1 = SYS1 SYS2
  COMMON.AOF_AAO_SDFROOT.3 = SYS3
  COMMON.AOF_AAO_SDFROOT.4 = SYS4

You may use other panel/tree members than the default members AOPNLS and AOTREE for some or additional names like:

SYS1 SYS2/MYPNLS SYS3&SLASH./MYTREE SYS4/MYPNLS2/MYTREE

The panel member defaults to AOPNLS for SYS3. For SYS2 the tree member defaults to AOTREE. And, for SYS1 both default members are being used.
Step 22: Configure the Status Display Facility (SDF)

Note: NetView interprets two consecutive slashes as the begin of a line comment. For this reason the sample above uses the symbol for the slash character followed by the slash character itself.

- Ensure that the inform list in the customization dialog contains SDF for the resources that you want to monitor (consider using, for example, system and sysplex defaults).
- Color and priority assignments for resource status types. These have default values that are set up by SA z/OS (see IBM Tivoli System Automation for z/OS User's Guide for details), but you can define overrides to color and priority assignments with the SA z/OS customization dialog.
- SDFROOT. You can specify a root name for the SDF tree on the Environment Setup Panel of the customization dialog. If you do not specify a new root name, it defaults to the value specified for SYSNAME.

See IBM Tivoli System Automation for z/OS Customizing and Programming for detailed information about customizing SDF.

Step 23: Check for Required IPL

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

An IPL is only required if:

- In “Step 4A: Update IEAAPFxX” on page 82, you used the IEAAPFxX member to define authorized libraries to the APF
- In “Step 4D: Update LPALSTxx” on page 83 you decided not to use the solution to dynamically add the modules to the LPALST
- In “Step 4E: Update LNKLSTxx” on page 84 you updated LNKLST and you decided not to use the solution to dynamically add the modules to the LNKLST
- “Step 4G: Update IEFSSNxx” on page 84 was required because the IEFSSNxx member was not updated during NetView installation and you cannot use the z/OS command SETSSI for a dynamic update of the subsystem name table.

Step 24: Automate System Operations Startup

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

Sample: INGECOM

Add commands to the COMMNDxx member of SYS1.PARMLIB to start the automation NetView when z/OS starts. You may also need to modify an IIEASYSxx member of SYS1.PARMLIB to specify which COMMNDxx or other PARMLIB members to use during IPL. SA z/OS initialization begins with starting system operations. If an SA z/OS automation policy is used, system operations subsequently starts processor operations and I/O operations.

Make the described changes to the following SYS1.PARMLIB data set members:
Step 24: Automate System Operations Startup

Sample COMMNDxx

Make sure that the procedure names you choose match those specified in the SYS1.PROCLIB data set.

Compare the contents of the COMMNDxx member with the INGECOM member which resides in the SINGSAMP sample library. Edit the COMMNDxx member and do the following:

1. If you want to use the recording of IPL function (INGPLEX IPL command) add the following statement in the COMMNDxx member:
   
   ```
   COM='S HSAPIPLC, SUB=MSTR'
   ```

   This procedure collects the IPL information in MVS. Return codes for this procedure are documented in the HSAPIPLC sample.

2. If you are running more than one NetView on your system, ensure that you have included start commands for the Automation NetView.
   
   ```
   COM='S CNMSJ010, JOBNAME=SYSVSSI, SUB=MSTR'
   COM='S INGENVSA, JOBNAME=SYSVAPPL, SUB=MSTR'
   ```

   Note:

   CNMSJ010 is the name of the sample that is provided by NetView that you copied in “Step 5: Configure SYS1.PROCLIB Members” on page 86.

   INGENVSA is the name of the sample that is provided by SA z/OS that you copied in “Step 5: Configure SYS1.PROCLIB Members” on page 86.

3. Adapt the NetView Application and NetView Subsystem Interface jobname to agree with the four-character prefix defined in the IEFSSNxx member, which is described in “Step 4G: Update IEFSSNxx” on page 84. For example, if the name of the NetView Application jobname is SYSVxx, SYSV must be specified in the IEFSSnxx member as the character prefix.

Sample IEASYSxx

Edit the IEASYSxx member to specify which SYS1.PARMLIB data set members to use during the IPL process. This is done by specifying the 2-character suffix of the SYS1.PARMLIB member names. If you choose SO, the statements in the IEASYSxx member would be as follows:

- APF=SO
- CMD=SO
- CON=SO
- SSN=SO
- SCH=SO
- LNK=SO
- LPA=SO

For example, because APF=SO, the system uses the IEAAPFSO member during the IPL process.

How to Automate the Automation Manager Startup

Note: The system that the automation manager should be started on must be defined as policy object System in the policy database which will be used to create the automation manager configuration file that this automation manager uses (see also “Step 18A: Build the Control Files” on page 122).
**Step 24: Automate System Operations Startup**

To enable automatic startup of the automation manager whenever SA z/OS is started, add the following start command for the automation manager to the COMMNDxx PARMLIB member:

```
S INGEAMSA,JOBNAME=AM,SUB=MSTR
```

You can find the sample startup procedure called INGEAMSA in the SINGSAMP sample library.

**Step 25: Verify Automatic System Operations Startup**

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

After you have installed the host components of SA z/OS, it is recommended that you perform the following steps for verification purposes:

1. **Perform an IPL**, if you have not done this according to “Step 23: Check for Required IPL” on page 130. Then start SA z/OS. The following messages should appear on the system console:

   AOF532I hh:mm:ss AUTOMATION ENVIRONMENT HAS BEEN INITIALIZED
   AOF540I hh:mm:ss INITIALIZATION RELATED PROCESSING HAS BEEN COMPLETED

2. Use the NetView LIST command to confirm that the following SA z/OS tasks are active:

<table>
<thead>
<tr>
<th>Task Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOPTSTS</td>
<td>automation status file task</td>
</tr>
<tr>
<td>INGPXDST</td>
<td>XCF communication task</td>
</tr>
</tbody>
</table>

To confirm that these tasks are active, log on to NetView and enter the NetView LIST command to display the status for each task:

```
LIST taskname
```

3. Use the commands INGAMS and INGLIST to verify that they work.

4. Check that the subsystem status and automation flag settings are what you expect. Enter the DISPSTAT ALL command to display the status of automated subsystems and the DISPFLGS command to display the automation flag settings: See *IBM Tivoli System Automation for z/OS Operator’s Commands* for information about these commands.

5. Use the SA z/OS DISPAUTO command in NetView to display a menu that allows you to initiate further command dialogs. These display information about your automation. Enter DISPAUTO and then choose one of the menu options. See *IBM Tivoli System Automation for z/OS Operator’s Commands* for information about the DISPAUTO command.

6. Confirm that the automation shuts down and restarts the subsystems as you expect. You can shutdown and restart each automated resource individually using the following SA z/OS command:

   ```
   INGREQ resource REQ=STOP SCOPE=ONLY RESTART=YES
   ```

   If any of the resources (subsystems) do not restart as you expect, make corrections to your automation policy.
Step 26: Configure USS Automation

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Step 26A: Securing USS Resources

When setting up USS automation then SAF related actions are required. Therefore the user IDs:

- must have an OMVS segment
- must be permitted to the appropriate SAF profiles.

Refer to the chapter [Chapter 11, “Security and Authorization,” on page 161](#) to generate a security definition member INGESAF by using the configuration assistant.

Each user ID definition in INGESAF contains an OMVS segment (ADDUSER/ALTUSER). The section Set OMVS Security within this member lists the USS SAF profiles.

Step 26B: Preparing for USS Automation

Use the common global variable, AOFUSSWAIT, that you can set in your startup exit, to change the way SA z/OS behaves. This variable should be set only once for an SA z/OS system.

AOFUSSWAIT is the time that SA z/OS waits for the completion of a user-specified z/OS UNIX monitoring routine (defined in the z/OS UNIX Control Specification panel) until it gets a timeout. When the timeout occurs, SA z/OS does no longer wait for a response from the monitoring routine and sends a SIGKILL to the monitoring routine.

Step 27: Enabling the System Automation WebServices Adapter

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: The description of this step assumes that you use the /bin/sh shell within USS.

The System Automation WebServices Adapter is installed by default through SMP/E into the zFS directory of the E2E adapter. Also the customization directory might be the same as for the E2E adapter. Therefore it is assumed that there is:

- a zFS installation path, read only. Default is /usr/lpp/ing/adapter
- a zFS customization path, read/write

The following files reside in the customized path:

<table>
<thead>
<tr>
<th>Customization-path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/customization-path/config</td>
<td>Configuration file inghttps.properties used by the JCL procedure of the started task via DD STDENV.</td>
</tr>
<tr>
<td>inghttps.properties</td>
<td></td>
</tr>
</tbody>
</table>
Step 27: Enabling the System Automation WebServices Adapter

<table>
<thead>
<tr>
<th>Customization-path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/customization-path/data</td>
<td>Standard error and output defined by the JCL procedure of the started task.</td>
</tr>
<tr>
<td>stderr.fwnn.txt</td>
<td></td>
</tr>
<tr>
<td>stdout.fwnn.txt</td>
<td></td>
</tr>
<tr>
<td>/customization-path/data/ing/logs</td>
<td>Message and trace log files within the output directory. Flat files and files in XML format.</td>
</tr>
<tr>
<td>ingnnMsgFlat.log</td>
<td></td>
</tr>
<tr>
<td>ingnnMsgXml.log</td>
<td></td>
</tr>
<tr>
<td>ingnnTraceFlat.log</td>
<td></td>
</tr>
<tr>
<td>ingnnTraceXml.log</td>
<td></td>
</tr>
<tr>
<td>/customization-path/ssl</td>
<td>Keystore file used for secure HTTP connections to the clients.</td>
</tr>
<tr>
<td>inghttps.jks</td>
<td></td>
</tr>
</tbody>
</table>

Step 27A: Copy and Configure the Start Procedure

Copy the sample JCL procedure INGXJAVA from ING.SINGSAMP into your proclib. Rename it to a name of your choice and configure it. Follow the comments in the sample. Specify the installation and customization path in parameter DIRI and DIRC. For example, you may start the WebServices Adapter via start-command:

S INGXJAVA, JOBNAME=WEBADPT, XID=34, RID=00, PORT=9443

The JCL procedure supports the following input parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>XID</td>
<td>2 byte XCF group id of the associated System Automation NetView Agent</td>
</tr>
<tr>
<td>RID</td>
<td>2 byte suffix used to build the PPI receiver id INGJVMnn</td>
</tr>
<tr>
<td>PORT</td>
<td>Port used by HTTP adapter</td>
</tr>
</tbody>
</table>

This gives you more flexibility while defining the corresponding started task as an automated application using the System Automation policy.

Step 27B: Copy and Adapt the Configuration File

Copy the sample configuration file /usr/lpp/ing/adapter/config/inghttps.properties into your customization directory and configure it to your needs. Specify the configuration file in the start procedure DD SDTENV. The following parameters may be configured.

<table>
<thead>
<tr>
<th>Name</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INGHTTP_XID</td>
<td>empty string</td>
<td>XCF group id of associated System Automation / NetView agent</td>
</tr>
<tr>
<td>INGHTTP_RID</td>
<td>00</td>
<td>2-byte suffix for PPI receiver id of the HTTP adapter which becomes INGJVM00.</td>
</tr>
<tr>
<td>INGHTTP_PORT</td>
<td>9443</td>
<td>HTTPS port number</td>
</tr>
</tbody>
</table>
### Step 27: Enabling the System Automation WebServices Adapter

<table>
<thead>
<tr>
<th>Name</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INGHTTP_HOST</td>
<td>automatic</td>
<td>IP address or host name of the HTTP Adapter. Default is automatically determined by USS command hostname.</td>
</tr>
<tr>
<td>INGHTTP_INSTALL_ROOT</td>
<td>/usr/lpp/ing/adapter</td>
<td>This is the root directory of the installation directory.</td>
</tr>
<tr>
<td>INGHTTP_CUSTOM_ROOT</td>
<td>/usr/lpp/ing/adapter</td>
<td>The root directory of the customization directory. It contains logger output files, configuration file and SSL related files.</td>
</tr>
<tr>
<td>INGHTTP_SSL_FNAME</td>
<td>inghttps</td>
<td>The name of the keystore file which must be a Java keystore file, for example inghttps.jks. The file extension jks will be appended automatically. It is required that the keystore file INGHTTP_CUSTOM_ROOT/ssl/inghttps.jks does exist and contains exactly one certificate. This certificate will be passed as server certificate to the HTTP clients.</td>
</tr>
<tr>
<td>INGHTTP_SSL_PASSW</td>
<td>password</td>
<td>Password of the keystore file. You may specify an encrypted password. For how to generate an encrypted password see &quot;Step 27D: Encrypt the Password of the Keystore File&quot; on page 136.</td>
</tr>
<tr>
<td>INGHTTP_REQUEST_MAX</td>
<td>9</td>
<td>Maximum number of HTTP requests that can be processed concurrently.</td>
</tr>
<tr>
<td>INGHTTP_SESSION_TIMEOUT</td>
<td>600</td>
<td>Number of seconds until a HTTP idle session is deleted. A new session requires authentication of userid and password again. If you specify 0 then the session never times out.</td>
</tr>
<tr>
<td>INGHTTP_PPI_TIMEOUT</td>
<td>30</td>
<td>Maximum number of seconds to wait for response from NetView via PPI.</td>
</tr>
<tr>
<td>JAVA_HOME</td>
<td></td>
<td>Must contain the JAVA installation path. It is assumed that Java is installed into the following directories:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$JAVA_HOME/lib</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$JAVA_HOME/bin</td>
</tr>
<tr>
<td></td>
<td></td>
<td>It is required to use the 32-bit version of Java 1.6.</td>
</tr>
<tr>
<td>INGHTTP_JAVA_XMX</td>
<td>128</td>
<td>Maximum heap size of Java process</td>
</tr>
<tr>
<td>INGHTTP_CODEPAGE</td>
<td>cp1047</td>
<td>Codepage for ASCII EBDIC translation</td>
</tr>
</tbody>
</table>

### Step 27C: Copy the Keystore File

Create a keystore file with the server certificate of your choice. The keystore file must be a Java keystore file with the file extension .jks. Copy the keystore file into the SSL directory:
Step 27: Enabling the System Automation WebServices Adapter

/ customization-path/ssl/inghttps.jks

For test purposes you may create a self-certified certificate.

This step explains how to create a keystore file with a self-certified server certificate.

**How to generate Self-Certified Certificate and Keystore**

Ensure that you copied the configuration file inghttps.properties into your customization directory and customized it to your needs.

The environment variables from the configuration file must be exported by calling the shell script ingexport.sh. This script is located in the installation directory as specified in the INGHTTP_INSTALL_ROOT key within inghttps.properties. Change into this directory and execute the following command:

```
. ./ingexport.sh /customization-path/config/inghttps.properties
```

Note, between the first dot and the ./ingexport.sh there is one blank. The first dot is the source command and may only work within the /bin/sh shell. This allows the export of variables to be used by the script inghttps.sh.

In order to generate a self-certified certificate and the corresponding keystore for test purposes you may invoke the WebServices Adapter script:

```
inghttps.sh -keytool your_password common_name
```

The keystore will be placed in the zFS path /customization-path/ssl/ inghttps.jks. You specify the password for the keystore. You may specify the common name of the certificate which may be used by the client program in order to validate that the server certificate comes from the proper host. If you omit the parameter common_name then by default the hostname is used and automatically extracted from the system where you are currently running.

**Step 27D: Encrypt the Password of the Keystore File**

In order to generate an encrypted password for parameter SSL_PASSW, you must invoke the script inghttps.sh with the option -encrypt. But before, you must change into the installation directory as specified in the HTTP_INSTALL_ROOT key within your inghttps.properties file and export the environment variables of the configuration file. Execute the following command:

```
. ./ingexport.sh /customization-path/config/inghttps.properties
```

Note that between the first dot and ./ingexport.sh there is a one blank. The first dot is the source command and may only work within the /bin/sh shell. This allows the export of variables to be used by the script inghttps.sh.

Now invoke the HTTP Adapter script with encrypt the password:

```
./inghttps.sh -encrypt your_password
```

Copy the encrypted password from the command output and paste it into the variable SSL_PASSW, as described in "Step 27E: Update the Configuration File again" on page 137.
Step 27: Enabling the System Automation WebServices Adapter

Step 27E: Update the Configuration File again

There are two variables within the configuration file where you must specify the keystore file name and password:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSL_FNAME=inghttps</td>
<td>Name of the keystore file. The file extension .jks is assumed.</td>
</tr>
<tr>
<td>SSL_PASSW=sapwd</td>
<td>Password of the keystore. This is either clear text password or an encrypted password.</td>
</tr>
</tbody>
</table>

Step 27F: Configure the WebServices Started Task

The WebServices Adapter runs as a started task. It should be started by the Automation manager when System Automation has finished initialization.

Import the following PDB entries within *IBMCOMP and customize them to your needs:

<table>
<thead>
<tr>
<th>PDB Entry Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEBADTR</td>
<td>APL</td>
</tr>
<tr>
<td>WEBADTR_X</td>
<td>APG, Sysplex server group</td>
</tr>
</tbody>
</table>

WEBADPT_X defaults to a server group with availability target 1 which means that the adapter runs on the primary system only.

According to your needs you may change the server group to:

- primary and backup adapter (set availability target 2 and both base preferences to 700)
- MOVE group (in this case you need to setup a corresponding VIPA)

Step 27G: Set Up Virtual IP Address for the WebServices Started Task

If WEBADPT_X becomes a move group then it is necessary for the client program to connect to the WebServices Adapter via a virtual IP address (VIPA).

A VIPA stays the same, even though the application (WebServices Adapter) using it can be moved from one system to another. There are different possibilities to configure and activate a VIPA.

You may define a dynamic VIPA with a VIPARANGE statement in the TCP/IP configuration. The VIPA is activated automatically upon startup of the WebServices Adapter address space. Upon termination, when the WebServices Adapter closes the socket, the VIPA is also automatically deactivated. With this configuration, the VIPA can be managed by the z/OS Communication Server product alone, without requiring any additional automation support.

For more details refer to the documentation for the z/OS Communication Server.
Step 28: Install an SA z/OS Satellite

This step is only required if your enterprise runs an Automation NetView and a Networking NetView with GMFHS on the focal point system or on another focal point NetView. You must then install SA z/OS on the automation NetView that is used for system automation.

Step 28A: Configure the Networking NetView or Focal Point NetView Startup Procedure

In SYS1.PROCLIB or another procedure library, find members used to start the Networking NetView application. Insert the data set names from the following table into the indicated DD concatenations.

Notes:
1. The data sets listed in Table 19 should appear last in your concatenation. If they appear before other data sets (for example, data sets containing members configured for automated network operations [AON/MVS]), results are unpredictable.
2. The ING.SINGMOD1 library needs to be authorized for APF.

Table 19. Members to Start the Networking NetView

<table>
<thead>
<tr>
<th>DDNAME</th>
<th>System Operations Data Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB</td>
<td>ING.SINGMOD1</td>
</tr>
<tr>
<td>DSICLD</td>
<td>ING.SINGNREX</td>
</tr>
<tr>
<td>DSIPARM</td>
<td>ING.SINGNPRM</td>
</tr>
<tr>
<td>DSIMSG</td>
<td>ING.SINGNMSG</td>
</tr>
<tr>
<td>DSIPRF</td>
<td>ING.SINGNPRF</td>
</tr>
<tr>
<td>CNMPNL1</td>
<td>ING.SINGNPNL</td>
</tr>
</tbody>
</table>

Step 28B: Configure the Networking NetView or Focal Point NetView DSIPARM Data Set

Several members in the DSIPARM concatenation must be configured for the SA z/OS satellite. Before editing an SA z/OS member, remember to copy it from ING.SINGNPRM into a new, user-defined data set that is placed before ING.SINGNPRM in the concatenation.

NetView style sheet

To enable SA z/OS, make sure that the following TOWER statements are activated in the NetView style sheet:
- For a satellite SA z/OS on a Networking NetView:
  TOWER = SA
  TOWER.SA = SATELLITE
- For full SA z/OS:
  TOWER = SA
  TOWER.SA = SYSOPS
Step 28: Install an SA z/OS Satellite

AOFMSGST
If you do not choose to use the NetView operator IDs defined by SA z/OS, copy and edit AOFMSGST to contain the appropriate definitions of the synonyms %AOFOPMSU%, %AOFOPHB% for your Networking NetView. %AOFOPMSU% is a synonym for the operators that can be routed commands as a result of alerts trapped in the NetView automation table. %AOFOPHB% is a synonym for the operator that can be routed heartbeat alerts trapped in the NetView automation table. (Note that there can be only one operator defined for %AOFOPHB% and it must be unique and not used for any other functions). Other synonyms in the member are not specific to the Networking NetView environment.

AOFRODM
Copy and edit AOFRODM to contain the correct name for your RODM and a user ID authorized to update it.
- Specify a RODM name by changing RODMNAME=NONE to RODMNAME=xxxxxxxx, where xxxxxxxx is your RODM name.
- Specify a user ID by changing RODMUSER=XXAOCFR to RODMUSER=xxxxxxxx, where xxxxxxxx is your user ID for batch updates from NetView.

Step 29: Installing and Customizing the NMC Focal Point

<table>
<thead>
<tr>
<th></th>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>*</td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>

Communication between SA z/OS and the NMC focal point 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. This information is available if you have completed the previous installation steps, that is, SA z/OS is fully functional.

There are two possible configurations when setting up the NMC focal point:
1. Full SA z/OS
2. A satellite SA z/OS on a Networking NetView

The following sections describe how to configure the SA z/OS topology manager for the operators.

Step 29A: Preparing for NMC
Some of the tasks in this step are different for full SA z/OS and a satellite SA z/OS, as indicated.

Applications Required by NMC
1. The applications that NMC uses must be available, so make sure one of the following:
   - RODM is running and the RODM load function has loaded the data model into RODM
   - GMFHS and MultiSystem Manager are installed and working.

Note: For information about how to do this step, refer to:

Chapter 10. Traditional SA z/OS Configuration 139
Step 29: Installing and Customizing the NMC Focal Point

- Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer’s Guide
- Tivoli NetView for z/OS Graphic Monitor Facility User’s Guide
- Tivoli NetView for z/OS MultiSystem Manager User’s Guide

Configuration

2. Perform the following configuration:
   - **Full SA z/OS**: Import the “NMC best practice policy that is delivered with SA z/OS into your policy database and customize its definitions there to fit your environment.
   - **SA z/OS Satellite**: You must start the environment manually.

   SA z/OS delivers a NetView message automation table fragment AOFMSGST that automates this setup:
   - **Full SA z/OS**: You must define this fragment in the customization dialog to be loaded on the focal point only. With this table, the SA z/OS topology manager is started after the completion message from the MultiSystem Manager.
   
   Alternatively, you can specify the following statement in the NetView style sheet:
   
   TOWER.SA = SYSOPS SATELLITE
   
   - **SA z/OS Satellite**: Specify the following statement in the NetView style sheet:
   
   TOWER.SA = SATELLITE

   For additional information, refer to the description of the NetView style sheet and AOFMSGST in “Step 28B: Configure the Networking NetView or Focal Point NetView DSIPARM Data Set” on page 138.

Security


NetView Operators Tasks

4. The RODM name and RODM user must be configured in member AOFRODM on the focal point system (see “Step 28B: Configure the Networking NetView or Focal Point NetView DSIPARM Data Set” on page 138). Customizing AOFRODM on any other system is not necessary.

   - **Full SA z/OS**: You must define the following Automation Operators in the customization dialog for both the satellite and target systems:
     - EVTOPER (the default for the primary is AUTEVT1 and for the backup is AUTEVT2)
     - HBOPER (the default is AUTHB)
     - HBSLV (the default is AUTBSLV)
     - POSTOPER (the default is AUTPOST)
     - POSTSLV (the default is AUTPOSTS)

   **Note**: If the task defined on the target systems is different to the task defined on the satellite, INGSEND definitions must be defined with the customization dialogs to provide the mapping by using the SEND COMMAND OPERS policy item of the Enterprise policy object.
   
   These automation operator definitions are included in the “NMC best practice policy. Customize them there to fit your environment.
   
   You must also define these automation operators in DSIOPF or RACF or both.
Step 29: Installing and Customizing the NMC Focal Point

- **SA z/OS Satellite**: No customization is required because this is done automatically by SA z/OS during the initialization of the satellite.

**SA z/OS Data Storage**
5. Two repositories are provided for SA z/OS data:
   - The Automation manager (for target systems)
   - RODM (for the focal point)

**NetView Common Global Variables**
6. You must set the following NetView common global variables for the target system. Set them in the NetView style sheet. The defaults are underlined.

### AOFUPDAM
Determines whether SA z/OS data should be stored in the automation manager:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
<th>Can Be Set For:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>SA z/OS data is stored in the automation manager.</td>
<td>☑️</td>
<td>X</td>
</tr>
<tr>
<td>NO</td>
<td>SA z/OS data is not stored in the automation manager.</td>
<td>☑️</td>
<td>☑️</td>
</tr>
</tbody>
</table>

### AOFUPDRODM
Determines whether SA z/OS data should be stored in RODM:

<table>
<thead>
<tr>
<th>Value (NMC user)</th>
<th>Meaning</th>
<th>Can Be Set For:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>YES (NMC user)</td>
<td>SA z/OS data is stored in RODM.</td>
<td>☑️</td>
<td>☑️</td>
</tr>
<tr>
<td>NO (non-NMC user)</td>
<td>SA z/OS data is not stored in RODM.</td>
<td>☑️</td>
<td>☑️</td>
</tr>
</tbody>
</table>

### AOFSENDALERT
Defines the mechanism that is used to forward data from the target to the focal point. It is only relevant if AOFUPDRODM has been set to YES.

<table>
<thead>
<tr>
<th>Value</th>
<th>Mechanism</th>
<th>Can Be Set For:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>Alerts</td>
<td>☑️</td>
<td>☑️</td>
</tr>
<tr>
<td>NO</td>
<td>Command Handler</td>
<td>☑️</td>
<td>X</td>
</tr>
</tbody>
</table>

Setting the values of AOFUPDAM, AOFUPDRODM, and AOFSENDALERT on a Networking NetView (for a satellite) or a Focal Point NetView is not necessary because this is done automatically.

AOFUPDAM, AOFUPDRODM, and AOFSENDALERT must be set to the same value on each target system in a sysplex.

AOFUPDAM used in conjunction with AOFUPDRODM will control if and where the SA z/OS is stored as shown in Table 20 on page 142.
### Step 29: Installing and Customizing the NMC Focal Point

**Table 20. Use of AOFUPDAM and AOFUPDRODM to Control SA z/OS Data Storage.**

<table>
<thead>
<tr>
<th>AOFUPDAM</th>
<th>AOFUPDRODM</th>
<th>Storage Outcome</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>YES</td>
<td>SA z/OS data stored in the automation manager and also in RODM.</td>
<td>NMC user, any loss of contact between the target systems and the focal point will be followed by the RODM data being rebuilt from the SA z/OS data that had previously been stored in the automation manager, this will ensure no loss of SA z/OS data shown on the NMC.</td>
</tr>
<tr>
<td>YES</td>
<td>NO</td>
<td>SA z/OS data stored in the automation manager only.</td>
<td>Non-NMC user, it is possible to create a feed from the SA z/OS data held in the automation manager (not used at present, may be used in future releases of SA z/OS).</td>
</tr>
<tr>
<td>NO</td>
<td>YES</td>
<td>SA z/OS data stored in the RODM only.</td>
<td>NMC user, no requirement to rebuild the RODM SA z/OS data.</td>
</tr>
<tr>
<td>NO</td>
<td>NO</td>
<td>SA z/OS data not stored in the automation manager or in RODM.</td>
<td>Non-NMC user.</td>
</tr>
</tbody>
</table>

### Step 29B: Modify the NetView DSIPARM Data Set for the SA z/OS Topology Manager

There are a few things you have to do to prepare for the SA z/OS topology manager to run. Table 21 lists the data sets to be modified for this.

**Table 21. DSIPARM Members to be modified for the SA z/OS Topology Manager**

<table>
<thead>
<tr>
<th>DSIPARM Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFOPFFP</td>
<td>System operations automation operator definitions</td>
</tr>
<tr>
<td>CNMSTYLE/CxxSTGEN</td>
<td>NetView system level parameters for NetView initialization</td>
</tr>
<tr>
<td>DSL6INIT</td>
<td>Initialization member for the NetView DSL6DST task.</td>
</tr>
<tr>
<td>DSICRTTD</td>
<td>NetView CNM router initialization member</td>
</tr>
<tr>
<td>DUIFPMEM</td>
<td>NetView focal point definitions</td>
</tr>
<tr>
<td>DUIGINIT</td>
<td>GMFHS initialization member</td>
</tr>
<tr>
<td>FLCSAINP</td>
<td>MultiSystem Manager initialization member</td>
</tr>
<tr>
<td>INGTOPOF</td>
<td>NMC definition member</td>
</tr>
</tbody>
</table>

**NetView Stylesheet**

*Note:* This is only necessary if you have chosen to use alert forwarding as your communication method.

To avoid further changes, alert forwarding ALERTFWD NV-UNIQ is recommended. However, any of the following SNA-MDS settings can be defined:

- ALERTFWD SNA-MDS=LOGONLY
Step 29: Installing and Customizing the NMC Focal Point

- ALERTFWD SNA-MDS=AUTHRCV
- ALERTFWD SNA-MDS=SUPPRESS

Although SNA-MDS is not absolutely required, it might be important as it allows the construction of networks with intermediate focal points and hot backups.

If the network contains an intermediate focal point, ALERTFWD SNA-MDS must be specified in CNMSTYLE/CxxSTGEN. If the network does not contain an intermediate focal point, ALERTFWD NV-UNIQ may be specified in CNMSTYLE/CxxSTGEN.

If ALERTFWD SNA-MDS is specified in CNMSTYLE/CxxSTGEN, the following entries must be added to sample BNJRESTY:

```plaintext
E0 AUTO SYSTEM AUTOMATION FOR z/OS
E1 DOWN SYSTEM AUTOMATION FOR z/OS
E2 NET SYSTEM AUTOMATION FOR z/OS
```

**Note:** The three values shown above ('E0', 'E1', and 'E2') are the first three user-defined values. If you already have user-defined entries in BNJRESTY, you may use alternative values for these entries.

For more information about how to add user-defined entries (E0–EF) to BNJRESTY, refer to the following chapters in *Tivoli NetView for z/OS Customization Guide*:
- Customizing Hardware Monitor Displayed Data
- Using NMVT Support for User-Written Programming
- Adding or Modifying Resource Types

For more information about the ALERTFWD statement, refer to *Tivoli NetView for z/OS Administration Reference*.

**DSI6INIT**

This is the initialization member for the NetView DSI6DST task and needs to have the appropriate focal point defined.

```plaintext
DEFFOCPT TYPE=ALERT, PRIMARY=NETA.CNM02, BACKUP=NETA.CNM03
```

Note that on the focal point and the backup you will need different members, as NetView complains if a definition references its own system.

Usage of the LU 6.2 alert forwarding mechanism allows for the construction of focal point networks that include intermediate focal points.

**Automation Operator IDs**

Each focal point that will be running the SA z/OS topology manager must have an autotask defined for it. Your environment may have one or more of the following types of focal point:
- The primary focal point
- The secondary focal point
- The intermediate focal point (IFP)

This requires a definition in DSIPARM.DSIOPF, as follows:

```plaintext
&domain.TPO OPERATOR PASSWORD=&domain.TPO PROFILEN AOPFRFA0
```
Step 29: Installing and Customizing the NMC Focal Point

This definition must be made on the focal points and on each target system. It should only be started as an autotask on the focal point.

An include member, DSIPARM.AOFOPFFP, has been provided to help you centralize and manage these operator IDs. You need to customize it to contain the operator IDs for your focal points.

The &domain. variable contains the focal point's domain ID. This is just a suggestion for the naming scheme.

Note: The names must be unique on the focal point and the target systems.

Additionally, on the focal point, the operator ID must be defined in the DSIPARM.AOFMSGST member, as the value for the %AOFOPTOPOMGR% synonym.

SYN %AOFOPTOPOMGR% = '&domain.TPO';

You should not include any backup operators in this synonym.

Installing and customizing needs to be done on the NMC focal point system or on each target system. (This is only for ProcOps.)

It is recommended to use system symbols for the focal point, backup, and intermediate focal point specification. In this case, you can update AOFOPFFP and AOFMSGSY accordingly and make it available in a general data set to all your systems, focal points, and targets. This avoids the same specification of two members on any single system.

You will need one set of autotasks for your primary focal point and a second set for your backup focal point. If you are using intermediate focal points, you will also need a set of operators for each of those (but only on the target systems that are defined to the IFP). Note that even in an IFP situation, the focal point will contact all target systems directly to obtain status and configuration data. The IFP is only used for alert forwarding.

Operator Profiles

This concerns statements in the NetView operator definition file (DSIOPF), which associate operator IDs with logon profiles and the profiles themselves, which are defined in the DSIPRF concatenation.

Each operator who will be an NMC Administrator must be assigned a NetView logon profile which includes the NGMFADMN=YES key/value pair on its AUTH tag.

Each NMC user who needs to issue commands against resources through the NMC interface needs to be linked to a profile with the NGMFCMDS=YES key/value pair on its AUTH tag.

DSICRTTD

The focal points need to be identified to your target systems. Uncomment and adapt the following lines for any of your target systems:

* DEFFOCPT PRIMARY=CNM02LUC,TYPE=ALERT,BACKUP=CNM99LUC
* alerts
* RMTCMD/XCF
DUIFPMEM

Uncomment and adapt the following 4 statements.

*USETCPIP = NO
*TCPANAME = &CNMTCPN
*Sockets = 50
*PORT = 4020

Change USETCPIP to YES. Change the PORT number to an unused number in your system if necessary.

DUIGINIT

Change the domain specification to your focal point domain.

If you use Kanji support check that GMFHS is enabled to send Japanese text to an NMC console for display. In DUIGINIT you have to set JAPANESE=ON.

INGTOPOF

Define your sysplex to your NMC as described in “Step 29D: Configure the INGTOPOF File” on page 146.

Step 29C: Configure RODM

You need to configure RODM so that it will dynamically refresh the workstation when a number of fields other than DisplayResourceStatus is changed. To do this you need to ensure that certain RODM loader statements are processed whenever the GMFHS Data Model is reloaded.

Add the DD statement with member INGDYNRF in the NetView sample procedure EKGLOADP.

```
:* //**EKGIN1 DD DSN=EKGIN1,DISP=SHR
*//EKGIN1 DD DSN=SQ1..V&NETVER..CNMSAMP(DUIFSTRC),DISP=SHR
*// DD DSN=SQ1..V&NETVER..CNMSAMP(FLBTRDM1),DISP=SHR
*// DD DSN=SQ1..V&NETVER..CNMSAMP(FLBTRDM2),DISP=SHR
*// DD DSN=SQ1..V&NETVER..CNMSAMP(FLBTRDM3),DISP=SHR
*// DD DSN=SQ1..V&NETVER..CNMSAMP(FLBTRDM4),DISP=SHR
*// DD DSN=SQ1..V&NETVER..CNMSAMP(FLBTRDM5),DISP=SHR
*// DD DSN=SQ1..V&NETVER..CNMSAMP(FLBTRDM6),DISP=SHR
*// DD DSN=SQ1..V&NETVER..CNMSAMP(FLBTRDM7),DISP=SHR
*// DD DSN=SQ1..V&NETVER..CNMSAMP(FLBTRDMM),DISP=SHR
*// DD DSN=SQ1..V&NETVER..CNMSAMP(FLBTRDMM9),DISP=SHR
*// DD DSN=SQ1..V&NETVER..CNMSAMP(FLBTRDMA),DISP=SHR
*// DD DSN=SQ1..V&NETVER..CNMSAMP(FLBTRDBM),DISP=SHR
*// DD DSN=SQ1..V&NETVER..CNMSAMP(FLBTRDMC),DISP=SHR
*// DD DSN=SQ1..V&NETVER..CNMSAMP(FLBTRDMDCF),DISP=SHR
*// DD DSN=SQ1..V&NETVER..CNMSAMP(FLBTRDMZ),DISP=SHR
*// Dynamic update of resources
*// DD DSN=SQ2..V&SAMVER..SINGSAMP(INGDYNRF),DISP=SHR
*//*
*//*
*:// Figure 11. Sample of RODM Load Procedure EKGLOADP
```
Step 29: Installing and Customizing the NMC Focal Point

Step 29D: Configure the INGTOPOF File

The generic name for the topology control file is “INGTOPOF”. Local versions of the INGTOPOF file may also be created.

The naming (format) of the local versions will be “TPF” concatenated with the domain name of the focal point. For example, if the focal point has a domain name of “IPSNM”, the local INGTOPOF name will be “TPFIPSNM”.

Multiple INGTOPOF files (generic and local) may exist with a single DSIPARM. This will provide the flexibility to tailor each INGTOPOF to suit the requirements of each focal point.

When the topology manager attempts to read the topology control file, in the first instance it will look for the local INGTOPOF member name in DSIPARM. Processing is as follows:

1. If the local INGTOPOF member exists in DSIPARM, the content of that member will be used by the topology manager.
2. If the local INGTOPOF member does not exist in DSIPARM, the topology manager will attempt to read the INGTOPOF member in DSIPARM.
3. If the INGTOPOF member exists in DSIPARM, the content of that member will be used by the topology manager.
4. If the INGTOPOF member does not exist in DSIPARM, the topology manager will terminate with RC = 9.

The following overview of the operation mode of the SA z/OS topology manager supplies some background for discussing the INGTOPOF file. Some familiarity with the class structure of RODM and with the BLDVIEWS tool is assumed.

During initialization, the SA z/OS topology manager gathers information about generated SA z/OS resources from the sysplex and stores the resources in RODM, prefixing their names with the current sysplex name. Usually not only the resources, but also the dependencies and major/minor relationships between resources will be represented in RODM (this depends on the OPTION statement in the INGTOPOF file, see Appendix B, “Syntax for INGTOPOF File,” on page 215).

The INGTOPOF file supplies the SA z/OS topology manager with the following information:

- which sysplexes there are and which of their member systems contain a SA z/OS topology agent.
- the names of the data sets (members) that contain the definitions of the views.
- when views must be rebuilt during runtime, it is desirable that only those views be rebuilt to which new members have been added.

You will need to prepare the INGTOPOF input file. This contains information about the target domains and how they are grouped into sysplexes along with some additional information that affects the resources that are dynamically created.

The INGTOPOF file contains configuration information for the SA z/OS topology manager. It must reside in DSIPARM. The records of the file consist of a keyword with one or more parameters. Comment lines must have an asterisk (*) in the first column. A ‘+’ at the end of a line indicates that the record is continued in the next line.
Step 29: Installing and Customizing the NMC Focal Point

The information is passed from the INGTOPOF file to the SA z/OS topology manager with the help of the following keywords:

- SYSPLEX
- PROCOPS
- BLDVIEWS
- [LOCATION]
- [ANCHOR]
- [OPTION]
- [TEMPLATE]
- [MAPCOLOR]

The syntax of the statements in the INGTOPOF file is described in Appendix B, “Syntax for INGTOPOF File,” on page 215.

A sample of INGTOPOF is provided in the SINGNPRM library.

To start the MultiSystem Manager and load the INGTOPOF file, use the MultiSystem Manager start command FLCAINIT.

Step 29E: Prepare BLDVIEWS Cards

You need to provide files with BLDVIEWS cards. These are required for the SA z/OS resources to appear on the NMC workstation. These files will become part of the BLDVIEWS statement in the INGTOPOF file. The BLDVIEWS statement in the INGTOPOF file is used by the SA z/OS topology manager to pass information to the BLDVIEWS tool which it invokes to produce the views of the objects. The BLDVIEWS tool writes information about views into RODM. The SA z/OS topology manager is automatically invoked whenever you start SA z/OS or you can invoke it with the INGTOPO command whenever you changed information in the INGTOPOF file or in the files with the BLDVIEWS cards.

To run the BLDVIEWS tool, use one of the following methods:

- via the SA z/OS topology manager which invokes the tool
- via an external invocation of this tool (as a NetView command in a NetView session)

For information about the BLDVIEWS cards syntax refer to the appropriate NetView documentation.

The following three SA z/OS BLDVIEW samples are provided in the SINGNPRM library matching the INGTOPOF sample file:

- INGBVIEW (sample view for SysOps objects)
- INGPVIEW (sample view for ProcOps objects)
- INGCVIEW (sample view for common objects)

Note: To start MultiSystem Manager and load the INGTOPOF file, use the MultiSystem Manager start command: FLCAINIT

Step 30: Copy and Update Sample Exits

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>
Step 30: Copy and Update Sample Exits

Several sample exits are provided in the SINGSAMP library (for example, AOFEXC01). You can use these samples to create your own exits. If used, they must be copied into a data set (either the enterprise-specific or domain-specific) in the DSICLD concatenation. These exits are called at fixed points during SA z/OS processing. Therefore, you should look into each of the sample exits to determine whether you need to use and update it.

Updating and copying the sample exits allows you to add your specific processing. For more information on user exits, provided samples and advanced automation options, refer to IBM Tivoli System Automation for z/OS Customizing and Programming.

Step 31: Install Relational Data Services (RDS)

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If you plan to use Relational Data Services (RDS) an extra VSAM cluster needs to be defined in order to make RDS tables persistent.

The sample job INGEMUVS is provided in ING.SINGSAMP to define the VSAM cluster.

Adapt your NetView startup procedure and add DD statement:

```
//INGEMUGL DD DSN=#hlq#.domain#.EMUGLBL,DISP=SHR
```

You may also refer to sample startup procedure INGENVSA in ING.SINGSAMP.

Note: Due to the maximum records size of 32000 for a VSAM KSDS record, a RDS table row cannot be larger than 32000 bytes.

Step 32: Install CICS Automation in CICS

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This section describes the basic CICS Automation definitions that take place on CICS. Refer to the CICS documentation while performing these steps, especially the CICS Resource Definition Guide. These steps are performed on each CICS region.

Step 32A: SIT or Startup Overrides

On each CICS, ensure that the system initialization table (SIT) or startup overrides include the following:

- `PLTPI=xx`, where xx is the suffix to the startup PLT
- `PLTSD=yy`, where yy is the suffix to the shutdown PLT
- `MSGLVL=1`,
- `BMS=(STANDARD|FULL)`

If CICS is started with option MSGLVL=0, some of the messages may not be passed to automation.
Step 32: Install CICS Automation in CICS

You may optionally add CN as your last startup override, whether from SYSIN or through the JCL. However, this is not necessary if you have added the &APPLPARMS variable to the PARM of the CICS start command in the STARTUP item of the APPLICATION policy object. The following is an example:

MVS $ cics,...,PARM='SYSIN,START=xxxx&APPLPARMS'

This is also how the start commands are predefined in the sample databases.

Step 32B: Program List Table Definitions

Add the TYPE=ENTRY definitions shown in the following example to the post-initialization program list table (PLT) for each CICS after the entry for DFHDELM (as in phase 2):

DFHPLT TYPE=INITIAL,SUFFIX=xx
DFHPLT TYPE=ENTRY,PROGRAM=DFHDELM
DFHPLT TYPE=ENTRY,PROGRAM=EVEPYINI
DFHPLT TYPE=ENTRY,PROGRAM=EVESTISP
DFHPLT TYPE=FINAL

The EVESTISP program definition in this example is only needed when using the CICS PPI communication.

Add the TYPE=ENTRY definitions shown in the following example to the shutdown program list table (PLT) for each CICS.

DFHPLT TYPE=INITIAL,SUFFIX=yy
DFHPLT TYPE=ENTRY,PROGRAM=EVESPLTT
DFHPLT TYPE=ENTRY,PROGRAM=DFHDELM
DFHPLT TYPE=FINAL

Assemble the PLT tables.

Step 32C: Define Consoles

CICS Automation uses EMCS consoles to issue Modify CICS commands when managing CICS. Console definitions are required for correct CICS Automation operation.

Define consoles for autotasks to enable CICS Automation functions. This step can be skipped if you enable CICS Auto-Installed Consoles. This can be achieved by specifying "AICONS=YES" in the CICS system initialization parameters.

In an EMCS environment the autotask console names are determined, in order of precedence as follows:

1. If you are using AOCGETCN (that is, using the profiles shipped with the product) the name is determined by AOFCNMASK. For more information, see IBM Tivoli System Automation for z/OS Customizing and Programming or IBM Tivoli System Automation for z/OS Defining Automation Policy.

2. The CONSNAME parameter on the PROFILE statement in the task profile determines the EMCS console name. For more information, see Tivoli NetView for z/OS Administration Reference and Tivoli NetView for z/OS Security Reference.

3. By default the autotask name is used for the EMCS console name.

A console has to be defined for each SA z/OS work operator. These are typically named AUTWRKxx. In addition, a console has to be defined for each NetView operator that may want to inquire or control a CICS region. This can be simplified by specification of the CICS Console Auto-Install function.
RACF security is provided by z/OS for EMCS and MCS consoles. This function enables a user on NetView with a RACF user ID (ACEE) to open an EMCS console and have the user ID associated with the EMCS console. All commands that are issued to the EMCS console will have the user ID of the NetView user. Furthermore, CICS supports EMCS and MCS consoles with RACF user IDs by inheriting the user ID that is associated with a command from the EMCS or MCS console.

The net result is that for CICS auto-installed consoles, the user ID that is assigned to the console is the user ID that issued the command. In the case of SA z/OS this would be the NetView user's user ID (only if NetView is using RACF to verify user IDs). This means that all tasks in NetView that require consoles will also require RACF user IDs and the appropriate permissions in CICS. This includes all human operators and all auto operators.

For those users who want to have a predefined user ID instead of the all the possible user IDs from NetView, the Console Model Terminal definition should specify a user ID in its definition.

**Step 32D: Transaction and Program Definitions**

This step describes how to define the standard CICS Automation transactions and programs to CICS. The DFHCSDUP program is used to do this.

The members required to run these jobs are provided with CICS Automation. However, some modifications are required, as described below:

**Hint:** You might want to back up your CSDs before doing this step.

For each CSD, run the EVESJ015 sample job. This job defines transactions and programs for CICS automation in a group called EVEGRP1.

Before you run it, modify the job as directed in the JCL comments.

When using the CICS PPI, run the EVESJPP1 sample job to define the necessary transactions and programs in a group called EVEGRP2.

**Step 32E: DFHRPL and the CICS Automation Library**

Update the DFHRPL concatenation to add the ING.SINGMOD1 library for every CICS subsystem that is to be managed by SA z/OS.

**Note:** Do not add these libraries to the DFHRPL for CICSPlex CMAS subsystems.

**Step 32F: Add Libraries for NetView**

Uncomment any libraries that you require in the INGENVSA member of the SINGSAMP data set. Refer to the sample for more details.

**Step 32G: Installing CICSPlex SM REXX API**

The CICSPlex System Manager REXX API is required for the interaction between SA z/OS and the CICSPlex System Manager. The REXX runtime interface to the API is supplied as a function package or host command environment. It should preferably be added to the function package table in the NetView module DSIRXPRM, as shown in “Step 6F: Add the REXX Function Packages to DSIRXPRM” on page 95.
Step 32: Install CICS Automation in CICS

For details about the installation of a function package, see CICS Transaction Server for z/OS Installation Guide and IBM Tivoli NetView for z/OS Tuning Guide.

Step 33: Install IMS Automation in IMS

<table>
<thead>
<tr>
<th></th>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Step 33A: Specify Required Control Region Parameters

Modify all IMS Control region and IMS DB control region JCL to specify the following parameter:

**CMDMCS=Y**

This is required for correct operation of IMS product automation.

**Note:** Depending on your security requirements and authority assignments, CMDMCS can also be set to values of R, C, or B. For more information, refer to the *IMS System Definition Reference*.

**PREMSG=N**

This is required for correct operation of IMS Product Automation.

**Note:** If PREMSG=Y is selected, all system messages and command responses are issued as multi-line messages. The first line is: DFS0001 MESSAGE(S) FROM ID=XXXX where XXXX is the IMSID. The message starts on the second line. As a result, IMS message automation will not work as expected.

Step 33B: Install DFSAOE00 Exit

There are three ways to install the exit.

- Use the default z/OS exit router as supplied by SA z/OS.
  - This involves concatenating the ING.SINGMOD1 library before the IMS.SDFSRESL library in the STEPLIB concatenation.
  - Add PROGxx members to SYS1.PARMLIB to define the exit. Sample member EVISI005 contains the base required definitions. See IBM Tivoli System Automation for z/OS Product Automation Programmer's Reference and Operator's Guide for further customization details.
- Use the exit that is supplied by SA z/OS on its own.
  - This involves concatenating the ING.SINGMOD1 library after the IMS.SDFSRESL library in the STEPLIB concatenation, unless ING.SINGMOD1 is in the linklist concatenation chain.
  - Relink the EVIPVEX1 module and give it an ALIAS of DFSAOE00 into a library concatenated before IMS.SDFSRESL in the STEPLIB concatenation. Sample EVISJ001 is an example of how to do this.
- Call the SA z/OS exit from your routine.
  - This involves concatenating the ING.SINGMOD1 library after the IMS.SDFSRESL library in the STEPLIB concatenation, unless ING.SINGMOD1 is in the linklist concatenation chain.
  - Call the EVIPVEX1 module from your exit program as detailed in IBM Tivoli System Automation for z/OS Product Automation Programmer's Reference and Operator's Guide.
Step 33: Install IMS automation in IMS

Step 33C: Add Libraries for NetView

Uncomment any libraries that you require in the INGENVSA member of the SINGSAMP data set. Refer to the sample for more details.

In order to issue IMS type 2 commands, access must available to the IMS modules, CSLSRG00 and CSLSDR00. These modules are shipped in the IMS product library named hlq.SDFSRESL. The entire product library can be allocated, or a private data set with just those modules and perhaps an explicit allocation or a LNKLST entry.

Step 34: Install TWS Automation in TWS

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Step 34A: Add Libraries to TWS

Add your SINGMOD1 library and the NetView CNMLINK library containing CNMNETV to the TWS steplib. Alternatively, you may add these libraries to LINKLST. You should have already APF-authorized these libraries.

Step 34B: Add Libraries to NetView

Allocate the EQQMLOG library according to your TWS definitions. This data set contains any error messages that may occur when using the TWS APIs on this NetView.

EQQMLIB should point to the appropriate message library for the level of TWS that you are running.

Uncomment any libraries that you require in the INGENVSA member of the SINGSAMP data set. Refer to the sample for more details.

Step 34C: Update TWS Parameters and Exits

Install the exit module EQQUXSAZ. This exit is required for TWS Automation workstation processing.

A recycle of TWS is required to install the exit 7 module EQQUX007 or the exit 11 module EQQUX011. If you are using an existing exit 7 or exit 11, you can combine these exits with modules that are supplied by TWS Automation.

TWS Automation supplies EQQUX007 to detect workstations that are used for NetView communication. The following modules are used as part of this process:

- EQQUX007
- UX007001
- UX007004
- EQQUX011
- UX011001

EQQUX007 and EQQUX011 are the exit driver programs. They call other modules in turn, as though TWS is calling each module directly.

The EQQUX007 driver searches for UX007001 through UX007010, and the EQQUX011 driver searches for UX011001 through UX011010. UX007001, UX007004, and UX011001 are supplied with TWS Automation.
Step 34: Install TWS Automation in TWS

If you have an existing exit 7, rename your module from EQQUX007 to UX007005. If you have an existing exit 11, rename your module from EQQUX011 to UX011002.

The called routines are passed the same parameters as the call to EQQUX007 or EQQUX011.

If you want to add additional exit 7 or exit 11 modules, use the next available name, such as UX007005 or UX011002. This makes it easier to integrate exits that are supplied by various products. Also, because modules are loaded dynamically by the exit driver on each invocation, you may add, delete, or modify an exit module without recycling TWS.

You must specify the CALL07(YES) parameter in the TWS/ESA initialization parameters.

You must specify the CALL11(YES) parameter in the TWS/ESA initialization parameters if you want to monitor CP deletes. CP delete monitoring allows TWS Product Automation to clear outstanding SDF and NMC alerts when an application or operation is deleted from the current plan.

Other initialization parameters must be specified in the TWS initialization member (EQQPARM) so that TWS will issue some of its messages to the MVS console.

The DURATION, ERROROPER, LATEOPER, and OPCERROR messages are automated by TWS Automation. The RESCONT and QLIMEXCEED messages are useful for further customer automation.

You must specify the following in EQQPARM:

```
ALERTS WTO (DURATION
   ERROROPER
   LATEOPER
   RESCONT
   OPCERROR
   QLIMEXCEED)
```

In addition, you must edit the TWS-supplied message members for certain messages.

The following messages are automated and may require changes to the TWS-supplied message members in the SEQQMSG0 data set:

<table>
<thead>
<tr>
<th>Member</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>EQQE026I</td>
<td>EQQE02</td>
</tr>
<tr>
<td>EQQE036I</td>
<td>EQQE03</td>
</tr>
<tr>
<td>EQQE037I</td>
<td>EQQE03</td>
</tr>
<tr>
<td>EQQE107I</td>
<td>EQQE10</td>
</tr>
<tr>
<td>EQQFCC1I</td>
<td>EQQFCC</td>
</tr>
<tr>
<td>EQQN013I</td>
<td>EQQN01</td>
</tr>
<tr>
<td>EQQPH00I</td>
<td>EQQPH0</td>
</tr>
<tr>
<td>EQQW011I</td>
<td>EQQW01</td>
</tr>
<tr>
<td>EQQW065I</td>
<td>EQQW06</td>
</tr>
<tr>
<td>EQQW079W</td>
<td>EQQW07</td>
</tr>
</tbody>
</table>
Step 34: Install TWS Automation in TWS

Table 22. SEQOMSG0 Data Set (continued)

<table>
<thead>
<tr>
<th>Member</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>EQQZ006I</td>
<td>EQQZ00</td>
</tr>
<tr>
<td>EQQZ086I</td>
<td>EQQZ08</td>
</tr>
<tr>
<td>EQQZ128I</td>
<td>EQQZ12</td>
</tr>
<tr>
<td>EQQZ200I</td>
<td>EQQZ20</td>
</tr>
<tr>
<td>EQQZ201I</td>
<td>EQQZ20</td>
</tr>
</tbody>
</table>

Modify these message members to include WTO=YES for the indicated message IDs. Full details for customizing TWS can be found in *Tivoli Workload Scheduler for z/OS Customization and Tuning*.

**Note:** If you use NMC and SDF to monitor the status of TWS operations, you should enable UX007004 and update INGMSGU1 to remove the Message Automation traps for EQQE026I and EQQE036I. This is to prevent you from receiving multiple NMC and SDF alerts for the same TWS event as a result of the following:

- NMC and SDF alerts that are generated from EQQE036I do not contain an operation number. Therefore, if an application contains operations that have identical job names (with the same IATIME and same workstation ID), it is possible that duplicate or ambiguous alerts are generated.
- Alerts that are generated from EQQE026I and EQQE036I are not removed from NMC and SDF if UX007004 is not active. This is because TWS does not issue a message when these operations exit error status.

Step 35: Configuring GDPS

<table>
<thead>
<tr>
<th></th>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This section describes the necessary customization and definitions when running GDPS on top of SA z/OS.

You can also import the best practice policy, "GDPS", which is delivered with SA z/OS, into your policy database and customize its definitions there to fit your environment.

Step 35A: Preparing NetView

1. Concatenate the SGDPARMP product data set to the DSIPARM DD-statement in the NetView startup procedure. See the INGENVSA sample that is provided by SA z/OS in the SINGSAMP library for more details.
2. If you need to modify the INGXINIT member, which is the initialization member of the SA z/OS communication task for the production system or its equivalent, copy them to your user data sets and make your modifications there. The GDPS controlling system uses the z/OS system symbol &SYSCLONE, as the XCF group ID. This allows the same member to be used for all controlling systems. The resulting XCF group will always be created in a unique way: INGXSGxx, where xx is the value of &SYSCLONE. This corresponds to HSAPRMKS as described in "Step 35B: Preparing the Automation Manager" on page 155.
Step 35: Customizing GDPS

3. If necessary, copy the INGSTGEN member from the sample library (SINGSAMP) to the CNMSTGEN member of the DSIPARM data set of each NetView instance in your sysplex and adapt the TOWER statements according to your installation. Additionally specify the GDPS product of your installation and whether this is the GDPS controlling system (KSYS) or production system (PROD) by removing the asterisk in front of the appropriate line:

*TOWER.SA.GDPS=PPRC KSYS
   If GDPS/PPRC is installed and this is controlling system

*TOWER.SA.GDPS=PPRC PROD
   If GDPS/PPRC is installed and this is production system

*TOWER.SA.GDPS=HM KSYS
   If GDPS/PPRC HM is installed and this is controlling system

*TOWER.SA.GDPS=HM PROD
   If GDPS/PPRC HM is installed and this is production system

*TOWER.SA.GDPS=XRC
   If GDPS/XRC is installed for all systems

*TOWER.SA.GDPS=GM
   If GDPS/GM is installed for all systems

*TOWER.SA.RCMF=PPRC
   If RCMF/PPRC is installed

*TOWER.SA.RCMF=XRC
   If RCMF/XRC is installed

Note: If the TOWER.SA includes GDPS, the VPCEINIT installation exit that is required by each supported GDPS product is automatically called during initialization of SA z/OS. You no longer need to specify it in each system’s SYSTEM INFO policy in the customization dialog. Also SA z/OS will automatically disable recovery for minor resources MVSESA.CS and MVSESA.XCF.

Step 35B: Preparing the Automation Manager

The GDPS controlling system must run in a separate XCF group (subplex) and therefore has its own automation manager. The automation manager parmlib member for the controlling system (K-system) is HSAPRMKS, using the z/OS system symbol &SYSCLONE as the XCF group ID. This allows the same parmlib member to be used for all controlling systems. The resulting XCF group will always be created in a unique way: INGXSGxx, where xx is the value of &SYSCLONE.

Copy and edit the automation manager startup procedure INGEAMSA. The same startup procedure can be used for the automation manager that controls the production systems and the automation manager that controls the K-system, assuming that the PARMLIB member suffix is specified on invocation of the procedure.

Step 35C: Defining the Automation Table Used by GDPS

SA z/OS provides a NetView automation table (AT) that contains all the messages that are required by GDPS. The relevant AT is loaded, depending on the specified GDPS Tower statement, as follows:
Step 35: Customizing GDPS

<table>
<thead>
<tr>
<th>Tower Statement</th>
<th>AT loaded</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOWER.SA.GDPS=PPRC</td>
<td>GEOMSGGP</td>
</tr>
<tr>
<td>TOWER.SA.GDPS=HM</td>
<td>GEOMSGHM</td>
</tr>
<tr>
<td>TOWER.SA.GDPS=XRC</td>
<td>GEOMSGXR</td>
</tr>
<tr>
<td>TOWER.SA.GDPS=GM</td>
<td>GEOMSGGM</td>
</tr>
</tbody>
</table>

**Note:** If the TOWER.SA statement includes GDPS, the INGMSGGP automation table that is required by each supported GDPS product is automatically loaded during initialization of SA z/OS, only if INGMSG01 (which is also the default) is specified in the system's SYSTEM INFO in the customization dialog. If this is not the case, you should also add INGMSGGP to the list of automation tables.

You can use the following AT fragments to process messages for the GEOMSGxx ATs that are supplied by GDPS:
- INGMSGG1 for messages that should not flow into the GEOMSGxx ATs
- INGMSGG2 for messages that do not have an entry in the GEOMSGxx ATs

For messages that should be processed by a user AT as well as the GDPS ATs, you should use a separate AT that is activated in parallel. You can achieve this by specifying multiple AT members in the AUTOMATION SETUP definitions for the system (SYS).

---

Step 36: Configuring I/O Operations

<table>
<thead>
<tr>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

Step 36A: Define OMVS Segment

The TCP/IP socket API that is used by I/O operations requires the definition of an OMVS segment for the user that is assigned to I/O operations when the application is started. Note that I/O operations must be started as a “started task”.

You can omit this step if the OMVS segment already exists for the user that is assigned to the I/O operations application.

Execute the following RACF commands that provide the necessary security definitions. The RDEFINE command assumes that the name of the I/O operations started task begins with IHV:

```
ADDGROUP ihvgrp OMVS(GID(nnn) [SHARED])
ADDUSER ihvusr OMVS(UID(nnn) [SHARED]) DFLTGRP(ihvgrp) NOPASSWORD
RDEFINE STARTED ihvu*. ATDATA(USER(ihvusr) GROUP(ihvgrp))
SETROPTS GENERIC(STARTED) RACLIST(STARTED) REFRESH
```

GID and UID can have the same value. SHARED needs to be specified if the GID value or the UID value is already in use. For more information about the security definitions see the chapter “RACF command syntax” in z/OS Security Server RACF Command Language Reference.
Step 36: Customizing I/O Operations

Step 36B: Prepare I/O Operations Startup

This section describes optional configuration when running I/O operations that you can make by editing the I/O operations startup JCL (see Figure 12).

```
//I0OPS EXEC PGM=IHV0INI, **IHV INITIALIZATION MODULE NAME**
// TIME=1440, **RUN FOREVER**
// REGION=0M, **REGION SIZE**
// DTPTY=(15,15), **PRIORITY OF TASK**
// PARM=''
//
//STEPLIB DD DISP=SHR,DSN=#hlqinst#.SINGMOD1
//HCDTRACE DD DISP=SHR,DSN=#hlq#.SYSNAME..HCDTRACE
//HCDPROF DD DISP=SHR,DSN=#hlq#.HCDPROF
```

Figure 12. Startup JCL of I/O operations

You can add one or more of the following definitions for the PARM parameter:

**COMM={[TCP|VTAM]}

This parameter restricts communication to TCP/IP or VTAM only.

**Note:** Running a mix of I/O operations applications, some started with COMM=TCP and some with COMM=VTAM, leads to unpredictable results.

COMM=TCP should only be specified when _all_ I/O operations applications run SA z/OS V3.2 or higher.

**Attention:** I/O operations does NOT support the same name for the VTAM applid and the TCP/IP host name at the same time.

You can use the same name when you run I/O operations with either type of communication and you want to switch the type of communication. However, you must ensure that all I/O operations applications have terminated normally before you change the type of communication. Otherwise, unpredictable results occur.

**CT=xx** This parameter defines the suffix of the CTIIHV:<x> PARMLIB member that I/O-OPS uses when registering the component trace. If you omit this parameter I/O-OPS uses the default CTIIHVZZ member.

**MSG={[MC|UC]}

This parameter defines the appearance of messages:

- MSG=MC leaves the message in mixed case as defined in the message module.
- MSG=UC translates each message to uppercase before it is issued.

If you omit this parameter I/O operations checks the environment for the following languages:

- **CHS** Chinese Simplified
- **CHT** Chinese Traditional
- **ENP** US English Uppercase
- **JPN** Japanese
- **KOR** Korean
- **TAI** Thai

If any of these languages is installed, each message is translated to uppercase before it is issued.
Step 36: Customizing I/O Operations

**TIMEOUT=0–999999**
This parameter sets the timeout value of the very first I/O operations application to the specified value. All other I/O operations applications ignore the parameter because they inherit the timeout value from the first running I/O operations application that the new application communicates with.

**TPNAME=tpname**
This parameter specifies the TCP/IP procedure name that is used for communication.

In a multi-stack environment (CINET) TCP/IP allows up to 8 different address spaces running in parallel on a single MVS image. This parameter restricts the TCP/IP communication of I/O operations to the specified procedure. Otherwise the first active procedure is used.

*Note:* Do not specify the parameter in a single-stack environment (INET).

Two or more parameters must be separated with a comma.

---

**Step 37: Installing Tivoli Enterprise Portal Support**

<table>
<thead>
<tr>
<th></th>
<th>SysOps</th>
<th>ProcOps</th>
<th>I/O Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Step 37A: Enabling the SA z/OS Monitoring Agent**

If you plan to use the SA z/OS monitoring agent you must perform the SMP/E installation of the support for the Tivoli Enterprise Portal (TEP). For further details, refer to *IBM Tivoli System Automation for z/OS Monitoring Agent Configuration and User’s Guide* and *IBM Tivoli Monitoring Services: Program Directory*.

You can import the best practice policy, *ITM, which is delivered with SA z/OS, into your policy database and customize its definitions there to fit your environment.

**Step 37B: Enabling SOAP over HTTPS for a TEMS**

This step is necessary if you want SA z/OS to direct SOAP queries to Tivoli Enterprise Monitoring Server (TEMS) using the HTTPS protocol. If you do not do this, you can only use the insecure HTTP protocol.

If you intend to communicate with multiple TEMS servers (for example, in a HA hub TEMS configuration not running on z/OS) from the same system you need to repeat for each one.

Please refer to the z/OS Communication Server documentation for details.

Be aware that the TCP/IP profile has to contain the statement TCPCONFIG TTLS to result in the activation of the processed policy definitions.
AT-TLS Policy

Figure 13 is a sample AT-TLS policy with the highest TCP/IP trace. Please specify `<tlsKeyring>` and `<ip_addr>` accordingly. The `<ip_addr>` is the IP address of the machine hosting the TEMS server that you wish to direct the SOAP query to:

```plaintext
TTLSRule NV_TEMS_WIN
{
  LocalAddr ALL
  RemoteAddrRef addr_TEMS
  LocalPortRange 0
  RemotePortRange 3661
  Direction Outbound
  Priority 255
  TTLSGroupActionRef XXGRP
  TTLSEnvironmentActionRef XXENV
  TTLSConnectionActionRef XXCON
}
TTLSGroupAction XXGRP
{
  TTLSEnabled On
}
TTLSEnvironmentAction XXENV
{
  HandshakeRole Server
  EnvironmentUserInstance 0
  TTLSKeyringParmsRef keyRing
  TTLSEnvironmentAdvancedParmsRef XXADV
  Trace 255
}
TTLSConnectionAction XXCON
{
  HandshakeRole Client
  Trace 255
}
TTLSEnvironmentAdvancedParms XXADV
{
  ApplicationControlled Off
  ClientAuthType PassThru
}
TTLSKeyringParms keyRing
{
  Keyring `<tlsKeyring>`
}
IpAddr addr_TEMS
{
  addr `<ip_addr>`
}
```

Figure 13. Sample AT-TLS policy

Certificate registration in keyring

The ITM Soap Server sends a self-signed certificate which has to be registered in the keyring. The certificate can be obtained easily if a web request is sent from a workstation browser.

Use the following URL for this purpose:
https://<ip_addr>:3661///cms/soap/kshhssoap.htm

You are asked to accept or deny the certificate. Store this certificate in X.509 PEM format (base64), upload this file to z/OS with ASCII to EBCDIC translation and add it to your keyring.
Chapter 11. Security and Authorization

You can secure the product. Only authorized personnel are able to access product-specific data sets, find out runtime information about automated resources, or change the status of such resources.

After the initial configuration, the product is set up so that you familiarize yourself with the functions for testing purposes and you make it secure for your production environment. However, before you begin, you are advised to change the default passwords of the operator IDs that come with the product. You locate the default operators that are defined in <nv_h1q_smpe>.DSIPARM member DSIOPFEX. Copy this member to <sa_h1q_user>.DSIPARM, edit it and change the PASSWORD parameter for each of them. For example, to change OPER1's password to XYZ123, specify:

OPER1 OPERATOR PASSWORD=XYZ123
PROFLEN DSI1PROFA

Use a System Authorization Facility (SAF) product, such as the z/OS Resource Access Control Facility (RACF) to secure your environment as follows:

- Operators are defined and authenticated by a SAF product
- Command authorization is done by a SAF product that is based on the issuer of a command
- Resource authorization is done by a SAF product that is based on the issuer of particular commands

SA z/OS facilitates the steps of securing your environment. The Configuration Assistant generates the INGESAF member that is based on the input in your Configuration Options file. The INGESAF member contains the following items:

- Profiles that protect commands and other resources
- Definitions of groups that represent roles
- Group membership that contain the individual operators in each role
- Necessary definitions for all the auto operators that are required by the product
- PERMIT statements that grant certain roles access to definitions for commands

You find the INGESAF member and all the other generated members in the CONFLIB data set. See Chapter 9, "Base SA z/OS Configuration Using the Configuration Assistant," on page 55 for details about using the Configuration Assistant.

It is assumed that you intend to follow the IBM recommendations to secure your automation environment, and to use the samples in the INGESAF member. See IBM Tivoli NetView for z/OS: Security Reference for a complete description for details about the recommended settings and other security options that you can use.

Notes:
1. For evaluation and browsing purposes a member INGESAF in a readable format is also provided in the SINGSAMP sample library. Refer to the description section of this member and discover the provided security definitions within this member. For establishing the SAF-based security environment it is required to use the Configuration Assistant.
2. Make sure you have APAR OA41282 installed. With this APAR, the z/OS RACF provides the new general SYSAUTO resource class as a system-provided resource class.

## Authorization of the Started Procedures

The started procedures for the Automation Manager, the Automation Agent, the Subsystem Interface, and the IPL Data Gatherer need authority to access SAF-protected resources.

Use the STARTED class. None of the started procedures requires the PRIVILEGED or TRUSTED attribute. You must check with your security administrator for details.

The names of the started procedures are listed in Table 23:

<table>
<thead>
<tr>
<th>Function</th>
<th>Default Procedure Name</th>
<th>Real Procedure Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automation Manager</td>
<td>INGEAMSA</td>
<td>Value of sa_am_start_proc, otherwise use what is specified for sa_am_start_job.1</td>
</tr>
<tr>
<td>Spare Automation Manager</td>
<td>INGEAMSA</td>
<td>Value of sa_am_start_proc, otherwise use what is specified for sa_am_start_job.2</td>
</tr>
<tr>
<td>Automation Agent</td>
<td>INGENVSA</td>
<td>Value of sa_saagent_start_proc, otherwise use what is specified for sa_saagent_start_job</td>
</tr>
<tr>
<td>Subsystem Interface</td>
<td>CNMSJ010</td>
<td>Value of sa_nvssi_start_proc, otherwise use what is specified for sa_nvssi_start_job</td>
</tr>
<tr>
<td>IPL Data Gatherer</td>
<td>HSAPIPLC</td>
<td>Value of sa_ipldata_start_proc, otherwise use what is specified for sa_ipldata_start_job</td>
</tr>
</tbody>
</table>

Table 24 lists the SAF-protected resources, that each started procedure needs access to:

<table>
<thead>
<tr>
<th>Function</th>
<th>SAF-Resources</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automation Manager</td>
<td>&lt;sa_automation_policy&gt; SOCNTL</td>
<td>READ</td>
</tr>
<tr>
<td></td>
<td>&lt;sa_hlq_smpe&gt;**</td>
<td>READ</td>
</tr>
<tr>
<td></td>
<td>&lt;sa_hlq_user&gt;**</td>
<td>UPDATE</td>
</tr>
<tr>
<td>Automation Agent</td>
<td>&lt;sa_automation_policy&gt; SOCNTL</td>
<td>READ</td>
</tr>
<tr>
<td></td>
<td>&lt;sa_hlq_smpe&gt;**</td>
<td>READ</td>
</tr>
<tr>
<td></td>
<td>&lt;sa_hlq_user&gt;**</td>
<td>READ</td>
</tr>
<tr>
<td></td>
<td>&lt;sa_hlq_user&gt;.*DSILIST</td>
<td>UPDATE</td>
</tr>
<tr>
<td></td>
<td>&lt;sa_hlq_user&gt;.*STATS</td>
<td>UPDATE</td>
</tr>
<tr>
<td>IPL Data Gatherer</td>
<td>&lt;sa_hlq_user&gt;.*INGXSG.*IPLDATA</td>
<td>CONTROL</td>
</tr>
<tr>
<td></td>
<td>SYSL.PARMLIB</td>
<td>READ</td>
</tr>
<tr>
<td>Subsystem Interface</td>
<td>&lt;sa_hlq_smpe&gt;**</td>
<td>READ</td>
</tr>
</tbody>
</table>
Authorization of the Started Procedures

To enable the Automation Manager to properly shut down OMVS, super user permission for UNIX System Services must be granted. The Automation Manager's user must have an OMVS segment and access to the BPX.SUPERUSER resource. Additionally, add library `<sys_h1q_sceerun>.SCEERUN`, `<sys_h1q_sceerun>.SCEERUN2` and `<sys_h1q_sceerun>.SCLBDLL` as Program Controlled and authorize the Automation Manager’s user accordingly. Check with your security administrator for details.

Roles

To facilitate the definition of command authorizations for human and auto operators, it is recommended to use groups. Each group corresponds to a certain usage profile or role.

The product comes with five predefined roles that are described in Table 25:

Table 25. Security Roles

<table>
<thead>
<tr>
<th>Role</th>
<th>Default Group</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>INGUSER</td>
<td>In this role, an operator can merely display a few things but cannot change or otherwise influence how the automation works.</td>
</tr>
<tr>
<td>Operator</td>
<td>INGOPER</td>
<td>In this role, an operator can use panels to do what is necessary to keep the system in running order on a day to day basis.</td>
</tr>
<tr>
<td>Administrator</td>
<td>INGADMIN</td>
<td>In this role, an operator has the rights to perform special commands. Such as loading of a new automation configuration or otherwise act beyond the scope of the daily work of a normal operator.</td>
</tr>
<tr>
<td>Auto Operator</td>
<td>INGAUTO</td>
<td>In this role, an operator has the rights to perform all commands and services that can be started from the product, user scripts, and the automation table that are required to bring resources into their Desired status or to recover from failures. The permissions for this role are required for the product to work correctly as specified in the INGESAF member.</td>
</tr>
<tr>
<td>Superuser</td>
<td>INGSUPER</td>
<td>In this role, an operator has no restrictions.</td>
</tr>
</tbody>
</table>

Refer to the INGESAF member for a complete reference of commands and services and the associated roles as provided by the product.
Roles

Note: The mapping of roles and commands in the INGESAF member is only a guideline. Following the recommendations in this member, however, reduces the time to secure the environment.

If you have groups in your environment that you would like to reuse, ensure that the groups are defined with similar characteristics as described in the INGESAF member. In particular, if you intend to automate UNIX System Services processes, for each group, an OMVS segment is required that contains the group ID for this group.

The following variables in the Configuration Options file are used to specify the SAF-group name for each of the roles that are listed here. You do not have to change the default names unless your organization follows a naming convention:

Table 26. Option File variables for SAF-group name

<table>
<thead>
<tr>
<th>Options File variable</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>racf_group_user</td>
<td>INGUSER</td>
</tr>
<tr>
<td>racf_group_oper</td>
<td>INGOPER</td>
</tr>
<tr>
<td>racf_group_admin</td>
<td>INGADMIN</td>
</tr>
<tr>
<td>racf_group_auto</td>
<td>INGAUTO</td>
</tr>
<tr>
<td>racf_group_super</td>
<td>INGSUPER</td>
</tr>
</tbody>
</table>

The following variables in the Configuration Options file are used to specify the USS group IDs of the groups that are listed here. You do not have to change the default unless there are conflicting assignments for other groups. Check with your security administrator.

Table 27. Option File variables for UNIX System Services Group IDs

<table>
<thead>
<tr>
<th>Options File variable</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>racf_omvs_gid_user</td>
<td>80002</td>
</tr>
<tr>
<td>racf_omvs_gid_oper</td>
<td>80003</td>
</tr>
<tr>
<td>racf_omvs_gid_admin</td>
<td>80004</td>
</tr>
<tr>
<td>racf_omvs_gid_auto</td>
<td>80001</td>
</tr>
<tr>
<td>racf_omvs_gid_super</td>
<td>80005</td>
</tr>
</tbody>
</table>

To associate human operators to these groups, the Configuration Options file provides the \(<\text{racf\_group\_xxxx}>\) variables, where you can specify which operators are members of a group. If you use these variables, the Configuration Assistant automatically generates the appropriate RACF statement in the INGESAF member. Otherwise, your security administrator has to connect these operators to the groups manually.

For example, to associate operator BOB with the INGUSER group and operators GABI and TIM with the INGOPER group, the following variable must be specified in the Configuration Options file:

\[
\text{racf\_group\_user=INGUSER:BOB} \\
\text{racf\_group\_oper=INGOPER:GABI,TIM}
\]

Note: It is not necessary to specify auto operators as the generated definitions in the INGESAF member define each auto operator with a default group as specified in variable \(<\text{racf\_group\_auto}>\).
Any data set access permissions that are required for all operators that are connected to these groups are provided automatically in the INGESAF member.

### Operators

All operators, human and automated operators, are defined and authenticated by an SAF product.

For example, to define a human operator who is called BOB with RACF, the following definition is needed:
- A NetView segment must be created.
  
  ```
  ALU BOB NETVIEW(IC(LOGPROF1) MSGRECVR(NO) CTL(GLOBAL))
  ```
- Data set permissions must be granted.

**Note:** If you use the Configuration Assistant and follow the IBM recommendations, the granting of permissions is accomplished implicitly through group membership and group permissions as defined in the generated INGESAF member. See also the previous subsection.

- (Optional) An OMVS segment must be created if you want to automate UNIX System Services processes
  
  ```
  ALU BOB OMVS(UID(uid) HOME('/u/bob') PROGRAM ('/bin/sh'))
  ```
  
  Where uid is a 1-10 digit integer value. It is the responsibility of your Security Administrator to define the human operators, appropriately.

A human operator might have other related SAF attributes, such as a default group it belongs to, a default data set profile, a TSO segment, and other information that is out of the scope of this document.

**Note:** You do not have to make the definitions for the auto operators yourself. The INGESAF member contains all the RACF commands that are necessary to add a user and set the necessary characteristics. Included is the definition of an OMVS segment and read access to BPX.SUPERUSER for those auto operators that can automate USS processes.

Finally, the SECOPTS.OPERSEC stylesheet option has to be set like follows:

```
SECOPTS.OPERSEC = SAFDEF
```

See also section "Stylesheet Options” on page 171 for more information.

### Commands

All commands and services that can be used by human operators and auto operators are protected by an SAF product.

If you use the Configuration Assistant and follow the IBM recommendations, nothing specific has to be done by you. The INGESAF member contains all the RACF commands necessary to define profiles and permissions on a group, that is, role basis.

For your reference, the profiles that are specified for SAF class NETCMDS are constructed with the following pattern:

```
netid.domain.command
```
The generated statements in the INGESAF member use wildcards. However, you can use wildcards for the variables here, only when the NETCMDS class has generics enabled. To enable generics, through RACF you can use the following command:

```
SETR GENERIC(NETCMDS)
```

The sample profile definitions in the INGESAF member do not allow for the use of all commands (product-provided and user scripts) in general. But because there are many commands that can be considered "safe" in the automation environment, it also grants all defined user roles access to all commands that are not explicitly listed in the INGESAF member. Thus, you avoid having an explicit profile that is defined for each of those commands. With RACF, the definitions in the INGESAF member look like this example:

```
RDEFINE NETCMDS *.*.* UACC(NONE)
RDEFINE NETCMDS *.*.* ID(INGUSER,INGOPER,INGADMIN,INGAUTO,INGSUPER) UACC(READ)
```

The INGESAF member lists profiles that allow for the use of commands that are based on roles. For more information about using the NETCMDS class, see *IBM Tivoli NetView for z/OS: Security Reference*.

The INGESAF member is a sample member that implements the mapping of commands to roles as recommended by the product. Your security administrator can take the generated commands as they are. Or adjust as needed, for example to add or remove certain groups for a particular command.

Finally, the SECOPTS.CMDAUTH stylesheet option has to be set like follows:

```
SECOPTS.CMDAUTH = SAF,PASS
```

See also section "Stylesheet Options" on page 171 for instructions.

### Use of Commands Cross System

All operator commands that are provided by the product supply a TARGET parameter that you can use to run the command on a remote system.

SA *z/OS* is limited to just a small number of systems in the sysplex by default. The INGESAF member contains a definition that explicitly allows the communication between those systems.

However, if you want to limit this capability or completely prevent that commands can be issued on one system but run on another system then more profiles and permission statements are required. The profiles are defined in the SYSAUTO general resource class and constructed according to the following pattern:

```
AGT.sysplexname.saxcfgroup.TARGET.FROMDOM.fromdom.TODOM.todom
```

The variables have the following meanings:

- **sysplexname**
  - This variable denotes the name of the physical sysplex.

- **saxcfgroup**
  - This variable denotes the XCF group name for this particular system. The name always starts with the prefix INGXSG, followed by the value that is specified in the `<sa_xcf_grpid_suffix>` variable in the Configuration Options file.
Use of Commands Cross System

fromdom
This variable denotes the NetView domain on which commands with the
TARGET parameter can be issued.

todom
This variable denotes the NetView domain on which commands with the
TARGET parameter can be run.

You can use wildcards in the profile, when the class SYSAUTO has generics
enabled. To enable generics, with RACF you can use the following command:
SETR GENERIC(SYSAUTO)

The following profile can be defined to prevent execution of commands on the
IPUFA domain. Using RACF, for example, the command looks like the example
here:
RDEFINE SYSAUTO AGT.*.*.TARGET.FROMDOM.*.TODOM.IPUFA UACC(NONE)

In order to allow BOB to run commands on domain IPUFA, the following
permission statement can be used:
PERMIT AGT.*.*.TARGET.FROMDOM.*.TODOM.IPUFA CLASS(SYSAUTO) ID(BOB) ACC(READ)

Note: The read access to such a profile enables execution of a command on a
remote system only when the issuer is authorized to start the command on the
local system.

Note: For the security checks to prevent unauthorized use of commands across
systems, it is important that the SYSAUTO class is activated, a profile exists, and
that the SAF-product is active. If a check fails indicating that any of these
conditions is not met, access is granted, regardless.

Use of Commands from TSO or Batch

You must define profiles if you want to use:
• the batch command interface
• AT overwrite syntax checking of the customization dialog
• Relational Data Services from TSO
• other SA z/OS provided REXX functions running in TSO.

Permissions must be granted also to those profiles for all users that might require
these capabilities.

Note: By default, a command cannot be issued from outside the NetView 3270
console or the system console. The security precaution is enforced when you use
the Configuration Assistant and deploy the definitions in the generated INGESAF
member.

Authorization is granted to all users, if the following checks are passed
successfully:
• Front-end check that basically allows permissions or rejects a user regardless of
the particular command that is used
• Back-end check that performs an authorization check inside the NetView
program on behalf of the TSO or batch user that starts the command
Use of Commands from TSO or Batch

Front-end Checking

The profiles for Front-end checking are defined in the SYSAUTO general resource class and are constructed according to the following pattern:

\texttt{TSO.sysplexname.systemname.CMDRCVR.SEND}

The variables have the following meanings:

\textbf{sysplexname}  
This variable denotes the name of the physical sysplex.

\textbf{systemname}  
This variable denotes the name of the system.

You can use wildcards, when the class SYSAUTO has generics enabled. To enable generics, with RACF you can use the following command:

\texttt{SETR GENERIC(SYSAUTO)}

For example, the following profile can be defined to prevent the issuing of any command from TSO or Batch on the SYS1 system. The RACF syntax would be as follows:

\texttt{RDEFINE SYSAUTO TSO.*.SYS1.CMDRCVR.SEND UACC(NONE)}

To allow BOB to issue commands on the SYS1 system, the following permission statement can be used:

\texttt{PERMIT TSO.*.SYS1.CMDRCVR.SEND CLASS(SYSAUTO) ID(BOB) ACC(READ)}

\textbf{Note:} Read access to such a profile enables issuing of a command on that system only, if the user also passes the back-end check and the NetView command check.

Back-end Checking

The profiles for Back-end checking are defined in the NETCMDS class and constructed as described in \textit{Commands} on page 165. If you use the Configuration Assistant and follow the IBM recommendations, the profiles are already defined for you. The INGESAF member contains all the RACF commands that you need to define profiles and permissions on a group, that is, role basis.

However, unless the TSO user or the user that is associated with the Batch job is already connected to any of the groups that represent different user roles, more definitions are required. See \textit{Roles} on page 163.

\textbf{Batch Command Interface AOFRYCMD/EVJRYCMD with SERVER=*}

- The user must be permitted to use the command EVJRVCMD (Batch only)
- The user must be permitted to use each command that wants to issue (TSO and Batch)

For example, to allow the RUNAUTO job that is associated with user PAUL to issue the INGREQ command, with RACF, the following permission statements as shown are required:

\texttt{PE *.*.EVJRVCMD CLASS(NETCMDS) ID(PAUL) ACC(READ)}
\texttt{PE *.*.INGRYRU0 CLASS(NETCMDS) ID(PAUL) ACC(READ)}

EVJRVCMD is the Batch receiver command name and INGRYRU0 is the true name of the INGREQ command, rather than just the command synonym of INGREQ.
Use of Commands from TSO or Batch

Relational Data Services INGRCRDX

The user must be permitted to use the command INGRCRDS. For example,

PE *.*.INGRCRDS CLASS(NETCMDS) ID(PAUL) ACC(READ)

AT Overwrite Syntax Checking for the Customization Dialog

The user must be permitted to use NetView command PIPE. For example,

PE *.*.PIPE CLASS(NETCMDS) ID(PAUL) ACC(READ)

Notes:

1. To help you finding the true name for a command, search the INGESAF member for the synonym that you are looking for.

2. Read access to such a profile ensures that the user really is authorized to issue the command even though that user might not even be known to the NetView program and the command is instead issued by an auto operator.

3. For the security checks to prevent the unauthorized use of commands from TSO or batch, it is important that a profile exists and that the SAF-product is active. If a check fails indicating that any of these conditions is not met, access is granted, regardless. If this is not what you want, set the advanced automation option of AOF_AAO_SEC_PPIAUTH=FAIL.

4. Read "Step 15B: Install SA Provided Authorized TSO Command INGPAUTH" on page 118 to install INGPAUTH as an authorized TSO command. The TSO REXX functions use INGPAUTH under cover for RACF checking.

Resources

SA z/OS supports to secure resources to a certain degree.

If you want to limit or completely prevent that resources are manipulated then you need to turn on resource level security. This requires you to define profiles and permission statements. There are two types of profiles in the SYSAUTO general resource class.

1. Profiles for resources that are controlled by SA z/OS:

Syntax:

AGT.sysplexname.saxcfgroup.RES.resource_name.resource_type [.resource_location]

The variables have the following meanings:

sysplexname
   This variable denotes the name of the physical sysplex.

.saxcfgroup
   This variable denotes the XCF group name for this particular system. The name always starts with the prefix INGXSG, followed by the value that is specified in the <sa_xcf_grpid_suffix> variable in the INGDOPT Configuration Options file.

resource_name
   This is the name of the System Automation resource (for example, TSO).

resource_type
   This parameter references the type of a resource (for example, APL, APG, MTR, SYG,....).
Resources

resource_location

This optional parameter references the location of a resource (for example, SYS1).

2. Profiles for SA z/OS special resources.

Syntax:
AGT.sysplexname.saxcfgroup.RES.special_res_name[.qualifiers]

The variables have the following meanings:

sysplexname
This variable denotes the name of the physical sysplex.

saxcfgroup
This variable denotes the XCF group name for this particular system. The name always starts with the prefix INGXSG, followed by the value that is specified in the <sa_xcf_grpid_suffix> variable in the INGD0PT Configuration Options file.

special_res_name
This is the name of a 'special' resource which is indicated by a leading underscore (for example, _MANAGER).

qualifiers
These are optional qualifiers for a special resource (for example, .DIAG).

Here are some examples for both kinds of resources:

<table>
<thead>
<tr>
<th>AM Notation</th>
<th>SAF definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application TSO/APL/SYS1</td>
<td>AGT.SYSPLEX1.INGXSG.RES.TSO.APL.SYS1</td>
</tr>
<tr>
<td>Application Group BASE/APG/SYS1</td>
<td>AGT.SYSPLEX1.INGXSG.RES.BASE.APG.SYS1</td>
</tr>
<tr>
<td>Application Group AM_X/APG</td>
<td>AGT.SYSPLEX1.INGXSG.RES.AM_X.APG</td>
</tr>
<tr>
<td>Special Resource _CONFIG</td>
<td>AGT.SYSPLEX1.INGXSG.RES._CONFIG</td>
</tr>
<tr>
<td>Special Resource _MANAGER.DIAG</td>
<td>AGT.SYSPLEX1.INGXSG.RES._MANAGER.DIAG</td>
</tr>
</tbody>
</table>

You can use wildcards in the profile, when the class SYSAUTO has generics enabled. To enable generics, with RACF you can use the following command:

```
SETR GENERIC(SYSAUTO)
```

The affected resource(s) and authority required is determined by looking at the parameters and/or the panel input according to the following table:
## Resources

**Table 28. Resource and Profile Security Relationships**

<table>
<thead>
<tr>
<th>Resources</th>
<th>Profile</th>
<th>Command</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA Resource</td>
<td>UPDATE</td>
<td>INGREQ</td>
<td>Unless CONTROL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INGRUN</td>
<td>REQ=SET (affected SYG)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>REQ=ADD/DEL (what is added/deleted)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INGGROUP</td>
<td>RECYCLE, CANCEL, RESET, DEFAULT, EXCLUDE, AVOID, INCLUDE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INGMOVE</td>
<td>Generally</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INGSET</td>
<td>CANCEL, KILL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INGVOTE</td>
<td>CANCEL, KILL from full screen</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SETSTATE</td>
<td>Generally (note that when START=YES is specified INGREQ does another security check)</td>
</tr>
<tr>
<td>CONTROL</td>
<td>INGREQ</td>
<td></td>
<td>With SCOPE, OVERRIDE, INTERRUPT other than default (IBM supplied or installation)</td>
</tr>
<tr>
<td></td>
<td>INGSET</td>
<td>SET</td>
<td>ACTIVATE, PACIFY, ADKUST</td>
</tr>
<tr>
<td>_MANAGER</td>
<td>UPDATE</td>
<td>INGAMS</td>
<td>SET</td>
</tr>
<tr>
<td></td>
<td>CONTROL</td>
<td>INGAMS</td>
<td>DISABLE, ENABLE, SUSPEND, RESUME</td>
</tr>
<tr>
<td>_MANAGER_DIAG</td>
<td>UPDATE</td>
<td>INGAMS</td>
<td>DIAG REQ = other than STATS</td>
</tr>
<tr>
<td>_CONFIG</td>
<td>UPDATE</td>
<td>INGAMS</td>
<td>REFRESH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INGCLEAN</td>
<td>Generally (note that security checks of ACF REQ=DEL are bypassed)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INGMDFY</td>
<td>Generally (note that security checks of ACF REQ=DEL are bypassed)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ACF</td>
<td>REFRESH, ATLOAD</td>
</tr>
<tr>
<td>CONTROL</td>
<td>ACF</td>
<td></td>
<td>COLD, REQ=DEL, REQ=REPL</td>
</tr>
</tbody>
</table>

## Stylesheet Options

You learned about the role concept, definitions that are required for human operators, and how commands can be secured according to IBM recommendations. There are stylesheet options that are required to implement this level of security.
The CNMSTGEN member, generated by the Configuration Assistant, uses security options that start the Automation Agent before you do any configuration to this member, with the defaults that are provided by the product. However, this level of security is not sufficient and in fact is not secure at all, unless you change the default passwords as explained here.

When you are ready to switch to SAF-based security, in your `<sa_h1q_user>.DSIPARM` data set, edit the CNMSTGEN member and activate the following options:

The first option specifies that operator identification and password or password phrase checking is done with an SAF security product.

```
SECOPTS.OPERSEC = SAFDEF
```

The second option specifies that the NetView component performs command authorization checking with an SAF security product. Users can issue all commands when the SAF product cannot make a security decision. This option avoids the need to define profiles and permissions for all non-critical NetView component commands explicitly.

```
SECOPTS.CMDAUTH = SAF.PASS
```

The third option specifies to check the authority of the original issuer or the ID closest to the original issuer.

```
SECOPTS.AUTHCHK = SOURCEID
```

Make sure, you specify each of the options once and you comment out the default settings in this member.

```
SECOPTS.AUTHCHK = SOURCEID
```

The fourth option specifies that commands routed tasks from the NetView automation table are not authority-checked by a SAF security product, unless `SEC=CH` was specified on the CMDDEF statement.

```
DEFAULTS.AUTOSEC = BYPASS
```

You activate resource level security checks by setting the following stylesheet option in CNMSTGEN:

```
SECOPTS.SARESAUT = ON.PASS
```

The user id used for SAF checking is either OPID() from the top level System Automation command or explicitly set for third party checking (for example, from the PPI Receiver).

### Other Security Options

Table 29 shows you what other optional areas matter in terms of security. Also, where you can find detailed information for setting up your security correctly.

<table>
<thead>
<tr>
<th>Area</th>
<th>Further Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Logger</td>
<td>See “Step 12: Configure the System Logger” on page 100 in Chapter 10, “Traditional</td>
</tr>
<tr>
<td></td>
<td>SA z/OS Configuration,” on page 73 in IBM Tivoli System Automation for z/OS Planning</td>
</tr>
<tr>
<td></td>
<td>and Installation Guide.</td>
</tr>
</tbody>
</table>
### Other Security Options

**Table 29. Information References for Security (continued)**

<table>
<thead>
<tr>
<th>Area</th>
<th>Further Information</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HMC Web API: See the &quot;Step 8: Preparing Ensemble HMC Communication&quot; on page 101 of Chapter 10, &quot;Traditional SA z/OS Configuration,&quot; on page 73 in IBM Tivoli System Automation for z/OS Planning and Installation Guide.</td>
</tr>
</tbody>
</table>

---

**Securing Focal Point Systems and Target Systems**

Your operations staff and automation facilities at both focal point system and target systems need to be authorized to manage the resources in their environment.

You can control human and automation operator authority through the password security provided by either:

- **NetView**
  - Operator definition file (DSIOPF)
  - RODM access information
- **An SAF-based security product such as RACF**
Securing Focal Point Systems and Target Systems

NetView facilities limit the use of commands and keywords to authorized operators and limit an operator’s span of control to specific systems. Access to the SA z/OS graphic interface is controlled by user ID, password, and RODM access information. SA z/OS provides the sample INGESCAT for NetView authorization.

RACF can be used to limit the use of z/OS system commands to authorized operators. SA z/OS provides the sample INGESAF for a RACF environment.

When a target system is in the same sysplex as the focal point system, and your security product supports it, it is recommended that you share security definitions.

Granting NetView and the STC-User Access to Data Sets

This section describes what levels of access authorities you need to assign to NetView and to specific started tasks.

Access to XCF Utilities

The CDS recovery as well as some operator commands use the XCF utilities to retrieve couple data set information. Because the DD name SYSPRINT is required by the utilities, but can also be assigned by NetView for holding log data, the call of the utilities is implemented as a started task in the PROCLIB. The input and output data sets used by the started tasks are dynamically allocated and deleted by the NetView address space. This requires the RACF ALTER access to these data sets for NetView.

When the address space of the started task is created, the operating system assigns a user ID to the started task. User IDs are assigned either by using STARTED profiles or by using the ICHRIN03 table (see z/OS Security Server RACF System Programmer’s Guide). The user ID must have RACF UPDATE authority to the data sets. The data set names are created as follows:

```
hlq.domain.HSAyyddd.Xhhmmss
```

- **hlq** is the high-level qualifier for temporary data set defined during the configuration
- **domain** is the domain ID of the current NetView
- **X** is I, O, or P

Access to HOM Interface

Sometimes after an IPL an operating system does not know its sender paths to the coupling facilities in the sysplex. In this case the automation functions call the HCD HOM interface to determine the missing path information. As the HOM interface must not run authorized the interface is called via a started task. The input and output data sets used by the started tasks are dynamically allocated and deleted by the NetView address space. This requires the RACF ALTER access to these data sets for NetView.

When the address space of the started task is created, the operating system assigns a user ID to the started task. User IDs are assigned either by using STARTED profiles or by using the ICHRIN03 table (see z/OS Security Server RACF System Programmer’s Guide). The user ID must have RACF UPDATE authority to the data sets. The data set names are created as follows:

```
hlq.domain.HSAyyddd.Xhhmmss
```
Granting NetView and the STC-User Access to Data Sets

hlq is the high-level qualifier for temporary data set defined during the configuration

domain is the domain ID of the current NetView

X O or P

Access to IPL Information

The automation function that collects, displays, compares, and deletes IPL information uses two started tasks. It is recommended that you run the first started task immediately after an IPL as part of COMMNDxx list processing to collect the IPL information in the SA z/OS VSAM data set "IPLDATA". The remaining functions are handled by a NetView command. Because the started task and the command can delete IPL information, both need RACF CONTROL access to the VSAM data set. The started task that collects the information needs RACF READ access to all parmlib members.

When a comparison of IPL information is requested, the NetView command schedules the second started task to call ISRSUPC (the compare utility provided by ISPF) because this utility requires a fixed ddname. The input and output data sets that are used by the second started tasks are dynamically allocated and deleted by the NetView address space. This requires RACF ALTER access to these data sets for NetView.

When the address space of the started task is created, the operating system assigns a user ID to the started task (the IBM default is STCUSER). This user ID must have RACF UPDATE access to the data sets. The data set names are created as follows:

hlq.domain.opid.INGPIPLx

Where:

hlq is the high-level qualifier for temporary data set defined during the customization
domain is the domain ID of the current NetView
opid is the NetView operator ID
x L, N, or O

Access to Spare Couple Data Sets

Because the CDS recovery allocates and deletes spare couple data sets via an XCF utility the user ID assigned to the started task address space must also have RACF ALTER access to these couple data sets. The names of the spare couple data sets are built as follows:

hlq.cdstype.Svvvvv

Where:

hlq is the high-level qualifier for couple data sets defined during the configuration
cdstype is ARM, CFRM, LOGR, SFM, SYSPLEX
Svvvvv is the volume name from the list of Alternate Volumes
Granting NetView and the STC-User Access to Data Sets

Access to User-Defined Couple Data Sets
In addition, the user ID of the started task address space needs RACF READ access to all user-defined couple data sets. And, when LOGGER recovery is enabled, the user ID needs RACF ALTER access to the LOGR couple data sets as well.

Access to Spare Local Page Data Sets
The new auxiliary shortage recovery allocates and formats spare page data sets. For this reason NetView requires RACF ALTER access to these page data sets. The names of the spare page data sets are built as follows:

{hlq}.{sysname}.{volume}.{nn}

Where:
- **hlq** is the high-level qualifier for page data sets defined during the configuration
- **sysname** is the name of system for which the data set is allocated
- **volume** is the serial number of the volume on which the data set is allocated
- **nn** is a unique sequence number

Access to JES Spool Output Data Sets
The task INGTJLM processes JES spool output data sets. It runs under the NetView user ID. For this reason, the NetView user ID must be granted READ access to the class JESSPOOL in general or to those data sets in this class which will be monitored. The data set name of a JES spooled data sets is built as follows:

{localnodeid}.{uid}.{jobnm}.{jobid}.{xxx} *

Where:
- **localnodeid** is the NJE node name of the node on which SYSIN or SYSOUT data set currently resides. The **localnodeid** appears in the JES job log of every job.
- **uid** user ID that owns the job
- **jobnm** job name
- **jobid** identifier of the job
- **xxx** may be one of the following:
  - JESMSGGLG
  - JESJCL
  - JESYSMSG
  - Dhhhhhh (h = hexadecimal character)

Access to the NetView UNIX Command Server
If access to the NetView UNIX Command Server is required, it is necessary to define the <sa_hlq_smpe>.SINGMOD1 library to PROGRAM CONTROL and permit the affected tasks appropriately:
Restricting Access to INGPLEX and INGCF Functions

This section describes how you can grant and control access of users to the INGCF and INGPLEX commands.

Access to sensitive functions of the INGPLEX and INGCF commands should be granted to certain operators only. To do this:

- Restrict access to the INGRCCHK command for the keyword INGPLEX or INGCF, and certain given values
- Permit certain operators or groups of operators to access these restricted commands, keywords, and values

To achieve this, use the NetView command authorization table or SAF command authorization.

The following keywords and values are applicable for restricting access to the functions of the INGPLEX and INGCF commands:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Value</th>
<th>Allows for</th>
</tr>
</thead>
</table>
| INGPLEX | CDS   | • Allocating an alternate CDS with the INGPLEX CDS command  
|         |       | • Controlling the SDUMP options and the SLIP traps sysplexwide |
| INGCF   | STR   | • Forcing the deallocation of a CF structure with the INGCF STRUCTURE command  
|         |       | • Rebuilding a CF structure on another CF with the INGCF STRUCTURE command  
|         |       | • Controlling the SDUMP options and the SLIP traps sysplexwide |
Restricting Access to INGPLEX and INGCF Functions

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Value</th>
<th>Allows for</th>
</tr>
</thead>
</table>
| INGCF   | CF    | • Preparing a CF for removal from the sysplex with the INGCF DRAIN command  
|         |       | • Integrating, or reintegrating, a CF into a sysplex with the INGCF ENABLE command  
|         |       | • Including the keyword INGCF with the value STR |
| INGPLEX | HW    | • Deactivating the LPAR of a CF with the INGCF DRAIN command  
|         |       | • Activating the LPAR of a CF (equivalent to starting the Coupling Facility Control Code) with the INGCF ENABLE command  
|         |       | • Including the keyword INGCF with the value CF |

To activate the authorization check via the NetView command authorization table, add the protect and permit statements for the INGRCHK command, the INGPLEX and INGCF keywords and the CDS, STR, CF and HW values as shown in the following example:

```
PROTECT *.*.INGRCHECK.INGPLEX.CDS
PROTECT *.*.INGRCHECK.INGCF.STR
PROTECT *.*.INGRCHECK.INGCF.CF
PROTECT *.*.INGRCHECK.INGPLEX.HW
PERMIT GRP3 *.*.INGRCHECK.INGPLEX.CDS
PERMIT GRP5 *.*.INGRCHECK.INGPLEX.HW
PERMIT GRP3 *.*.INGRCHECK.INGCF.STR
PERMIT GRP4 *.*.INGRCHECK.INGCF.CF
```

With these definitions operators of group GRP3 are authorized to issue all functions of the INGPLEX CDS and the INGCF STRUCTURE commands.

Operators of group GRP4 are authorized to issue all functions of the INGCF CF and the INGCF STRUCTURE commands, but are not authorized for the functions of the INGPLEX CDS commands.

Restricting Access to Joblog Monitoring Task INGJLM

The task INGTJLM processes JES spool output data sets. It runs under the NetView userid.

For this reason, the NetView userid must have read access to the data sets being monitored. However, the permission allows all NetView users to read the spool data, even sensitive data, using the command INGJLM unless the command is restricted. Use the NetView command authorization table (see below) or the equivalent SAF command authorization to restrict the parameters START, STOP, and SUSPEND.

```
PROTECT *.*.INGJLM.START
PROTECT *.*.INGJLM.STOP
PROTECT *.*.INGJLM.SUSPEND
PERMIT grpx *.*.INGJLM.START
PERMIT grpx *.*.INGJLM.STOP
PERMIT grpx *.*.INGJLM.SUSPEND
```
Security for IBM Tivoli Monitoring Products

This section describes security options for controlling access to IBM Tivoli Monitoring products (in particular for OMEGAMON XE) and to OMEGAMON classic monitors.

Please use the following RACF instructions to add the certificates uploaded in "Step 37B: Enabling SOAP over HTTPS for a TEMS" on page 158 to the user's keyring:

\[\text{racdcert id(#saf_user#) addring(<keyring>)}
\]
\[\text{racdcert id(#saf_user#) add ('<UID.ITM.PEM>') WITHLABEL ('ITM') TRUST}
\]
\[\text{racdcert id(#saf_user#) connect (ID(#saf_user#) RING(<keyring>) LABEL('ITM') USAGE(CERTAUTH))}
\]
\[\text{setropts raclist (digtring) refresh}
\]
\[\text{setropts raclist (digtcert) refresh}
\]

\#saf_user# represents the userid authorized for these certificates. If NetView option SECOPTS.OPERSEC is set to SAFDEF, then each human and ISQ* operator must be authorized separately. Otherwise the started task must be authorized.

For more information, refer to "Step 37B: Enabling SOAP over HTTPS for a TEMS" on page 158 for the SSL socket connection.

Controlling Access to IBM Tivoli Monitoring Products

The IBM Tivoli Monitoring (ITM) platform offers a series of Simple Object Access Protocol (SOAP) requests that can be issued from z/OS. SOAP is a communications XML-based protocol that lets applications exchange information through the Internet. For further information about creating SOAP messages, see the appendix “Tivoli Enterprise Monitoring Web services” in IBM Tivoli Monitoring: Administrator’s Guide.

Authentication of users (autotasks or operators) is done based on <userid> and <password> tags that are specified in a SOAP request, if security is enabled. Note, however, that before a SOAP request can be issued the user must be logged on to NetView.

The SOAP request is sent to the hub Tivoli Enterprise Monitoring Server (monitoring server) that is supplied in the INGOMX command and processed there.

SOAP requests can be authorized in terms of both user and hub monitoring server via a user access list. They can be further restricted to groups of users and particular SOAP servers using command authorization table identifiers however final authorization is performed on the hub monitoring server based on the user access list and logon validation.

The relevant keywords that are supported by the INGOMX command are SERVER and IPADDR:

- SERVER allows access based on either the server object that is defined in the SOAP SERVER policy item of a NTW policy object, or a host name. Note that you can only specify the first 8 characters for long host names.
- IPADDR allows access based on IP addresses, however this must be for all IP addresses or none because an address cannot be specified in the command authorization table.
Table 30 on page 181 shows the SA z/OS command names, keywords, and values that can be protected along with their associated SAF resource or command authorization table identifier.

Controlling Access to OMEGAMON Monitors

OMEGAMON provides both product level security and command level security:

- Product level security is applied when users log on to OMEGAMON
- Command level security is applied when users issue commands

A generic SA z/OS user ID must be defined to SAF for external product level security or to OMEGAMON for internal product level security.

For commands that are protected only by internal security, command locking must be enabled for this user ID, based on the command authority level needed by SA z/OS. For example, if only level 0 and 1 commands are issued from SA z/OS, an INITIAL1 rule must be defined and permission must be granted to the generic user, and at the same time there must be no INITIALb rule. In the absence of INITIALbr rules, the command authority level for SA z/OS is always 0. For further details, see the OMEGAMON documentation.

For commands protected by external security, appropriate command resource profiles have to be created and permission must be granted to the generic user.

NetView Command Authorization

Because SA z/OS uses a common user ID that establishes sessions between SA z/OS and any OMEGAMON, SA z/OS uses NetView and the command authorization table to control access to:

- OMEGAMON sessions
- OMEGAMON commands
- The administration of OMEGAMON sessions

For details about the command authorization table, see the NetView Security Reference manual.

The common user ID that is specified with the OMEGAMON session definitions represents the set of users (autotasks, operators) that interact with OMEGAMON sessions. It needs to be defined to OMEGAMON with the highest security level that has been granted to automation. This approach simplifies the configuration that is required in OMEGAMON to permit access to the monitor.

Table 30 on page 181 shows the new SA z/OS command names, keywords, and values that can be protected along with their associated SAF resource or command authorization table identifier.
Table 30. Command Authorization Identifiers

<table>
<thead>
<tr>
<th>Commands and Keywords</th>
<th>Command List Name</th>
<th>SAF Resource or Command Authorization Table Identifier</th>
</tr>
</thead>
</table>
| INGOMX NAME CMD SERVER IPADDR | INGROMX0 | netid.luname.INGROMX0. NAME.session_name  
netid.luname.INGROMX0.CMD.command  
netid.luname.INGROMX0.SERVER.server_name  
netid.luname.INGROMX0.IPADDR |
| INGSESS REQ START STOP | INGRYSS0 | netid.luname.INGRYSS0. REQ  
netid.luname.INGRYSS0.REQ.START  
netid.luname.INGRYSS0.REQ.STOP |

Notes:
1. For OMEGAMON commands that contain a period, replace it with an '@' when defining the command authorization entry, for example, to protect .RMF use:
   
   ```sh
   PROTECT *.*.INGROMX0.CMD.@RMF
   ```

2. If you want to use TRAP for OMEGAMON for IMS, CMD authorization for XIMS must be given and for the other monitors, CMD authorization for EXSY must be given.

Consider adopting the following approach to defining command authorization:

- For maximum security, protect all sessions and all commands.
- Permit access to sessions and commands only as needed.
- Administrators need INGOMX-NAME and INGSESS-REQ authorization.

Password Management

Logging on to OMEGAMON requires authentication with a user ID and password if product level security is active. Note that when a password is specified, it appears in readable format in the automation configuration file and in logs. When SAFPWD is specified, the password is stored in a VSAM data set in an encrypted format.

The SA z/OS command INGPW is used to access the password data set to set or read the password. SA z/OS uses INGPW as follows:

- Passwords are stored and retrieved by user_id and owner_id
- user_id is the common user defined to log on to an OMEGAMON session
- owner_id is a custom value representing one or more VTAM application IDs as defined in the authentication policy
- If no owner is defined for an application ID, it defaults to the 5 leftmost characters of the application ID.

To use SAFPWD, all applications denoted by the OMEGAMON applid that share the same password must be assigned to a single owner. You define the owner in the NETWORK (NTW) entry type with the AUTHENTICATION policy item. On the Authentication Definitions panel enter your definitions in the Owner and Share fields. See “AUTHENTICATION Policy Item” in IBM Tivoli System Automation for z/OS Defining Automation Policy for more details about this panel.
Security for IBM Tivoli Monitoring Products

Authentication Using the SA z/OS Password Data Set: The SA z/OS password data set is used as a password safe if you do not want to reveal passwords in your policy database. The password data set has to be created first and allocated upon the start of NetView. See “Step 2E: SA z/OS Password Store Data Set’ on page 80 for details.

You are responsible for setting the initial password for a user ID with a given owner in the password data set using the SA z/OS command INGPW. Whenever a logon is made to OMEGAMON, for sessions with SAFPW defined as the user password, SA z/OS attempts to look up that user’s password in the password data set. If the lookup succeeds, INGPW returns either the current password or, if the 30-day validity period has expired, the current and a new password. On logging on to OMEGAMON, the current password is used to authenticate the user ID. If a new password is available, the new password is also changed on the OMEGAMON logon screen. Upon successful password update in OMEGAMON, the new password is also updated in the password data set using INGPW. You are responsible for ensuring that the password in the password data set and the password known to SAF or OMEGAMON are the same, in particular when shared SAF databases are used in a multisystem complex, for example, a Parallel Sysplex. In this case, the password data sets should also be shared by the same group of systems.

Use the INGPW command to initialize the password data set. For example, suppose the session and password share definitions are set as in for user oper1 and owner AOMON, the INGPW command format would be:

```
ingpw oper1 AOMON,INIT=pw,MASK=%A%n%n%A%A%A%A%A,EXPINT=0
```

Where pw is the initial password for the user ID and the MASK parameter indicates that the password should be 8 characters long, beginning with a letter, followed by 2 numbers and then 5 letters and never expire.

See IBM Tivoli System Automation for z/OS Operator’s commands for further details about the INGPW command.

Security for Ensemble HTTP Connections

Adding SSL-Certificate to userid’s keyring

Please use the following RACF instructions to add the certificates uploaded in “Step 8B: Setting up AT-TLS for the SSL socket connection” on page 101 to the user’s keyring:

```
 racdcert id(#saf_user#) addring(<tllsKeyring>)
racdcert id(#saf_user#) add (’<UID:HMC.CERT>’) WITHLABEL (’<label>’) TRUST
racdcert id(#saf_user#) connect ID(#saf_user#) RING(<<tllsKeyring>>) +
   LABEL(’<label>’) USAGE(CERTAUTH)
setropts raclist (digtring) refresh
setropts raclist (digtcert) refresh
```

#saf_user# represents the userid authorized for these certificates. If NetView option SECOPTS.OPERSEC is set to SAFDEF, then each human and ISQ* operator must be authorized separately. Otherwise the started task must be authorized.

For RACF users, the following commands would complete the job:
Security for IBM Tivoli Monitoring Products

allowing netview to use the ensemble hardware commands

Each ensemble defined in your SA z/OS policy database must have a corresponding resource profile defined with your SAF product. The skeleton of the ensemble resource is:

```
ISQ.ENS.ensemble
```

The ensemble part of the resource name corresponds with the ensemble entry name definition specified in the customization dialog.

The following example shows how to define an ensemble resource in RACF:

```
SETROPTS CLASSACT(FACILITY)
SETROPTS RACLIST(FACILITY)
RDEFINE FACILITY ISQ.ENS.ENSR35 UACC(NONE)
PERMIT ISQ.ENS.ENSR35 CLASS(FACILITY) ID(stcuser) ACC(ALTER)
```

Levels of ensemble access

The following lists the access levels and their meaning for the ensemble resources:

- **READ**: Retrieve, get configuration information from the ensemble objects
- **CONTROL**: Initialize, discover and terminate the ensemble session
- **ALTER**: Issue operations management commands of the zBX objects:
  - **ACTIVATE**
  - **DEACTIVATE**

Depending on the NetView operator security (OPERSEC) chosen, the access level is checked differently. If your NetView operator security is set to MINIMAL, NETVPW, or SAFPW, the user ID that is checked for hardware access is always the user ID that started the NetView address space, which is usually a STC user ID. This user ID has to be authorized for all ensemble resources you want to manage with this NetView. If multiple users are allowed to start NetView, make sure they are all authorized.

If you have chosen a NetView operator security level of OPERSEC=SAFDEF or OPERSEC=SAFCHECK, several NetView autotasks need to be authorized to access the ensembles that are defined in the customization dialog. Refer to “Defining the CPC Access Lists” on page 186 for further details.

Password Management

Connecting to the ensemble HMC Web Services API requires authentication with a valid HMC user ID and password. Note that when a password is specified, it appears in readable format in the automation configuration file and in logs. When SAFPW is specified, the password is stored in a VSAM data set in an encrypted format. You define the userid and password for ensembles in the ENSEMBLE INFO policy item.

Use the predefined value SAFPW to allow NetView to maintain the password of the user ID.
Security for IBM Tivoli Monitoring Products

See ENSEMBLES Policy Item in IBM Tivoli System Automation for z/OS Defining Automation Policy.

The SA z/OS command INGPW is used to access the password data set to set or read the password.

SA z/OS uses INGPW as follows:
- Passwords are stored and retrieved by user_id and owner_id
- user_id is the common user defined to log on to an ensemble HMC
- owner_id is the name of the entry as used by the SA z/OS dialogs for the zEnterprise ensemble.

Authentication Using the SA z/OS Password Data Set: The SA z/OS password data set is used as a password safe if you do not want to reveal passwords in your policy database. The password data set has to be created first and allocated upon the start of NetView. See “Step 2E: SA z/OS Password Store Data Set” on page 80 for details.

You are responsible for setting the initial password for a user ID with a given owner in the password data set using the SA z/OS command INGPW. The HMC password value must be 4-32 characters long in order to be used with INGPW. Whenever a logon is made to HMC Web Services API, for sessions with SAFPW defined as the user password, SA z/OS attempts to look up that user’s password in the password data set. If the lookup succeeds, INGPW returns either the current password or, if the password validity period has expired, the current and a new password. On logging on to the HMC, the current password is used to authenticate the user ID. If a new password is available, the new password is also changed on the HMC.

Upon successful password update on the HMC, the new password is also updated in the password data set using INGPW. You are responsible for ensuring that the password in the password data set and the password known to the HMC are the same, in particular if you plan to use an alternate focal point. In this case, the password data sets should be shared by the group of systems where focal point can run.

Use the INGPW command to initialize the password data set. For example, suppose the session and password share definitions are set as in for HMC user enosoper1 and owner ENSZBX, the INGPW command format would be:

\[\text{INGPW enosoper1 ENSZBX,INIT=pw,MASK=%A%A%A%A%A%A%A,A,EXPIRINT=0}\]

Where pw is the initial password for the user ID and the MASK parameter indicates that the password should be 8 characters long, beginning with a letter, followed by 2 numbers and then 5 letters and never expire. See IBM Tivoli System Automation for z/OS Operator’s Commands for further details about the INGPW command.

Controlling Access to the Processor Hardware Functions

For processor operations SNMP processor connections, ensemble HTTP connections and for the Parallel Sysplex enhancements functions that use the BCP internal interface, a SAF product such as RACF must be used to define the required resources and grant access to these resources for the authorized NetView users and autotasks.
Allowing NetView to Use the BCP Internal Interface

Before you can use the enhanced sysplex functions of SA z/OS for CF or XCF automation, the hardware resource (HSAET32) must be defined in NetView.

1. Define resource HSA.ET32OAN.HSAET32 in the CLASS FACILITY
2. Permit NetView READ ACCESS to this facility class resource

The following example shows the RACF commands used to define the resource and to grant the required READ access for the NetView user.

```
SETROPTS CLASSACT(FACILITY)
SETROPTS RACLIST(FACILITY)
RDEFINE FACILITY HSA.ET32OAN.HSAET32 UACC(NONE)
PERMIT HSA.ET32OAN.HSAET32 CLASS(FACILITY) ID(stcuser) ACC(READ)
```

With the SETROPTS command, the RACF class FACILITY is made available. With the SETROPTS RACLIST command the FACILITY class resource profile copy in the RACF data space is enabled to increase performance. The next command, RDEFINE, fully qualifies the HSAET32 resource and sets universal access to none. With the PERMIT command, the RACF defined user stcuser gets READ access to this resource. User ID stcuser must be the user ID associated with your NetView started task. If you start NetView as a regular job, the user ID submitting the job must be authorized for the resource.

Note that you can use a wildcard character to specify the resource more generic if that is suitable for your environment.

Access to the CPCS

Each processor (CPC) defined in your SA z/OS policy database must have a corresponding resource profile defined with your SAF product. Note that this only applies for processors defined with a connection type SNMP or INTERNAL.

The skeleton of the CPC resource is:

```
HSA.ET32TGT.netid.nau
HSA.ET32TGT.netid.nau.lpar
```

The netid.nau part of the resource name corresponds with the netid.nau definition of the CPC entry specified in the customization dialog. The period between netid and nau is part of the resource name. For LPAR protection define a resource with the netid.nau.lpar specification.

The following example shows how to define a CPC resource in RACF.

```
RDEFINE FACILITY HSA.ET32TGT.DEIBMD1.X7F1F30A UACC(NONE)
```

The CPC with netid DEIBMD1 and nau X7F1F30A is defined as a resource in the RACF class facility with a universal access attribute of NONE.

Note that you can use a wildcard character to specify the resource more generic if that is suitable for your environment.

Levels of CPC Access

The following lists the access levels and their meaning for the CPC resources:

- **READ**: Retrieve, get configuration information from the CPC
- **WRITE**: Update, set configuration information of the CPC
- **CONTROL**: Issue operations management commands of the CPC
Controlling Access to the Processor Hardware Functions

Note: This access level scheme is for the CPC and its LPARs.

Defining the CPC Access Lists

Depending on the NetView operator security (OPERSEC) chosen, the access level is checked differently. If your NetView operator security is set to MINIMAL, NETVPW, or SAFPW, the user ID that is checked for hardware access is always the user ID that started the NetView address space, which is usually a STC user ID. This user ID has to be authorized for all CPC and CPC.Lpar resources you want to manage with this NetView. If multiple users are allowed to start NetView, make sure they are all authorized.

If you have chosen a NetView operator security level of OPERSEC=SAFDEF or OPERSEC=SAFCHECK, the following paragraph applies.

With SA z/OS, several NetView autotasks need to be authorized to access the CPCs that are defined in the customization dialog.

The following NetView autotasks need to be authorized with access level CONTROL for all defined CPCs and all its LPARs:

- The XCF and RPC autotasks
- The autotasks defined with SYN %AOFOPXCFCOPER% and %AOFOPRPCOPER% in automation table member AOFMSGSY
- The hardware interface autotasks AUTHWxxx
- Any operator issuing a hardware action with INGCF

The AUTXCFxx autotasks plus the additional ones from %AOFOPXCFCOPER% are used internally once INGCF drain or INGCF enable is invoked by an authorized user. IXC102A message automation is also performed by these autotasks.

The autotasks used for the hardware interface initialization and communication also need to be authorized. Use access level CONTROL for the AUTHWxxx autotasks in your environment.

The following example shows how to permit access to a CPC resource in RACF:

```plaintext
PERMIT HSA.ET32TGT.DEIBMD1.X7F1F30A CLASS(FACILITY) ID(AUTXCF) ACC(CONTROL)
```

The XCF autotask AUTXCF gets access level CONTROL for the CPC resource DEIBMD1.X7F1F30A.

LPAR access example:

```plaintext
PERMIT HSA.ET32TGT.DEIBMD1.X7F1F30A.* CLASS(FACILITY) ID(AUTXCF) ACC(CONTROL)
```

The XCF autotask AUTXCF gets access level CONTROL for the CPC resource DEIBMD1.X7F1F30A and all its defined logical partitions.

Implementing Granular Hardware Access

By giving operators READ access to a CPC resource and CONTROL access only to LPARs according to the business needs, a flexible security scheme can be implemented.

Password Management for SNMPv3 HMC/SE Connections

Connecting to a HMC or SE using SNMP Version 3 protocol requires authentication with predefined SNMPv3 user name and password. Note that when
Controlling Access to the Processor Hardware Functions

A password is specified in the SA z/OS PROCESSOR INFO policy, it appears in readable format in the automation configuration file and in logs.

When SAFPW is specified, the password is stored in a VSAM data set in an encrypted format. You define the SNMPv3 user name and password on the HMC or SE (see [Step 7A: Preparing the HMC (Console Workplace 2.9 and Later Versions)](#) and [“Step 7B: Preparing the SE (Console Workplace 2.9 and Later Versions)” on page 98] and specify them in the PROCESSOR INFO policy item for processors using the SNMP connection protocol. Use the predefined value SAFPW to allow NetView to maintain the password of the user name. See PROCESSOR policy item in IBM Tivoli System Automation for z/OS Defining Automation Policy.

The SA z/OS command INGPW is used to access the data set to set or read the password. SA z/OS uses INGPW as follows:
- Passwords are stored and retrieved by user_id and owner_id
- user_id is the SNMPv3 user name defined for CPC
- owner_id is the ProcOps Target HW Name of the CPC as used by the SA z/OS customization dialogs for the CPC.

Authentication using the SA z/OS Password Data Set: The SA z/OS password data set is used as a password safe if you do not want to reveal passwords in your policy database. The password data set has to be created first and allocated upon the start of NetView. See [“Step 2E: SA z/OS Password Store Data Set” on page 80](#) for further details.

You are responsible for setting the password for a SNMPv3 user name with a given owner in the password data set using the SA z/OS command INGPW. The SNMPv3 password must be 8-31 characters long in order to be used with INGPW.

Whenever an SNMP session is established to SE or HMC, for sessions with SAFPW defined as the user password, SA z/OS attempts to look up that user's password in the password data set. If the lookup succeeds, INGPW returns the current password, otherwise the connection fails. You are responsible for ensuring that the password in the password data set and password known to the SE or HMC are the same, in particular if you plan to use an alternate focal point. In this case, the password data sets should be shared by the group of systems where the focal point can run.

Use the INGPW command to initialize the password data set. For example:
NETVASIS INGPW v3testuser T99PRO, INIT=SNMPV3PWD, MASK=%A%A%A%A%A%A%A%A, EXPINT=0

Note: System z API does not support remote SNMPv3 password change, therefore do not use automatic password expiration feature of the INGPW command (use EXPINT=0 or omit the parameter). See IBM Tivoli System Automation for z/OS Operator’s Commands for further details about the INGPW command.
Defining a RACF Profile for I/O Operations

Assign authorization levels using RACF/SAF for individual commands or generically for all commands. Use the RACF RDEF command with a class of FACILITY.

<table>
<thead>
<tr>
<th>Function</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>To define the profile for the PROHIBIT command</td>
<td>RDEF FACILITY IHV.PROHIBIT</td>
</tr>
<tr>
<td>To define a profile that would allow all users to enter a command (for example, UNLOCK)</td>
<td>RDEF FACILITY IHV.UNLOCK UACC.READ</td>
</tr>
<tr>
<td>To permit the use of generics for a Class of Service facility</td>
<td>SETROPTS GENERIC FACILITY</td>
</tr>
<tr>
<td>To prevent unauthorized use of commands you can enter this RACF command to prohibit use of commands</td>
<td>RDEF FACILITY IHV.* UACC(NONE)</td>
</tr>
</tbody>
</table>

**Note:** If you have prohibited all user IDs from using these commands, you must explicitly assign RACF authorization to designated user IDs.

### Assign RACF Authorization

To give RACF authorization to a user ID, enter the RACF PERMIT command and its parameters.

### Assign a Profile Parameter

The profile parameter is IHVcommandname, where:
- IHV. is the three-character ID, followed by a period (.)
- commandname is the name of the command

**Notes:**
1. The profile parameter (for example, IHVALLOW, IHV.VARY, IHV.REMOVE.SWITCH) determines the authorization level of the user ID identified in the ID parameter.
2. The ACCESS parameter identifies the authorization given.
   - You can use an asterisk to designate a generic class on the PERMIT parameters. For example, to allow all users to send all commands that require read authority, enter:
     PERMIT IHV.* ACCESS(READ) CLASS(FACILITY) ID(*)

### Assign Authorization by ACCESS Level

You can authorize a user ID to enter one command at a given access level by entering one command.

For example, to allow a user (SUWAJDA) to send commands requiring control authorization, enter:
PERMIT IHV.* ACCESS(CONTROL) CLASS(FACILITY) ID(SUWAJDA)

For example, to authorize another user (FISHER) to enter all commands that require the update authorization, enter:
PERMIT IHV.* ACCESS(UPDATE) CLASS(FACILITY) ID(FISHER)
Assign Authorization by Command

You can use the PERMIT command to let all users send individual commands. For example, to authorize everyone to use the Unlock command, enter:

```
PERMIT IHV.UNLOCK ACCESS(READ) CLASS(FACILITY)
ID(*)
```

To authorize a user (DONC) to send all connectivity commands with the Noforce option, enter:

```
PERMIT IHV.* ACCESS(UPDATE) CLASS(FACILITY)
ID(DONC)
```

Use Specific Profile Names

Either specific profile names or generic profile names can be used in the PERMIT command. Use specific profile names to authorize use of specific I/O operations commands.

For example, to authorize a user (PHILOP) to use only the Allow and Prohibit commands with the Noforce option, enter:

```
PERMIT IHV.ALLOW ACCESS(UPDATE) CLASS(FACILITY) ID(PHILOP)
PERMIT IHV.PROHIBIT ACCESS(UPDATE) CLASS(FACILITY) ID(PHILOP)
```

For more information on the RACF commands, see Resource Access Control Facility (RACF) Command Language Reference.

On the NMC focal point the following is necessary to define users and access levels to RODM:

1. Define a general resources class named RODMMGR. This is the default class name used in EKGCUST initialization member for RODM.
2. Define instances of the RODMMGR resource class, for example,

   ```
   RDEF EKGXRODM1 CLASS(RODMMGR) UACC(NONE)
   RDEF EKGXRODM2 CLASS(RODMMGR) UACC(NONE)
   RDEF EKGXRODM3 CLASS(RODMMGR) UACC(NONE)
   RDEF EKGXRODM4 CLASS(RODMMGR) UACC(NONE)
   RDEF EKGXRODM5 CLASS(RODMMGR) UACC(NONE)
   RDEF EKGXRODM6 CLASS(RODMMGR) UACC(NONE)
   ```

Assign TCP/IP Port Authorization

When the physical file system is configured as INET, RACF can be used to restrict access to the ports that are used by I/O operations when using TCP/IP communication. For details about how to restrict access, see the section “Port access control” in “Chapter 3. Security” of z/OS Communications Server: IP Configuration Guide.

Access Authorization Levels

Table 31 on page 190 lists the I/O operations commands in alphabetical order with an indication of the access authorization levels they require and where they can be used.

Notes:

1. Access authorization is not required, although it is highly advisable.
2. The authorization level is not checked for I/O operations commands that are entered at the system console. The control level of authorization is assumed.
3. RACF profiles are defined in class FACILITY.
In Table 31, the access authorization level that is required by the command can be:

<table>
<thead>
<tr>
<th>Access Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Control</td>
</tr>
<tr>
<td>R</td>
<td>Read</td>
</tr>
<tr>
<td>U</td>
<td>Update</td>
</tr>
<tr>
<td>UC</td>
<td>Update for the NOForce</td>
</tr>
</tbody>
</table>

The following indicate where the command can be used and what RACF authorization is required:

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>The function is generated implicitly, but RACF authorization, or equivalent, is required if I/O operations is used together with a security program.</td>
</tr>
<tr>
<td>I</td>
<td>A command function is generated implicitly as part of a G function. RACF authorization, or equivalent, does not apply.</td>
</tr>
<tr>
<td>Y</td>
<td>The command must be entered explicitly. RACF authorization is required.</td>
</tr>
<tr>
<td>–</td>
<td>The command is not available, either explicitly or implicitly.</td>
</tr>
</tbody>
</table>

**Table 31. I/O Operations Commands, their Availability and Access Authorization Levels**

<table>
<thead>
<tr>
<th>RACF Profile Name</th>
<th>Access Authorization Level</th>
<th>Application Programming Interface (API)</th>
<th>Switch Configuration Matrix (I/O Operations)</th>
<th>ISPF Dialog</th>
<th>I/O Operations ISPF Dialog Command</th>
<th>System Console</th>
</tr>
</thead>
<tbody>
<tr>
<td>IHV.AUTO</td>
<td>UC</td>
<td>Y</td>
<td>I</td>
<td>Y</td>
<td>ALLOW</td>
<td></td>
</tr>
<tr>
<td>IHV.BLOCK</td>
<td>UC</td>
<td>Y</td>
<td>I</td>
<td>Y</td>
<td>BLOCK</td>
<td></td>
</tr>
<tr>
<td>IHV.CHAIN</td>
<td>UC</td>
<td>Y</td>
<td>–</td>
<td>Y</td>
<td>CHAIN</td>
<td></td>
</tr>
<tr>
<td>IHV.CHANGECHECK</td>
<td>R</td>
<td>Y</td>
<td>–</td>
<td>Y</td>
<td>DISPLAY CHANGECHECK</td>
<td></td>
</tr>
<tr>
<td>IHV.CHP</td>
<td>R</td>
<td>Y</td>
<td>–</td>
<td>Y</td>
<td>DISPLAY CHP</td>
<td></td>
</tr>
<tr>
<td>IHV.CONNECT</td>
<td>UC</td>
<td>Y</td>
<td>I</td>
<td>Y</td>
<td>CONNECT</td>
<td></td>
</tr>
<tr>
<td>IHV.DELETE.FILE</td>
<td>C</td>
<td>Y</td>
<td>–</td>
<td>–</td>
<td>DELETE FILE</td>
<td></td>
</tr>
<tr>
<td>IHV.DEV</td>
<td>R</td>
<td>Y</td>
<td>–</td>
<td>Y</td>
<td>DISPLAY DEV</td>
<td></td>
</tr>
<tr>
<td>IHV.DISCONNECT</td>
<td>UC</td>
<td>Y</td>
<td>I</td>
<td>Y</td>
<td>DISCONNECT</td>
<td></td>
</tr>
<tr>
<td>IHV.GETLOCK</td>
<td>C</td>
<td>Y</td>
<td>–</td>
<td>Y</td>
<td>GETLOCK</td>
<td></td>
</tr>
<tr>
<td>IHV.HOST</td>
<td>R</td>
<td>Y</td>
<td>–</td>
<td>Y</td>
<td>DISPLAY HOST</td>
<td></td>
</tr>
<tr>
<td>IHV.LOGREC</td>
<td>C</td>
<td>Y</td>
<td>–</td>
<td>Y</td>
<td>LOGREC</td>
<td></td>
</tr>
<tr>
<td>IHV.NAME</td>
<td>R</td>
<td>Y</td>
<td>I</td>
<td>Y</td>
<td>DISPLAY NAME</td>
<td></td>
</tr>
<tr>
<td>IHV.PORT</td>
<td>R</td>
<td>Y</td>
<td>I</td>
<td>Y</td>
<td>DISPLAY PORT</td>
<td></td>
</tr>
<tr>
<td>IHV.PROHIBIT</td>
<td>UC</td>
<td>Y</td>
<td>I</td>
<td>Y</td>
<td>PROHIBIT</td>
<td></td>
</tr>
<tr>
<td>IHV.QUERY.ENTITY</td>
<td>R</td>
<td>Y</td>
<td>–</td>
<td>–</td>
<td>QUERY ENTITY</td>
<td></td>
</tr>
<tr>
<td>IHV.QUERY.FILE</td>
<td>R</td>
<td>Y</td>
<td>G</td>
<td>–</td>
<td>QUERY FILE</td>
<td></td>
</tr>
<tr>
<td>IHV.QUERYINTERFACE</td>
<td>R</td>
<td>Y</td>
<td>–</td>
<td>–</td>
<td>QUERY INTERFACE</td>
<td></td>
</tr>
<tr>
<td>IHV.QUERY.RELATION</td>
<td>R</td>
<td>Y</td>
<td>–</td>
<td>–</td>
<td>QUERY RELATION</td>
<td></td>
</tr>
<tr>
<td>IHV.QUERY.SWITCH</td>
<td>R</td>
<td>Y</td>
<td>G</td>
<td>–</td>
<td>QUERY SWITCH</td>
<td></td>
</tr>
<tr>
<td>IHV.REMOVE.CHP</td>
<td>UC</td>
<td>Y</td>
<td>–</td>
<td>Y</td>
<td>REMOVE CHP</td>
<td></td>
</tr>
<tr>
<td>IHV.REMOVE.DEV</td>
<td>UC</td>
<td>Y</td>
<td>–</td>
<td>Y</td>
<td>REMOVE DEV</td>
<td></td>
</tr>
<tr>
<td>IHV.REMOVE.SWITCH</td>
<td>UC</td>
<td>Y</td>
<td>–</td>
<td>Y</td>
<td>REMOVE SWITCH</td>
<td></td>
</tr>
<tr>
<td>IHV.RESET.CHANGECHECK</td>
<td>C</td>
<td>Y</td>
<td>–</td>
<td>Y</td>
<td>RESET CHANGECHECK</td>
<td></td>
</tr>
<tr>
<td>IHV.RESET.HOST</td>
<td>C</td>
<td>Y</td>
<td>–</td>
<td>Y</td>
<td>RESET HOST</td>
<td></td>
</tr>
<tr>
<td>IHV.RESET.SWITCH</td>
<td>C</td>
<td>Y</td>
<td>–</td>
<td>Y</td>
<td>RESET SWITCH</td>
<td></td>
</tr>
<tr>
<td>IHV.RESET.TIMEOUT</td>
<td>C</td>
<td>Y</td>
<td>–</td>
<td>Y</td>
<td>RESET TIMEOUT</td>
<td></td>
</tr>
<tr>
<td>IHV.RESTORE.CHP</td>
<td>U</td>
<td>Y</td>
<td>–</td>
<td>Y</td>
<td>RESTORE CHP</td>
<td></td>
</tr>
<tr>
<td>IHV.RESTORE.DEV</td>
<td>UC</td>
<td>Y</td>
<td>–</td>
<td>Y</td>
<td>RESTORE DEV</td>
<td></td>
</tr>
<tr>
<td>IHV.RESTORE.SWITCH</td>
<td>UC</td>
<td>Y</td>
<td>–</td>
<td>Y</td>
<td>RESTORE SWITCH</td>
<td></td>
</tr>
</tbody>
</table>
### Defining RACF Profile for I/O Operations

Table 31. I/O Operations Commands, their Availability and Access Authorization Levels (continued)

<table>
<thead>
<tr>
<th>RACF Profile Name</th>
<th>RACF Authorization Level</th>
<th>Application Programming Interface (API)</th>
<th>Switch Configuration Matrix (I/O Operations ISPF Dialog)</th>
<th>I/O Operations ISPF Dialog Command</th>
<th>System Console Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>IHV.RESULTS</td>
<td>R</td>
<td>Y</td>
<td>–</td>
<td>Y</td>
<td>DISPLAY RESULTS</td>
</tr>
<tr>
<td>IHV.SWITCH</td>
<td>R</td>
<td>Y</td>
<td>I</td>
<td>Y</td>
<td>DISPLAY SWITCH</td>
</tr>
<tr>
<td>IHV.SYNC.SWITCH</td>
<td>C</td>
<td>Y</td>
<td>–</td>
<td>Y</td>
<td>SYCH SWITCH</td>
</tr>
<tr>
<td>IHV.TIMEOUT</td>
<td>R</td>
<td>Y</td>
<td>–</td>
<td>Y</td>
<td>DISPLAY TIMEOUT</td>
</tr>
<tr>
<td>IHV.UNBLOCK</td>
<td>UC</td>
<td>Y</td>
<td>I</td>
<td>Y</td>
<td>UNBLOCK</td>
</tr>
<tr>
<td>IHV.UNCHAIN</td>
<td>UC</td>
<td>Y</td>
<td>–</td>
<td>Y</td>
<td>UNCHAIN</td>
</tr>
<tr>
<td>IHV.UNLOCK</td>
<td>U</td>
<td>Y</td>
<td>–</td>
<td>Y</td>
<td>UNLOCK</td>
</tr>
<tr>
<td>IHV.VARY</td>
<td>R</td>
<td>Y</td>
<td>–</td>
<td>Y</td>
<td>DISPLAY VARY</td>
</tr>
<tr>
<td>IHV.WRITE</td>
<td>C</td>
<td>Y</td>
<td>I</td>
<td>Y</td>
<td>WRITE</td>
</tr>
<tr>
<td>IHV.WRITEFILE</td>
<td>C</td>
<td>Y</td>
<td>G</td>
<td>–</td>
<td>WRITEFILE</td>
</tr>
<tr>
<td>IHV.WRITEPORT</td>
<td>UC</td>
<td>Y</td>
<td>–</td>
<td>–</td>
<td>WRITEPORT</td>
</tr>
<tr>
<td>IHV.WRITESWCH</td>
<td>UC</td>
<td>Y</td>
<td>G</td>
<td>–</td>
<td>WRITESWCH</td>
</tr>
</tbody>
</table>

Table 32 lists the access authorization levels grouped by function (display, connectivity or utility).

Table 32. Access Authorization Levels Grouped by Function

<table>
<thead>
<tr>
<th>RACF Profile Name</th>
<th>RACF Authorization Level</th>
<th>Application Programming Interface (API)</th>
<th>Switch Configuration Matrix (I/O Operations ISPF Dialog)</th>
<th>I/O Operations ISPF Dialog Command</th>
<th>System Console Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>System commands to start/stop</td>
</tr>
</tbody>
</table>

**Display Commands**

| IHV.CHANGECHECK   | R                        | Y                                        | –                                                      | Y                                | DISPLAY CHANGECHECK   |
| IHV.CHP           | R                        | Y                                        | –                                                      | Y                                | DISPLAY CHP           |
| IHV.DEV           | R                        | Y                                        | –                                                      | Y                                | DISPLAY DEV           |
| IHV.HOST          | R                        | Y                                        | –                                                      | Y                                | DISPLAY HOST          |
| IHV.NAME          | R                        | Y                                        | I                                                      | Y                                | DISPLAY NAME          |
| IHV.PORT          | R                        | Y                                        | I                                                      | Y                                | DISPLAY PORT          |
| IHV.RESULTS       | R                        | Y                                        | –                                                      | Y                                | DISPLAY RESULTS       |
| IHV.SWITCH        | R                        | Y                                        | I                                                      | Y                                | DISPLAY SWITCH        |
| IHV.TIMEOUT       | R                        | Y                                        | –                                                      | Y                                | DISPLAY TIMEOUT       |
| IHV.VARY          | R                        | Y                                        | –                                                      | Y                                | DISPLAY VARY          |
| IHV.QUERY.ENTITY  | R                        | Y                                        | –                                                      | –                                | QUERY ENTITY          |
| IHV.QUERY.FILE    | R                        | Y                                        | G                                                      | –                                | QUERY FILE            |
| IHV.QUERYINTERFACE| R                        | Y                                        | –                                                      | –                                | QUERY INTERFACE       |
| IHV.QUERY.RELATION| R                        | Y                                        | –                                                      | –                                | QUERY RELATION        |
| IHV.QUERY.SWITCH  | R                        | Y                                        | G                                                      | –                                | QUERY SWITCH          |

**Connectivity Commands**
### Defining RACF Profile for I/O Operations

#### Table 32. Access Authorization Levels Grouped by Function (continued)

<table>
<thead>
<tr>
<th>RACF Profile Name</th>
<th>RACF Authorization Level</th>
<th>Application Programming Interface (API)</th>
<th>Switch Configuration Matrix (I/O Operations)</th>
<th>I/O Operations ISPF Dialog Command</th>
<th>System Console Command</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>I/O Operations ISPF Dialog Command</td>
<td>I/O Operations ISPF Dialog Command</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IHV.ALLOW</td>
<td>UC</td>
<td>Y I Y Y</td>
<td>ALLOW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IHV.PROHIBIT</td>
<td>UC</td>
<td>Y I Y Y</td>
<td>PROHIBIT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IHV.BLOCK</td>
<td>UC</td>
<td>Y I Y Y</td>
<td>BLOCK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IHV.UNBLOCK</td>
<td>UC</td>
<td>Y I Y Y</td>
<td>UNBLOCK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IHV.CHAIN</td>
<td>UC</td>
<td>Y - Y Y</td>
<td>CHAIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IHV.UNCHAIN</td>
<td>UC</td>
<td>Y - Y Y</td>
<td>UNCHAIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IHV.CONNECT</td>
<td>UC</td>
<td>Y I Y Y</td>
<td>CONNECT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IHV.DISCONNECT</td>
<td>UC</td>
<td>Y I Y Y</td>
<td>DISCONNECT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IHV.REMOVE.CHP</td>
<td>UC</td>
<td>Y - Y Y</td>
<td>REMOVE CHP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IHV.RESTORE.CHP</td>
<td>U</td>
<td>Y - Y Y</td>
<td>RESTORE CHP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IHV.REMOVE.DEV</td>
<td>UC</td>
<td>Y - Y</td>
<td>REMOVE DEV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IHV.RESTORE.DEV</td>
<td>UC</td>
<td>Y - Y</td>
<td>RESTORE DEV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IHV.REMOVE.SWITCH</td>
<td>UC</td>
<td>Y - Y Y</td>
<td>REMOVE SWITCH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IHV.RESTORE.SWITCH</td>
<td>UC</td>
<td>Y - Y Y</td>
<td>RESTORE SWITCH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IHV.SYNC.SWITCH</td>
<td>C</td>
<td>Y - Y Y</td>
<td>SYNCH SWITCH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IHV.WRITEPORT</td>
<td>UC</td>
<td>Y - -</td>
<td>WRITEPORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IHV.Writeswch</td>
<td>UC</td>
<td>Y G -</td>
<td>WRITESWCH</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Utility Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>RACF Authorization Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IHV.DELETE.FILE</td>
<td>C</td>
<td>DELETE FILE</td>
</tr>
<tr>
<td>IHV.WRITEFILE</td>
<td>C</td>
<td>WRITEFILE</td>
</tr>
<tr>
<td>IHV.GETLOCK</td>
<td>C</td>
<td>GETLOCK</td>
</tr>
<tr>
<td>IHV.UNLOCK</td>
<td>U</td>
<td>UNLOCK</td>
</tr>
<tr>
<td>IHV.LOGREC</td>
<td>C</td>
<td>LOGREC</td>
</tr>
<tr>
<td>IHV.RESET.CHANGECHECK</td>
<td>C</td>
<td>RESET CHANGECHECK</td>
</tr>
<tr>
<td>IHV.RESET.HOST</td>
<td>C</td>
<td>RESET HOST</td>
</tr>
<tr>
<td>IHV.RESET.SWITCH</td>
<td>C</td>
<td>RESET SWITCH</td>
</tr>
<tr>
<td>IHV.RESET.TIMEOUT</td>
<td>C</td>
<td>RESET TIMEOUT</td>
</tr>
<tr>
<td>IHV.WRITE</td>
<td>C</td>
<td>WRITE</td>
</tr>
</tbody>
</table>

#### Establishing Authorization with Network Security Program

If you have installed Network Security Program (NetSP), you can create an authorization system requiring only one sign on for each user.

With it, a user who logs on from a workstation has access to RACF-protected host applications. These include 3270 emulation and log on scripts and APPC communications. This authorization is controlled by NetSP's PassTicket, which is recognized by the SAF-based security system and is valid for a fixed period of time.
Establishing Authorization with Network Security Program

To establish authorization for your users, you need to create in NetSP recorded input files as log on transfer scripts. This is done either by recording keystrokes in the emulator session or by entering them directly in a file with a text editor. How to do this is described in Network Security Product Secured Network Gateway Guide.

Security for WebServices Adapter

Authentication of the WebServices Adapter

Any HTTP client that sends a HTTP request to the System Automation WebServices Adapter must authenticate itself by a userid and password or by a previously created session id.

Basic Authentication

The System Authentication HTTP Adapter supports basic authentication with a SAF userid and password.

Thus the client must provide in the HTTP input request header the keyword: Authorization

which contains the BASE64 encoded userid and password. After decoding it has the form:
userid:password

The user and password is used by the System Automation WebServices Adapter in order to authenticate the user through SAF (for example, RACF).

Program-Controlled Libraries

If facility class BPX.DEAMON is defined in RACF, the dynamic load libraries that are used by the automation adapter must be RACF program-controlled. This can be done for modules of the HFS by setting the extended attribute $p$. During SMP/E installation the extended attribute $p$ is set automatically for all libraries (*.so) and ingwto within the automation adapter lib directory. If this is not the case, you may set it using the extattr shell command with the $p$ option. You may also have to define C++ and LE-provided runtime libraries, such as SCERUN, SCERUN2 and SCLBDLL, to be RACF program-controlled.

Coexistence of Basic or Session ID Authentication

The basic authentication mechanism takes place for each input request except if the session id is specified in the HTTP request header. In this case the authentication mechanism switches automatically to session id validation.

The request will be rejected with HTTP status code 401 if:
- the session id is available but the session id is invalid
- the session id is not available and basic authentication validation fails

Session Id Authentication

Authentication with a session id takes place once per session. The session id is unique for this session and is a placeholder for a previously performed basic authentication. A session id might have an expiration time, for example 600
Authorization of the WebServices Adapter

Any HTTP request sent to the System Automation WebServices Adapter might be rejected due to missing authorization. The purpose of authorization is to check whether an already authenticated user has the permission to perform the HTTP request for the specified resource on System Automation.

The System Automation Security rules will apply to the System Automation command and related System Automation resources that correspond to the HTTP request.

The userid used for any SAF checking is always taken from the HTTP basic authentication, refer to "Authentication of the WebServices Adapter" on page 193.

Basic Authorization

A userid that sends a HTTP request to System Automation WebServices Adapter must fulfill following 3rd-party SAF security check:

```plaintext
class NETCMDS
permission READ
profile: netid.netvdom.INGRHTTP
```

The basic authorization is sufficient to perform any HTTP GET request.

Extended Authorization

Extended authorization is needed when a URL needs to modify a System Automation resource.

The userid that sends a HTTP PUT request to the System Automation WebServices Adapter must fulfill following additional SAF prerequisites depending on the requested action.

Start, stop, or cancel a resource:

```plaintext
class NETCMDS
permission READ
profile: netid.netvdom.INGRHYU0
```

Move a resource home, away, or to another system:

```plaintext
class NETCMDS
permission READ
profile: netid.netvdom.INGRYMVX
```

Resource Authorization

System Automation resource authorization is needed when a URL needs to modify a System Automation resource. The userid that sends a HTTP PUT request to System Automation WebServices Adapter with action start, stop, cancel or move a resource must fulfill the following SAF prerequisites:
<table>
<thead>
<tr>
<th>class</th>
<th>SYSAUTO</th>
</tr>
</thead>
<tbody>
<tr>
<td>permission</td>
<td>UPDATE</td>
</tr>
<tr>
<td>profile:</td>
<td>AGT.sysplex.xcfgrp.RES.name.type.system</td>
</tr>
</tbody>
</table>
Security for WebServices Adapter
Chapter 12. Configuring SA z/OS Workstation Components

This chapter contains information about how to install those parts of SA z/OS that are required on workstations:

- "Configuring the NMC Workstation"
- "Configuring IBM Tivoli Netcool/OMNIbus" on page 204
- "Configuring Tivoli Service Request Manager through Tivoli Directory Integrator" on page 205
- "Configuring the AssemblyLines"

The workstation components can be installed on any workstation that meets the requirements listed in Chapter 1, "SA z/OS Prerequisites and Supported Equipment," on page 3. One or more workstations can be installed for users to monitor and control the systems that are being managed with SA z/OS.

The code for the SA z/OS NMC exploitation is supplied with the host code that is installed using SMP/E. Installing the SA z/OS NMC exploitation will enable you to issue the most important SA z/OS processor operations and system operations commands from all NMC workstations.

Note: The NMC installation described in "Configuring the NMC Workstation" is performed on the NMC Server and the NMC clients. After this installation, you need to restart the individual NMC clients.

Configuring the NMC Workstation

If you already have an NMC environment installed, you can continue with the actions described in the remainder of this section. Having completed these, you can use the SA z/OS NMC exploitation as described in IBM Tivoli System Automation for z/OS User’s Guide. This will enable you to issue a selection of SA z/OS processor operations and system operations commands from all NMC workstations.

The following packed files for the SA z/OS NMC exploitation are available after your SMP/E installation:

- ING.SINGPWS1(INGNMCZP):
  INGNMCZP is the packed file for Windows. Download it with the extension ZIP and unpack with an appropriate tool (WINZIP or PKZIP).
- ING.SINGPWS1(INGNMCTZ):
  INGNMCTZ is the SA z/OS workstation code for AIX®, UNIX and z/Linux workstations.

1. Download the member (INGNMCTZ) from the data set on the host system to your workstation in binary mode using, for example, FTP.
2. Rename INGNMCTZ to INGNMCTZ.tar.gz on the workstation.
3. Uncompress with the command:
Configuring the NMC Workstation

```
    gzip --decompress --verbose INGNMCTZ.tar.gz
```

4. Unpack with the command:
```
    tar --extract --verbose --file=INGNMCTZ.tar
```

This creates the subdirectory INGNMCEX on the workstation.

- **ING.SINGPWS1(INGNMCTZ):** Japanese version of the packed file for Windows workstations
  If you use the Japanese version of SA z/OS download this file with extension ZIP and unpack with an appropriate tool (WINZIP or PKZIP).

- **ING.SINGPWS1(INGNMCTZ):** Japanese version of the packed file for UNIX workstations
  If you use the Japanese version of SA z/OS download this file with extension TAR.Z and unpack and uncompress with an appropriate tool (uncompress and tar).

The content of each packed file is divided into support for system operations and processor operations commands. Both packages include two NMC response files. One response file contains the system operations commands, the other one contains the processor operations commands. The response files include the definitions and profiles for:

- **ING.SO_OPER**
  SystemOperation Operator

- **ING.PO_OPER**
  ProcessorOperation Operator

- **ING.SA_OPER**
  SystemAutomation Operator (definition for both the system operations and processor operations commands)

Furthermore there are two subdirectories for the related data definition files and two subdirectories with the online help in HTML format.

With this separation of system operations and processor operations commands you may install either the system operations commands or the processor operations commands or both depending on your needs. The installation must be done manually, because there is no common installation tool for the several supported platforms. This requires that you are familiar with the common commands of your workstation operating system.
Configuration Steps on the NMC Server

Perform the following steps to install SA z/OS NMC exploitation on the NMC Server (note that the term UNIX in the following steps refers to all forms of UNIX derivatives, including AIX, z/Linux, and so forth.):

1. Download the appropriate packed file in binary format to the NMC Server.
2. Unpack the file into a temporary directory of the NMC Server, using an appropriate tool for the NMC Server operating system. You will obtain the directory structure for the unpacked files as shown in Figure 14.
3. Copy the required help files as follows:

<table>
<thead>
<tr>
<th>Environment</th>
<th>From Directory</th>
<th>To Your Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>WIN</td>
<td><code>tmp\INGNMCEX\ING_NMCC_HELP</code> and/or <code>tmp\INGNMCEX\ISQ_NMCC_HELP</code></td>
<td><code>[BINDIR]\TDS\server\db\current\help</code></td>
</tr>
<tr>
<td>UNIX</td>
<td><code>tmp/INGNMCEX/ING_NMCC_HELP</code> and/or <code>tmp/INGNMCEX/ISQ_NMCC_HELP</code></td>
<td><code>$BINDIR/TDS/server/db/current/help</code></td>
</tr>
</tbody>
</table>

Where `tmp` stands for the directory that you downloaded the files to.

Note: BINDIR is an environment variable that is set by your NMC installation and indicates that is a subdirectory of your installed NMC product. For example:

`usr\local\Tivoli\bin\w32-ix86\`

4. Copy the required data definition files as follows:
Configuring the NMC Workstation

<table>
<thead>
<tr>
<th>Environment</th>
<th>From Directory</th>
<th>To Your Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>WIN</td>
<td>tmp\INGNMCEX\ING_NMCC_DDF and/or tmp\INGNMCEX\ISQ_NMCC_DDF</td>
<td>[BINDIR]\TDS\server\config\ddf\c</td>
</tr>
<tr>
<td>UNIX</td>
<td>tmp/INGNMCEX/ING_NMCC_DDF and/or tmp/INGNMCEX/ISQ_NMCC_DDF</td>
<td>$BINDIR/TDS/server/config/ddf/c</td>
</tr>
</tbody>
</table>

5. Copy the required response files from INGNMCEX as follows:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environment</td>
<td>To Your Directory</td>
</tr>
<tr>
<td>WIN</td>
<td>[BINDIR]\TDS\server\sample</td>
</tr>
<tr>
<td>UNIX</td>
<td>$BINDIR/TDS/server/sample</td>
</tr>
</tbody>
</table>

6. Copy the Java archive file INGNMCDial.jar from INGNMCEX as follows:

<table>
<thead>
<tr>
<th>Environment</th>
<th>From Directory</th>
<th>To Your Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>WIN</td>
<td>tmp\INGNMCEX</td>
<td>[BINDIR]\TDS\server\db\current\lib</td>
</tr>
<tr>
<td>UNIX</td>
<td>tmp/INGNMCEX</td>
<td>$BINDIR/TDS/server/db/current\lib</td>
</tr>
</tbody>
</table>

7. Verify the following:
   a. To operate the NMC Server you must be logged on to NetView via a 3270 host session.
   b. Your NetView user ID must have NGMF administrator rights.
   c. The NMC Server must be started and active.
   d. The connection from the NMC Server to NetView must be established.

8. Start the Command Profile Editor batch utility (CPEBATCH) with:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>For the following environments Command/Parameters</td>
<td></td>
</tr>
<tr>
<td>WIN</td>
<td>• [BINDIR]\TDS\server\sample\ING_NMCS_CMD.RSP and/or • [BINDIR]\TDS\server\sample\ISQ_NMCS_CMD.RSP and the -i and -g parameters.</td>
</tr>
<tr>
<td>UNIX</td>
<td>• $BINDIR/TDS/server/sample/ING_NMCS_CMD.RSP and/or • $BINDIR/TDS/server/sample/ISQ_NMCS_CMD.RSP and the -i and -g parameters</td>
</tr>
</tbody>
</table>

With this step, you load the delivered commands into the NetView internal database. For more information about how to use this batch utility, refer to NetView Management Console User’s Guide. For a detailed description of how to maintain and manipulate response files for the NMC topology server, go to:

9. Use the Command Profile Editor batch utility (CPEBATCH) to apply the new profiles installed in 8 to apply individual operators defined in your installation. Only these operators that are linked to one of the SA z/OS profiles can execute
SA z/OS commands. No other operators can display or execute SA z/OS commands. For more details on CPEBATCH refer to the appendix 'Topology Server Commands' in Tivoli NetView for z/OS: NetView Management Console User's Guide.

Notes:

a. The NetView CPE online utility was retired with NetView 5.1 Installations that are running NetView 1.4 can still use the CPE online utility to modify the definitions. The CPE online utility was never available for UNIX installations.

b. The recommended way to maintain definitions such as operators, profiles, and so forth is to use the tool delivered in the INGRSPTOOL.ZIP file. The tool comes with a detailed description. It can be downloaded from:

Configuration Steps on the NMC Client

You must have the NetView 3270 Management Console installed if you want to use full screen commands. See NetView Management Console User's Guide for information about how to do this.

Note: You cannot use full screen commands when the NMC focal point is a satellite installation. Use line mode commands instead. More details can be found in the ingnmce/ex/readme.txt mentioned above.

1. Set the environment variable TCONSOLE_CLASSPATH:
   a. For WIN environments pointing to:
      [NMC_Client_Installation_path]\TDS\client\lib\INGNMCJDial.jar
   b. For UNIX environments pointing to:
      [NMC_Client_Installation_path]/TDS/client/lib/INGNMCJDial.jar

   See Figure 15 on page 203 for a sample batch file.

2. On the individual NMC Clients: Restart your NetView Management Console to incorporate your changes.

3. Customize the NetView 3270 Management Console. Execute these steps only if you use full screen commands:
   a. On the NMC, select an SA z/OS resource from an existing view. For this resource, select an SA z/OS command that needs to be transferred to the NetView 3270 Management Console, for example, the INGVOTE_FS command. Click on INGVOTE_FS to display the NetView 3270 Management Console, which does not show any output yet.
   b. Select Session Services from the NMC menu bar, and choose Add/Delete/Modify Session from the menu items. This opens the Add/Delete/Modify Session window.
   c. In the Full Screen Session Name field of this window type: SA
   d. In the Start command String field type, for example: window date (You can enter any valid NetView command.)
   e. Select the radio button Immediate.
   f. From the Session Options select Start Automatically.
   g. Click the Add button, then the Save button to save your changes.
   h. Click the Done button to exit this window.
   i. In the NMC, select the newly-added SA pull-down choice from the Session Services menu bar item.
Configuring the NMC Workstation

j. To verify the customization, issue the INGVOTE_FS command to display the desired output.

Sample to Start the NMC (for Windows NT Environment)
Configuring the NMC Workstation

@rem ***************************************************
@rem IBM System Automation for z/OS NetView Management Console Exploitation
@rem Sample Program - 5645-006
@rem (C) Copyright IBM Corp. 2004
@rem All rights reserved.
@rem
@rem SAMPLE PROGRAM - NO WARRANTY EXPRESSED OR IMPLIED
@rem
@rem You are hereby licensed to use, reproduce, and distribute these sample
@rem programs as your needs require. IBM does not warrant the suitability or
@rem integrity of these sample programs and accepts no responsibility for their
@rem use for your applications. If you choose to copy and redistribute
@rem significant portions of these sample programs, you should preface such
@rem copies with this copyright notice.
@rem ***************************************************
@rem
@rem PRODUCT (System Automation for z/OS)
@rem COMPONENT (NMC Exploitation)
@rem FIRST_RELEASE (V2R1)
@rem LAST_CHANGE (11Jan2002)
@rem
@rem MODULE_NAME (ingnmncst.bat)
@rem DESCRIPTIVE_NAME (Start the NMC Topology Console)
@rem ***************************************************
@rem
@rem Function: This sample shows how the NMC Topology Console can be
@rem started. This sample was written for the Windows NT environment
@rem and NMC 1.3.0.1.
@rem
@rem Usage:
@rem
@rem - The following is a sample which will NOT properly work until customer
@rem installation specific data is provided.
@rem
@rem - Adapt the drive and path statements to reflect your installation
@rem environment.
@rem
@rem - This example assumes that the NMC Topology Console was installed on
@rem drive E:
@rem
@rem - A good location to put this file is the directory:
@rem E:\usr\local\Tivoli\bin\generic_unix\TDS\client\bin
@rem
@rem - Call this file from a icon on your desktop or from Windows
@rem Start-Programs-Netview-... pull-down or from the command line.
@rem ***************************************************

@setlocal
@rem Changes the user's current working directory to the 'bin' directory in
@rem the "base" console installation path.
E:
\usr\local\Tivoli\bin\generic_unix\TDS\client\bin
@set TIVOLI=e:\usr\local\Tivoli\bin\generic_unix\Tds
@set INGJAR=\client\lib\INGNMCJDial.jar
@set FLBJAR=\ibmflb\jars\tivflb13.jar
set TCONSOLE_CLASSPATH=%TIVOLI%FLBJAR%;%TIVOLI%INGJAR%
tconsoleNT.bat .. -key nmc
@endlocal

Figure 15. Sample to Start the NMC (for WIN Environment)
Configuring IBM Tivoli Netcool/OMNibus

Because SA z/OS uses Tivoli Event Integration Facility (EIF) events for communication you need the following components:

- IBM Tivoli Netcool/OMNibus (OMNibus)
- The OMNibus Probes Library for Nonnative Base
- The Tivoli EIF Probe (EIF Probe)

It is assumed that you have all of the above installed and verified before you begin with the customization for SA z/OS. For details please see the product manuals.

For more information about the infrastructure on host systems, refer to "Step 16: Configure Alert Notification for SA z/OS" on page 118.

Although OMNibus can run on various operating systems the following example describes the installation and customization on Windows 2003 Server.

1. Download the sample files ING_event.rules and ING_db_update.sql from the host system to your workstation as text files:
   a. To download the files, you can use, for example, FTP. Choose as the target path name any directory where you want to store temporarily the sample files:
      cd <PATH>
   b. Start FTP with:
      ftp <hostname>
   c. You will be prompted to enter your user ID and password. After logging on to your z/OS system, enter:
      ascii
get /usr/lpp/ing/dist/OMNibus/ING_event.rules
get /usr/lpp/ing/dist/OMNibus/ING_db_update.sql
quit

2. Inspect ING_db_update.sql. This file creates new columns in your ObjectServer’s alert.status table that will later hold the information from the SA z/OS events. It will also add some triggers and a trigger group. Normally you should not have to change this file.

3. Update the alert.status table of your ObjectServers:
   a. Run the SQL processor:
      %OMNIHOME%\bin\redist\isql.exe -S <server> -U <username> -P <password> -i <PATH>/ING_db_update.sql
   b. Repeat the previous step for each ObjectServer.

4. Adapt your EIF probe tivoli_eif.rules. There are two possibilities:
   - Your Tivoli EIR Probe is for SA z/OS events only so you can simply replace the original rules file with the one supplied by SA z/OS:
     copy ING_event.rules C:\Program Files\IBM\Tivoli\Netcool\omnibus\probes\win32\tivoli_eif.rules
   - Otherwise you must merge the logic of ING_event.rules into your existing tivoli_eif.rules

5. Restart your ObjectServers and your EIF Probe.

**Configuring the Triggers**

ING_db_update.sql installs a trigger called ing_count_events. This trigger is designed to prevent multiple lines to be displayed for multiple occurrences of the same event. Instead of that it maintains a counter that is increased each time the
same event arrives repeatedly. The ing_count_events trigger is initially disabled because the installation process of the EIF Probe installs another trigger called deduplication. If you have both triggers enabled your event counter will be increased twice.

You should proceed based on the following options:

- Your EIF Probe is for SA z/OS events only: It is recommended that you have ing_count_events enabled and deduplication disabled.
- Your EIF Probe is also for other events: You must review both triggers and merge the logic.
- You want to see all occurrences of an event as a separate line: You must disable both triggers.

Note that you can manipulate the triggers in IBM Tivoli Netcool/OMNibus Administrator by connecting to your ObjectServers and selecting Automation > Triggers.

**Configuring the Event View**

The event views of IBM Tivoli Netcool/OMNibus Conductor can be customized to show the fields that have been newly inserted into the alert.status table for SA z/OS events. In the event view select Edit > Edit View.

A recommended setup is:
- Node
- AlertGroup
- Summary
- Tally
- INGEventDate
- INGEventTime
- INGEventResName
- INGEventResType
- INGEventResSystem
- INGEventJobname

**Note:** SA z/OS uses the OMNibus event class 89320. Make sure that you define this class.

**Configuring Tivoli Service Request Manager through Tivoli Directory Integrator**

Because SA z/OS integrates with IBM Tivoli Service Request Manager (TSRM) through IBM Tivoli Directory Integrator (TDI) you need the following components:
- TSRM and all prerequisite software
- TDI Runtime Server and Config Editor

It is assumed that you have all of the above installed and verified before you begin with the customization for SA z/OS. For details see the product manuals.

To create a trouble ticket from SA z/OS in TSRM there are no adaptations required in TSRM. Everything is done in TDI. Although TDI can run on various operating systems the following example describes the installation and customization on Windows 2003 Server.

Download the sample file ING_event.xml from the host system to your workstation as a text file:
Configuring TSRM through TDI

1. To download the file, you can use, for example, FTP. Choose as the target path name any directory where you want to store temporarily the sample files:
   ```shell
cd <PATH>
```
2. Start FTP with:
   ```shell
   ftp <hostname>
   ```
3. You will be prompted to enter your user ID and password. After logging on to your z/OS system, enter:
   ```shell
   ascii
   get /usr/lpp/ing/dist/TDI/ING_event.xml
   quit
   ```

Configuring the AssemblyLines

To perform the steps described in this section you should be familiar with the TDI Config Editor. A good overview can be found in *IBM Tivoli Directory Integrator User’s Guide*.

The sample file `ING_event.xml` defines two AssemblyLines:
- `TicketServer` that receives a request from SA z/OS, starts TicketWriter and returns a response
- `TicketWriter` that parses the request and creates a trouble ticket in TSRM

Note that if you have a different service desk than TSRM you can adapt TicketWriter to feed your application. TicketServer can remain the same.

Because they are samples, the AssemblyLines will probably not work unchanged in your environment. You should review both and make any necessary adaptations:
1. Start the TDI Config Editor and open `<PATH>ING_event.xml`.
2. Modify the AssemblyLine `TicketServer` as follows:
   a. Open TicketServer and select the **Data Flow** tab.
   b. Open the ReadXML component in the **Feeds** section.
   c. Adapt the port number. This is a TCP Connector working in server mode. The **Config** tab shows the port number that the server listens to. A value of 8000 is provided in the sample but you are free to change it.
   d. Leave the other components unchanged.
   e. Start the Ticketserver
3. Modify the AssemblyLine `TicketWriter`:
   a. Open TicketWriter and select the **Data Flow** tab.
   b. Modify how the details text is generated:
      1) Review all of the components with names like `Map...Description`.
      2) The `FixDescription` and `SpecificDescription` attributes are set to text that is formatted with the attributes that are mapped by the `Map...Attributes` components. You can adapt the text your needs here.
   c. Modify the TSRM settings:
      1) Open the `WriteTicket` component. This is a Generic Maximo Connector.
      2) Adapt the TSRM communication settings. Select the **Config** tab. Specify various options that must match your TSRM installation:

<table>
<thead>
<tr>
<th>Configuration Tab</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEA Server</td>
<td>specify the URL (server address port) of your TSRM</td>
</tr>
</tbody>
</table>
Configuring TSRM through TDI

<table>
<thead>
<tr>
<th>Configuration Tab</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEA Objects</td>
<td>specify a setting such as the external system name and the names of the Web services for CREATE, DELETE, QUERY and UPDATE operations</td>
</tr>
<tr>
<td>MEA Advanced</td>
<td>leave as is.</td>
</tr>
</tbody>
</table>

d. On the **Output Map** tab TicketWriter sample maps DESCRIPTION and DESCRIPTION_LONG DESCRIPTION, as well as REPORTEDPRIORITY, URGENCY and IMPACT. You can also use this tab to map fixed installation-dependent values. The sample maps the REPORTEDBY user ID to the value SAZOS. You may want to change this or add other user IDs, or do both.
Part 3. Appendixes
Appendix A. Planning for the NMC Environment

The information in this section helps you to plan the configuration of the components in your NMC environment.

NMC Exploitation Topology

Figure 16 shows how in a SA z/OS configuration the involved components communicate to produce graphical output information:

1. At initialization time, the SA z/OS topology manager knows the target systems for automation.
2. The SA z/OS topology manager contacts the SA z/OS topology agents on all sysplexes or stand-alone systems or, for processor operations, it contacts the processor operations focal point to obtain the required information.
3. The SA z/OS topology agents contact the related automation managers or the processor operations component respectively to find out the status from the systems and resources.
4. Then the SA z/OS topology agents report this information to the SA z/OS topology manager on the focal point.
5. The SA z/OS topology manager feeds the RODM data base with the achieved information.
6. The NMC workstation on the operator's request can retrieve the RODM data to produce the defined views.
7. Also, at initialization time, the automation managers get the order to inform the related SA z/OS topology agents whenever status changes occur. Then the
NMC Exploitation Topology

SA z/OS topology agents will route the status change information to the SA z/OS topology manager which will update the RODM data base.

Planning to Install the NMC Workstation

Make sure that you have a working NMC environment with the required functions (for example, RODM, GMFHS, NMC Topology Server, NMC Topology Console, NMC 3270 Management console), as part of your NetView installation available.

For information about how to install the NMC, refer to Tivoli NetView for z/OS Installation: Configuring Graphical Components and NetView Management Console User’s Guide. The information about what to do to enable your NMC environment installation for use in SA z/OS is described in “Configuring the NMC Workstation” on page 197.

If you plan to use Kanji support for NMC keep in mind that all the NetView workstations in the domain must support the character set you decide to use. Multilingual support is not available.

Running Multiple NetViews

If you use two NetViews and you want to monitor resources using the NMC workstation, bear in mind that the NMC workstation must be linked to NetView Graphic Monitor Facility Host Subsystem (GMFHS) on the Networking NetView which has a connection to RODM. See Figure 18 on page 213. You can operate network and SA z/OS resources via RODM and have SA z/OS running in another NetView to control the automation resources. This, however, requires a subset of SA z/OS, referred to as the SA z/OS satellite, to be installed on the Networking NetView. See “Step 28: Install an SA z/OS Satellite” on page 138 for details.

If you run the Networking Automation NetView only on the focal point, you cannot have your resources automated by SA z/OS.

If you run the System Automation NetView only on the focal point, you cannot have networking resources in RODM, but only SA z/OS resources that you automate.

Alternatively, you can run both the Networking Automation and the System Automation on the same NetView. This way, you can save storage and CPU costs because of the reduction in the duplication of, for example, tasks and logs. But more important, it reduces maintenance and system programmer costs. See Figure 17 on page 213 for details.

In such an environment all functions are handled by that NetView. You may want to give the individual NetView tasks different priorities, for example, the System Automation tasks need to run above the VTAM’s priority, whereas others (Networking Automation) need to run at a lower priority. This is achieved with z/OS Workload Manager Enclaves support.
Running Multiple NetViews

Figure 18 illustrates the flow of data from a target system to the focal point when two NetViews are used on the focal point: one for Networking Automation and one for System Automation.

1. The target system data is sent to the Networking NetView at the focal point via Command Handler or Alerts; the AAO AOFSENDALERT will dictate which forwarding mechanism is used. (Alerts from processor operations are sent directly to the Automation NetView).

2. The satellite z/OS automation (focal point) receives the data that is sent from the targets and updates objects in RODM appropriately.

3. NetView Graphic Monitor Facility Host Subsystem (GMFHS) becomes aware of status updates.

4. GMFHS broadcasts updates to the operator workstation.

When an operator initiates a command or routine from a workstation, the action flows back to the Networking NetView for processing in the reverse direction from that shown in Figure 18.
Running Multiple NetViews
Appendix B. Syntax for INGTOPOF File

The INGTOPOF file contains configuration information for the SA z/OS topology manager. It must reside in any of the data sets allocated under the DSIPARM concatenation. The records of the file consist of a keyword with one or more parameters. Comment lines must start with an asterisk (*). A '+' at the end of a line indicates that the record is continued in the next line.

The following keywords can occur in the INGTOPOF file: SYSPLEX, PROCOPS, LOCATION, ANCHOR, BLDVIEWS, OPTION, and TEMPLATE.

The SYSPLEX Statement

For every sysplex, the SA z/OS topology manager must be told which systems of the sysplex are able to communicate with it. This is done with the SYSPLEX statement according to the following format:

```
SYSPLEX sysplex_name domain_id
```

The `sysplex_name` must be different from every name that you specify in a PROCOPS statement (see “The PROCOPS Statement” on page 216). The systems must be identified to the SA z/OS topology manager by their NetView domain ID. If the `net_id` is omitted, it is assumed to be the same as that of the focal point. The INGTOPOF file must contain at least one SYSPLEX statement; in particular, you cannot have a PROCOPS statement in the INGTOPOF file without a SYSPLEX statement.

The SA z/OS topology manager tries to contact the systems in the order in which they appear in the list. When it finds a system that contains a functional SA z/OS topology agent, it searches no further, but gathers the SA z/OS information from the automation manager through this SA z/OS topology agent. It then stores the retrieved information in RODM, prefixing all resource names with the `sysplex_name` that it found in the SYSPLEX statement.

It follows from this that the order in which the domains are specified should reflect eventual decisions about primary and backup systems for communication with the SA z/OS topology manager. Also, the sysplexes as defined in the INGTOPOF file must correspond to the sysplex groups in the policy database.

Because standalone systems are treated as sysplexes, they must also be introduced to the SA z/OS topology manager by a SYSPLEX statement. In this case, the list of domain IDs will comprise just one item.

If you want to have a network anchor for a system, this system’s domain ID must be included in the SYSPLEX statement.
The PROCOPS Statement

With this statement, you specify a focal point for processor operations and its backup focal point. It has the following format:

```
PROCOPS procops_name focal_point backup_focal_point
```

The `procops_name` must be different from every name that you specify in a SYSPLEX statement. The focal point processor and its backup must be identified to the SA z/OS topology manager by a NetView domain ID. If the `net_id` is omitted, the SA z/OS topology manager assumes it to be identical to that of its own focal point.

There must be at least one SYSPLEX statement in the INGTOPOF file if you want to insert a PROCOPS statement.

The LOCATION Statement

The LOCATION statement is used to group system related events, for example, geographically rather than logically. The events that are attached to a LOCATION must be posted to the SA z/OS topology manager by the user with the INGPOST command. For more information on the INGPOST command, see *IBM Tivoli System Automation for z/OS Operator's Commands*.

The Location statement has the following format:

```
LOCATION target_domain location_name
```

Examples:

```
* TSCF1 thru 3 are in Boeblingen, 4 and 5 are in Perth
* LOCATION T2 BB_LAB
LOCATION NETOZ.CNMT4 PERTH
LOCATION NETOZ.CNMT5 PERTH
* AOCA thru D are in Boeblingen
* LOCATION AOCPLEX BB_LAB
* OZ1 thru OZ4 are in Perth
* LOCATION OZPLEX PERTH
```

The ANCHOR Statement

ANCHORS are entered via the customization dialogs on the target systems. For more information about how to define anchors see *IBM Tivoli System Automation for z/OS Defining Automation Policy*.

The ANCHOR statement will remain in the INGTOPOF to allow ANCHORS to be defined for downlevel systems where ANCHORS are not entered via the customization dialogs.

ANCHORS for downlevel systems will occur in RODM, but not in the automation manager.
The ANCHOR statement serves to define anchors for arbitrary user defined events.

Anchors serve to collect events of a certain type that are to be displayed on the NMC. Anchors play the role of major resources for events of this type, and the events themselves are treated as minor resources of their anchor. The SA z/OS topology manager automatically creates anchors for heartbeats but not for WTOR or tape mount requests. For more information on anchors and events see *IBM Tivoli System Automation for z/OS User's Guide*.

With the ANCHOR statement, you can introduce your own anchors for any events. These events must be posted to the SA z/OS topology manager with the INGPOST command; the anchor must be specified in the command as the major resource (RESOURCE parameter). For more information on the INGPOST command, see *IBM Tivoli System Automation for z/OS Operator's Commands*; for information on major and minor resources, see *IBM Tivoli System Automation for z/OS Defining Automation Policy*.

### The BLDVIEWS Statement

A RODM resource can only be displayed on the NMC when it is included in a view. With the BLDVIEWS statement, you can pass data sets (members) that contain view definitions for BLDVIEWS to the SA z/OS topology manager. The SA z/OS topology manager will then call the BLDVIEWS tool for (all or some of) these data sets (members) in order to build or rebuild the specified views. The view definitions must be supplied by the installation.

Every BLDVIEWS statement associates one sysplex (as defined by a SYSPLEX statement) or one processor operations focal point configuration (as defined by a PROCOPS statement) with a list of such data sets (members). This enables the SA z/OS topology manager to rebuild views at runtime only for those sysplexes (sets of target processors) whose SA z/OS information has in fact changed.

The BLDVIEWS statement has the following format:

```
  BLDVIEWS sysplex_name data_set_or_member procops_name
```

You can exploit the association of the data sets (members) to sysplexes to reduce the overhead caused by rebuilding views at runtime. Suppose, for example, that all your sysplex views either contain objects from only one sysplex or from all sysplexes. Then you should proceed as follows.

1. For every sysplex, create a separate data set (member) with the view definitions specific for that sysplex.
2. Create one data set (member) for the common views.
3. Code a BLDVIEWS statement for every sysplex, where the list of data sets (members) comprises two items, namely the data set (member) with the views specific for this sysplex, and the data set (member) with the common views.

In this way, the sysplex specific views are rebuilt only when the SA z/OS resources for the sysplex in question have changed in RODM in such a way that a rebuild is necessary.
Syntax for INGTOPOF File

For more details on view definitions, see IBM Tivoli System Automation for z/OS User’s Guide.

The OPTION Statement

With the OPTION statements you can:
- control whether or not dependencies and major/minor resource relationships are stored in RODM, and are therefore represented on the NMC, and
- specify the default layout for the automatically generated subviews of group objects.

A separate OPTION statement is required for each option.

The OPTION statement has the following format:

```
  >> OPTION
    PARENTS
    NOPARENTS
  >> OPTION
    LINKMINOR
    NOLINKMINOR
  >> OPTION
    MDL_RADLINK
    MDL_GRID
```

The parameters have the following meaning:

**PARENTS**
Dependency relationships are stored in RODM (and displayed on the NMC in network views). This is the default.

**NOPARENTS**
Dependency relationships are not stored in RODM.

**LINKMINOR**
Relationships between major and minor resources are stored in RODM (and displayed in network views). This is the default.

**NOLINKMINOR**
Relationships between major and minor resources are not stored in RODM.

**MDL_RADLINK**
The automatically created subviews are radially arranged. This is the default. The default for this option is defined in RODM.

**MDL_GRID**
The automatically created subviews are arranged in a grid. The default for this option is defined in RODM.

If you want to use the default values, no explicit OPTION statement is required.
The TEMPLATE Statement

The name displayed beneath a resource on the NMC is the DisplayResourceName field of the resource. It can be customized using the TEMPLATE parameter in the INGTOPOF file. The template entries in the INGTOPOF file control how the DisplayResourceName of a resource is formatted.

When using the locate function on the NMC, it is the DisplayResourceName field of the resource that is compared with the search criteria of the locate for an exact match.

It is not a requirement to have any template parameters in the INGTOPOF file. If no template parameter is found in INGTOPOF, the format of the DisplayResourceName will default to the following:

- PLEX.SYSTEM.TYPE.SUBSYSTEM EVENT for major resources
- PLEX.SYSTEM.TYPE.SUBSYSTEM.MINOR EVENT for minor resources

To change the format of the default DisplayResourceName, special type templates are required to specify how the default DisplayResourceName should be formatted. There are the following two types:

- DRN for major resources
- DRNM for minor resources

Customization of the DisplayResourceName can be defined for all resource types (DRN, DRNM), or individually for each resource type (APL, APLM, APG, APGM).

When a resource is created, the type (for example, APL or APG) of the resource is searched for in the INGTOPOF file to find a matching template.

- If a match is found, the DisplayResourceName will be formatted as specified by the type template in the INGTOPOF file.
- If no match is found, the DisplayResourceName will be formatted using the default.

The major resource types supported by the template parameter in the INGTOPOF file are:

- APL applications
- APG application groups
- APGP application groups (sysplex)
- SYS system
- SYG system groups
- GRP groups
- MTR monitor resources

Because minor resources can be attached to major resources, the following types are also supported by the template parameter in the INGTOPOF file for minor resources:

- APLM application minors
- APGM application group minors
Syntax for INGTOPOF File

APGPM
  application group (sysplex) minors

SYSM  system minors

SYGM  system group minors

GRPM  group minors

HEARTBEATM  heartbeat minors

WTORM  WTOR minors

TAPEM  tape minors

CFM  coupling facility minors

CDSM  coupled data set minors

ETRM  external timer minors

SYSPLEXM  sysplex minors

MTRM  monitor resource minors

All, any, or none of the above type templates can be used.

When user defined anchors are created, the following applies:

- If the format of the default DisplayResourceName is acceptable, no additional template will be required in the INGTOPOF file. The DisplayResourceName is formatted using the default.
- If you customized the format of the DisplayResourceName, it is necessary to create a template for the user-defined anchors, to specify how the DisplayResourceName must be formatted for the user-defined anchors and any minor resources attached to the user-defined anchors.

If the anchor statement ANCHOR k1 USER exists in the INGTOPOF file, define the following two type templates in the INGTOPOF file to control the formatting of the DisplayResourceName for the anchor and any attached minor resources:

- USER for the anchor
- USERM for the minor resources attached to the anchor

To define how the DisplayResourceName is formatted, substitution parameters are employed. Substitution parameters can appear in any order. The following substitution parameters are supported:

&STR.
  system.type.subsystem

&RES.
  subsystem/type/system

&MNR.
  minor resource name (minor resources only)
Syntax for INGTOPOF File

%SUB.
  subsystem
%TYP.
  type
%SYS.
  system
%EVT.
  event
%PLX.
  sysplex
%DATE.
  date
%TIME.
  time

If event (%EVT.) is specified as a substitution parameter and no event field exists for the resource, an * is inserted in the DisplayResourceName. If a substitution field does not exist for a resource where a substitution parameter has been specified, the substitution parameter itself (for example, %SYS. &STR.) will appear in its place in the DisplayResourceName.

Examples

Customizing DisplayResourceName for APLs:

If the requested DisplayResourceName for APLs was system name and subsystem name (for example, SYSX.RODMX), the following entry would be required in the INGTOPOF file:
 TEMPLATE APL %SYS..%SUB.

Customizing DisplayResourceName for all resources:

If the requested DisplayResourceName for all resources was system name, subsystem name, and event (for example, SYSX.RODMX Event Text), the following entry would be required in the INGTOPOF file:
 TEMPLATE DRN %SYS..%SUB. %EVT.

Customizing DisplayResourceName for user anchors:

The following anchor statement is found in INGTOPOF file:
 ANCHOR K1 PLEX1

If the requested DisplayResourceName was subsystem, date, and time (for example, RODMX 19 MAY 2002.02:16:45), the following entry would be required in the INGTOPOF file:
 TEMPLATE PLEX1 %SUB. %DATE..%TIME.

The above examples are for major resources. If customization of the DisplayResourceName is also required for minor resources attached to the major resources, similar template entries in the INGTOPOF file would be required:
 • TEMPLATE APLM %SYS..%SUB..%MNR.
 • TEMPLATE DRNM %SYS..%SUB..%MNR. %EVT.
Syntax for INGTOPOF File

- TEMPLATE PLEXIM &SUB..&MNR. &DATE..&TIME.

For an example of the template statements in the INGTOPOF file, refer to "Sample INGTOPOF File" on page 224.

As the DisplayResourceName can now be customized, it is possible to create different resources with the same DisplayResourceName. Although duplicate DisplayResourceNames cause no problems to the NMC or RODM, it will be the responsibility of each installation to ensure that any duplication is correctly processed by any user-written code.

BLDVIEW creates views containing resources, and can identify resources for inclusion by the MyName field or the DisplayResourceName field of the resource.
- No further change to your BLDVIEW statements will be required.
- The format of the MyName field may NOT be modified.
- The format of the MyName field is, PLEX.SUBSYSTEM/TYPE/SYSTEM.MINOR
- The MyName field may have parts omitted that are not relevant.
- The following are examples of the MyName:

  - PLEX.SUBSYSTEM/TYPE
  - PLEX.SUBSYSTEM/TYPE/SYSTEM
  - PLEX.SUBSYSTEM/TYPE.MINOR
  - PLEX.SUBSYSTEM/TYPE/SYSTEM.MINOR

- If you currently use the DisplayResourceName in your BLDVIEW statements and you are customizing the DisplayResourceName, it will be necessary to review your BLDVIEW statements to ensure that the correct resources are included in your views.

The RUNOPID Statement

When submitting commands via the NMC, the commands are run under the user ID of the operator signed on to the NMC at that time.

It is possible to select a predefined user ID by using the RUNOPID statement in the INGTOPOF file. When a command is submitted via the NMC for a non-local resource, the command will be run under the predefined user ID, and not the user ID of the operator signed on to the NMC at that time.

Commands that are issued via the NMC against a local resource are never preceded by a label.

Commands that are issued via the NMC against a non-local resource are preceded by a label. This label has three separate fields:
- Netid
- Domain
- User id

Examples of the label are as follows:
- Netid:
- Netid.Domain:
- Netid.Domain/User id:

To provide an amount of flexibility, the RUNOPID statement has been introduced to the INGTOPOF file. This will allow a predefined user ID to be used in the label, rather than the user ID of the operator signed on to the NMC at that time.
If the RUNOPID statement exists in the INGTOPOF file, the associated user ID will be substituted in the label.

The syntax of the RUNOPID statement in the INGTOPOF file is

```
RUNOPID user id
```

An example of the RUNOPID statement in the INGTOPOF file is

```
RUNOPID ACDMON
```

If multiple RUNOPID statements appear in the INGTOPOF file, only the first RUNOPID statement will be used, all subsequent RUNOPID statements will be discarded.

### The HBDELETE Statement

The HBDELETE statement specifies whether or not old heartbeat entries should be deleted. The default is yes, which provides behavior consistent with earlier releases. The syntax is:

```
>> HBDELETE Y N
```

When Y is specified, all previous heartbeat minor resources from the same sysplex are deleted when any heartbeat minor resource from the sysplex is updated. This incurs a measurable resource consumption.

When N is specified, only the update to the heartbeat minor resource is made. This means that RODM may end up containing old (stopped or failed) heartbeats from other systems in the sysplex, long after the heartbeat has been picked up by another system in the sysplex. This is measurably more efficient than the Y option.

### The LINKTOVIEWS Statement

The LINKTOVIEWS statement determines which RODM fields will be used to connect major and minor resources in RODM. Specifying BASE or NONE makes processing faster, but at the cost of losing some NMC functionality. The syntax is:

```
>> LINKTOVIEWS resource linkage
```

The resource parameter may be either a qualified major resource name (sysplex.major), a sysplex name (sysplex) or the constant 'DEFAULT'.

The linkage values are:

- **FULL** All links are made, this is the default behavior. Fields linked are:
  - IsPartOf/ComposedOfLogical
  - ContainedInView
  - Aggregationparent
  - ExceptionViewList

- **BASE** The only fields linked are IsPartOf/ComposedOfLogical and AggregationParent. The missing fields mean the minor resource will not appear in any views containing the major resource and will not appear in any exception views containing the major resource (unless placed there by an alternate mechanism such as RCM or BLDVIEWS).
The MAPCOLOR Statement

The color for a resource icon of status “Unavailable” can be changed with the keyword “MAPCOLOR”. The updated color will be displayed on all NMC topology clients. The syntax is:

```
MAPCOLOR UNAVAILABLE user positive value user negative value
```

It is possible to map the status of “Unavailable” to all “User positive” and “User negative” values. These are:
- User positive: 136 137 138 139 140 141 142 143
- User negative: 152 153 154 155 156 157 158 159

Example:

The default dark green color can be changed to light green by placing the following line in the topology file (INGTOPOF):

```
MAPCOLOR UNAVAILABLE 136
```

On the NMC topology client, the color of each “User positive” or “User negative” value can be displayed and changed with:

Options → Console properties... → Status

Technical Note:

Refer to the RODM DisplayStatus field in Tivoli NetView for z/OS Data Model Reference.

Note:

The DisplayStatus field has a major impact on the decision whether an object should be placed in an exception view.

SA z/OS expects that the RODM and GMFHS defaults put the ‘UserNegative’ values into exception views. ‘UserPositive’ values are assumed not to appear in exception views.

Sample INGTOPOF File

```
*********************************************************************
* INGTOPOF sample
* The sysplex name in this example is: K1
* The sysplex consists of the following four domains: IPSNM, IPSNN, IPSNO and IPSNP
* The KEY1VIEW and CMNVIEW members contain BLDVIEWS control cards.
* They are necessary for the SA topology manager to create 'views'
* in RODM to display SA resources.
* For more details refer to the SA User's Guide,
* Using the NetView Management Console for SA z/OS,
*********************************************************************
```
* Creating Views

* This sample also contains a user defined anchor 'USER' and
* shows the usage of the 'HBDELETE', 'LINKTOVIEWS', 'OPTION' and
* 'TEMPLATE' statements.
* For a description of all keywords please refer to the
* System Automation for z/OS Planning and Installation guide.
* Use a trailing '+' for continuation.

*********************************************************************

SYSPLEX K1 IPSNM IPSNN +
  IPSNO +
  IPSNP

BLDVIEWS K1 KEYIVIEW CMNVIEW

ANCHOR K1 USER

* HBDELETE N
  When heartbeat minor resources for the SYSPLEX are updated via the INGPOST
  command, heartbeat minor resources will be created on receipt of the initial
  INGPOST command, these heartbeat minor resources will then be updated for
  subsequent INGPOSTs commands.

* HBDELETE Y
  When heartbeat minor resources for the SYSPLEX are updated via the INGPOST
  command, any existing heartbeat minor resources for the SYSPLEX will be deleted
  and new heartbeat minor resources for the SYSPLEX will be created.

* In the following LINKTOVIEWS examples,
  o The sysplex is 'K1',
  o The major resource is 'KEY1/SYS/KEY1'

* LINKTOVIEWS DEFAULT FULL
  LINKTOVIEWS K1 BASE
  LINKTOVIEWS K1.KEY1/SYS/KEY1 NONE

* OPTION NOPARENTS
* OPTION NOLINKMINOR
OPTION MDL_RADLINK

* To define how the DisplayResourceName is formatted,
  substitution parameters are employed. Substitution
  parameters may appear in any order. The following
  substitution parameters are supported,

* &STR. - SYS.TYPE.SUB
  &RES. - SUBTYPE/SYS
  &MNR. - MINOR RESOURCE NAME (Minor Resources only)
  &SUB. - SUBSYSTEM
  &TYP. - TYPE
  &SYS. - SYSTEM (NULL FOR SYSPLEX RESOURCE)
  &EVT. - EVENT
  &PLX. - SYSPLEX
  &DATE. - DATE
  &TIME. - TIME

* To activate a TEMPLATE statement remove the leading asterisk from
  the following samples.

*TEMPLATE DRN &PLX..&STR. &EVT.
*TEMPLATE DRNM &PLX..&STR..&MNR. &EVT.
Syntax for INGTOPOF File

* 
  *TEMPLATE APL &SYS..&SUB. 
  *TEMPLATE APLM &MNR. 
  * 
  *TEMPLATE APG &PLX..&SYS. &RES. 
  *TEMPLATE APGM &PLX..&SYS. &RES. &MNR. 
  * 
  *TEMPLATE APGP &PLX..&RES. 
  *TEMPLATE APGPM &PLX..&RES. &MNR. 
  * 
  *TEMPLATE MTR &SYS..&SUB. 
  *TEMPLATE MTRM &MNR. 
  * 
  *TEMPLATE SYS &PLX..&RES. 
  *TEMPLATE SYSM &PLX..&RES. &MNR. 
  * 
  *TEMPLATE SYG &PLX..&RES. 
  *TEMPLATE SYGM &PLX..&RES. &MNR. 
  * 
  *TEMPLATE GRP &RES. GRP 
  *TEMPLATE GRPM &RES..&MNR. GRPM 
  * 
  *TEMPLATE HEARTBEATM &PLX..&RES. &MNR. &EVT. &DATE..&TIME. 
  * 
  *TEMPLATE WTORM &MNR. &EVT. 
  *TEMPLATE TAPEM &MNR. &EVT. 
  * 
  *TEMPLATE CFM &PLX..&RES. &MNR. &EVT. 
  *TEMPLATE CDSM &RES. &MNR. &EVT. 
  *TEMPLATE ETRM &MNR. &EVT. 
  *TEMPLATE SYSPLEXM &PLX..&RES..&MNR. &EVT.* 
  *TEMPLATE USER &STR. &DATE. &TIME 
  *TEMPLATE USERM &MNR. &PLX..&SUB. &DATE. &TIME. &EVT. 

*******************************************************************************
Appendix C. Miscellaneous Information

This section tells you how to do the additional installation tasks involved in using the enterprise monitoring functions of SA z/OS.

Running Two NetViews on the NMC Focal Point System

If your focal point system runs one NetView for automation (Automation NetView) and another NetView for networking (Networking NetView) that includes an NMC focal point system, you must install SA z/OS on both NetViews. The SA z/OS installation on the NetView used for networking involves only a subset of SA z/OS code, called an SA z/OS satellite, and fewer installation steps are required.

Where the Networking NetView is an enterprise monitoring focal point, the SA z/OS NetView’s DS16INIT Parm should specify the Networking NetView on the same system as its focal point. The focal point needs to receive heartbeats from the SA z/OS domain on the same system to set the necessary RODM focal point fields.

Installation of an SA z/OS satellite is covered as an optional step. See “Step 28: Install an SA z/OS Satellite” on page 138.

Users and RODM Authorization

When RODM is installed on your system, it is necessary to authorize users and applications to access RODM services. This authorization is accomplished using RACF or an equivalent security application. See Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer’s Guide for details about specifying RODM authorization. This section describes any additional user IDs that must be created for system operations enterprise monitoring and indicates whether they require RODM authorization.

Table 33. RODM Authorization for user IDs

<table>
<thead>
<tr>
<th>User ID</th>
<th>RODM Authorization Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetView Graphic Monitor Facility operators</td>
<td>No</td>
</tr>
<tr>
<td>SA z/OS operators</td>
<td>Yes</td>
</tr>
<tr>
<td>User ID for bulk updates from NetView (specified in AOFRODM)</td>
<td>Yes</td>
</tr>
<tr>
<td>User ID for GMFHS to connect to RODM (defined when you install GMFHS and RODM)</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Graphic Monitor Facility Host Subsystem (GMFHS) operator IDs are usually created to be the same as NetView operator IDs so that operators can use the same ID and password to log on to GMFHS as they use to log on to NetView. RODM authorization is not required for use of GMFHS, but the IDs may require authorization for other purposes such as using RODMVIEW.

Note: If you assign an GMFHS operator ID of OPER1 on the NMC focal point system, GMFHS automatically uses the same GMFHS operator ID on other NetViews in the enterprise as the target for commands.
In addition to logging on to GMFHS, operators using system operations enterprise monitoring need to log on to SA z/OS. You may choose to use the same set of IDs for SA z/OS as you do for NetView and GMFHS. However, SA z/OS IDs must be authorized to RODM. Because an ID can only be used to connect to RODM from one application at a time, you should create a unique system operations ID for each operator who connects to RODM from another application.

**Verifying Installation of SA z/OS Satellite (Optional)**

You should now test your Networking NetView (with added system operations satellite). An outline procedure for this is:

1. Schedule a testing period. You will require your focal point system and expertise on how the Networking NetView should behave.
2. Shut down your Networking NetView. This means you no longer have any network automation.
3. Start your Networking NetView with the SA z/OS satellite.
4. Check that it initializes without error.
5. Check that your Networking NetView still works.
6. Start the NetView with the satellite installed and the SA z/OS topology manager configured. At this point, the SA z/OS topology manager should automatically contact all defined target sysplexes, retrieve their configuration information and create corresponding objects in RODM. Finally it will run the BLDVIEWS statements that you have defined for each sysplex. These will create views in RODM allowing you to see the objects created by the SA z/OS topology manager.
7. Start an NMC server connected to the focal point system and then connect to it from an NMC client. You should see the views defined by your BLDVIEW statements. These should contain objects representing the automated resources on the target sysplexes. There should be a green heartbeat icon for each active target sysplex.
8. If you select an icon representing an automated resource and right-click, you should see SA z/OS commands on its context menu. Select INGINFO and see that the command is issued properly.
9. Shut down the new Networking NetView, bring up the former one, and plan for production cutover.

**Enabling SA z/OS Support for Extended Multiple Console Support**

This section describes how to set up extended multiple console support (EMCS) and also describes its restrictions and limitations.

Note: EMCS support is mandatory for the successful operation of SA z/OS.

**Setting Up EMCS**

- Add the AOCGETCN command to the initial CLIST of your operator profiles.
- Switch on the SA z/OS global variable AOF_EMCS_AUTOTASK_ASSIGNMENT, to assign an autotask to EMCS consoles.

**EMCS Restrictions and Limitations**

- There must be only one NetView running SA z/OS in each machine.
- Do not:
Enabling SA z/OS Support for Extended Multiple Console Support

- Use route codes to route messages to any NetView task console
- Deactivate the action message retention facility (AMRF) (by coding COM='K M,AMRF=N' in the COMMNDxx member of SYS1.PARMLIB)
- Change the MSCOPE setting on the xxxCSSIR task/console
- Define the AUTO attribute for any NetView task/console under the RACF OPERPARMS
- Define an SAF OPERPARM definition for extended MCS console authority to anything other than MASTER

Violation of these restrictions will cause unpredictable results.
Appendix D. Using the Hardware Integrated Console of System z for External Automation with SA z/OS

The Hardware Integrated Console provides a message and command interface for operating system images running on System z hardware to cover system initialization, recovery situations, or emergency operator tasks.

Especially when channel-attached or otherwise-connected 3270/ASCII operator console devices are not configured or cannot be used with the System z processor hardware, the integrated console is the only console interface for an operating system at initialization time.

For the SA z/OS processor hardware interfaces, the integrated console is the exclusive facility to communicate with the target operating systems running on System z processors. Other console interfaces that become available after target OS initialization is complete are not used. With the SA z/OS hardware interfaces, you can control and automate System z processors externally. This means the controlling SA z/OS program can run on a different processor or LPAR than the target system to be controlled. One typical example is to monitor or automate the IPL prompts of a remote system displayed on its integrated console.

This appendix provides background, usage, and performance information important to know if you plan to use the hardware integrated console support (CI) of the SA z/OS processor hardware interfaces for your automation. The System z hardware commands, like SYSRESET, LOAD for example, are not discussed in this chapter. For more information about automating these commands, refer to IBM Tivoli System Automation for z/OS Operator’s Commands and IBM Tivoli System Automation for z/OS User’s Guide. However the automation interface and remote configuration information in this chapter is valid for both hardware commands and CI automation. This appendix includes the following sections:

- “CI Usage in IBM Tivoli System Automation Products” on page 232
- “CI Protocols and Automation Interfaces” on page 232
- “CI Configuration for Remote Automation” on page 233
- “CI Automation Basics” on page 235
- “CI Differences to 3270-Based Console Devices” on page 236
- “CI Performance Factors” on page 236
- “Network Dependencies” on page 237
- “IP Stack Considerations” on page 237
- “ProcOps SNMP Sessions” on page 237
- “OS Message Format Support with ProcOps/BCPii” on page 237
- “Automating Multi-Line z/OS Messages” on page 238
- “Limiting the Number of z/OS IPL Messages Displayed on CI” on page 238
- “Recommended z/OS Console Settings for CI Usage with SA z/OS” on page 238
- “Using CI in a z/OS Sysplex Environment” on page 239
- “Running with the z/OS System Console Deactivated” on page 239
- “z/OS Health Checker Considerations” on page 239
- “CI Security with SA z/OS” on page 240
**CI Usage in IBM Tivoli System Automation Products**

**SA z/OS Processor Operations (ProcOps)**

Processor operations is a NetView and SNMP-TCP/IP protocol-based automation interface and API to monitor and control System z mainframes. ProcOps is a focal point application that allows external mainframe automation. See the ProcOps API command ISQSEND in *IBM Tivoli System Automation for z/OS Operator’s Commands* as an example of a ProcOps command using CI. The integrated IPL automation for z/OS and z/VM are other examples of using CI. With ProcOps, CI messages are sent automatically to the focal point system as soon as the network connection is established to the Support Element (SE) or Hardware Management Console (HMC) and the targeted system (LPAR) is registered. The ProcOps API command ISQXIII is used to perform these steps.

**System Automation for Integrated Operations Management**

System Automation for Integrated Operations Management (SA IOM) is a client server product for the Windows platform that provides SNMP-TCP/IP protocol-based REXX automation sample scripts to monitor and control System z mainframes, including the monitoring of CI messages from the HMC. Refer to the *System Automation for Integrated Operations Management User’s Guide* for more information.

With SA IOM, the SNMP Agent of the HMC that is to be used for the System z hardware access must be customized to send operating system message event SNMP traps to the IP address of the SA IOM server. This ensures that the CI messages are available for the automation scripts running on the SA IOM server.

**Related Information**

The IBM Service Offering GDPS (Geographically Dispersed Parallel Sysplex, an IBM disaster recovery solution for System z mainframes), requires NetView and SA z/OS to be active. It uses the CI facility with the internal services of SA z/OS. Refer to the *GDPS/PPRC Installation and Customization Guide* for more information. An example of CI exploitation of GDPS is DUPLICATE VOLSER automation at IPL time.

Depending on the function performed, CI message registration or deregistration is controlled internally by the GDPS code.

**CI Protocols and Automation Interfaces**

In order to use the hardware integrated console (CI), the SA z/OS program uses two communication protocols. These protocols use the System z application programming interfaces. You use Option 10 (Processors) on the Entry Type Selection panel of the SA z/OS customization dialog to configure the communication protocols for a processor. See *IBM Tivoli System Automation for z/OS Defining Automation Policy* for more information.

**INTERNAL (BCPii Base Control Program Internal Interface)**

This protocol is based on a System z internal communication service (SCLP) between the LPARs and the processor support element (SE) to perform hardware
Using the Hardware Integrated Console

operations and configuration management tasks. No network IP stack is needed. See “Planning the Hardware Interfaces” on page 18 for more information. The scope of processors that can be controlled with this protocol is the Hardware LAN.

SNMP

This protocol requires a Internet Protocol network stack. From a ProcOps focal point system, which must be connected to a business LAN, you can monitor and control processors and operating system messages (CI) from LPARs running on the controlled processors. Network access from the business LAN to the hardware LANs of the processors is required. ProcOps supports SNMP connections to HMCs and SEs.

System z Application Programming Interface

The API covers all network-specific programming services (Bind, Connect, and so on) and allows applications to concentrate on hardware function and event control. The API uses the SNMP MIB data format. Applications using the API can dynamically register for events, such as operating system messages, from the CI of a particular LPAR.

For detailed information, refer to System z Application Programming Interfaces, which is available under your HMC’s Books View or on IBM Resource Link® for download. The document also contains information about how to download the API itself for various OS platforms and Java™. This generally available API version supports the TCP/IP SNMP protocol.

A special version of the API is distributed with the SA z/OS that supports the BCPii and the TCP/IP protocol. This version can only be used together with SA z/OS.

Related Information

With z/OS V1R11, BCPii can also be used independently of SA z/OS or GDPS by applications that are written in high-level languages to automate CI operations. See z/OS MV5 Programming: Callable Services for High-Level Languages for more information. For this BCPii implementation, a special version of the System z API code is provided with the services.

Regardless of the System z APIs, you can write an SNMP manager application to process operating system message (CI) SNMP traps from an HMC or a SE. However, without using the API, you must register your application permanently with the SNMP agent to receive the SNMP trap data. You must perform this SE/HMC customization step manually.

The System z HMC can be configured to act as a Common Information Model (CIM) server. CIM client applications can be written to receive CI messages using the IBMZ_OSMessage CIM class. See System z Common Information Model (CIM) Management Interface, which is available on your HMC, if you need more information.

CI Configuration for Remote Automation

Figure 19 on page 234 illustrates how the CI of three systems is connected to an SA z/OS system, which is acting as a remote automation focal point.
SA IOM and SA z/OS ProcOps use the TCP/IP connections that are always from a focal point (SA IOM Server, ProcOps FP System) to target processors and systems. The SA z/OS BCPii (INTERNAL) is a peer connection protocol. In a system cluster like a z/OS Parallel Sysplex, all participating systems can be configured in the SA z/OS policy to have BCPii connections with one another. Focal points can be located close to the systems they control or located remotely from them. For the TCP/IP SNMP protocol that is used by SA z/OS this can be a Business LAN or Intranet, or a global Internet distance. For the BCPii (INTERNAL) protocol the distance between two BCPii connected systems depends on the dimension of the Hardware LAN.

With GDPS in a Parallel Sysplex environment, the distances between BCPii-connected systems is also affected by the connectivity requirements of the Coupling Links. Refer to the IBM Redbooks® publication, *System z Connectivity Handbook* and the available GDPS documentation for more information.

How the Hardware LAN is connected to the Business LAN depends on the security policies that apply. Router/Bridge hardware and firewall software are
Using the Hardware Integrated Console

typically used to control access. For more information refer to the Installation Manual for Physical Planning and System Overview manuals that are available for your System z mainframe.

The Hardware Management Console Application (HWMCA) is a licensed software application that is installed on the Hardware Management Console and the Support Element (SE). It provides the GUI and the interfaces for automation software. BCPii connections and TCP/IP SNMP connections use the HWMCA.

SA z/OS ProcOps runs as a NetView application and uses a Communications Server TCP/IP stack to communicate with an SE or HMC. In Figure 19 on page 234, the HMC is attached to the Hardware LAN of the mainframes, however configurations with HMCs that are attached to the Business customer LAN are also supported. Support Elements must be attached to an HareWare LAN. CI message events and commands are exchanged between the connection end points of the SE or HMC and the SA z/OS ProcOps application.

The SA IOM server workstation is attached to the Business LAN. CI message events and commands are exchanged between the connection end points of the HMC and the SA IOM application on the server. The HMC receives the CI message events from all CPCs and images (LPARs) that have been defined for it.

GDPS, which runs as a NetView application, uses SA z/OS internal services to communicate with the Support Elements over the BCPii. The BCPii protocol itself uses the z/OS support processor interface services (SCLP) to do this. If a GDPS BCPii request targets an SE other than the local one, the HMC is used to route the request to the target.

In Figure 19 on page 234, the CI of three target systems is shown. One CPC has two logical partitions, LPAR1 and LPAR2, each with a CI. The third CI is shown for a single system that is running on another CPC. Together with the CPC of the focal point system, all the CPCs are connected to the same Hardware LAN.

Although not shown in Figure 19 on page 234, a fourth CI, that of the focal point system itself, can also be automated. Both of the TCP/IP SNMP and BCPii protocols can be used to do this.

CI Automation Basics

The CI facility uses a physical (cable) connection between the processor hardware (CPC) and the attached processor support element (SE) unit. With the CI, the message and command information is exchanged between a system image running on the CPC and its SE.

For automated operations, CI has an interface to the console application (HWMCA), running on each SE or HMC. If there is a ProcOps session to a HMC/SE, or a GDPS session to a SE, the console application generates an event for each new CI message. This event is sent to all registered applications (ProcOps, GDPS), using the transport protocol configured in SA z/OS. This is SNMP for ProcOps or INTERNAL (BCPii) for GDPS.

Automation applications can send operating system console commands to a CI for execution. With SA z/OS this can happen either in response to messages that are received only over CI, or independent of that at any time. The only requirement is that a SA z/OS hardware session exists between the SE/HMC and the automation application (SA z/OS/ProcOps or GDPS). The advantage for automation of using
Using the Hardware Integrated Console

the CI is that there is no 3270-specific information and screen formatting burden. This makes the interface robust and easier to use for automation purposes than 3270 console screen emulation and interpretation.

Related Information

The Support Element (SE) provides the GUI for local CPC operation. It is connected to a processor hardware LAN, together with SEs from other CPCs that may use this HWLAN. As the next higher systems management level, Hardware Management Consoles (HMCs) can be connected to the processor hardware LAN. Within a hardware LAN, an HMC represents a single point of control for the CPC objects defined to it. HMC users can log in directly at the console, or they can use its Web interfaces to log in. In a hardware LAN environment, multiple HMCs can coexist, either sharing or splitting the control of the CPCs attached to it.

With an HMC, the normal manual CI operation is done by using the Operating System Messages task. One or multiple image objects (LPARs) can be selected, which can be located on different CPC objects. Each selected LPAR allows the use of its integrated console by clicking the desktop message window tab of this LPAR. This allows the operator to view the individual message streams and to send commands to the operating system running in this LPAR. For more information refer to the Hardware Management Operations Guide of your processor.

Manual CI operation of the SE is possible, by either accessing the SE unit located in the CPC cage, or by using the Single Object Operation Task from an HMC to control the SE remotely. These methods however are not considered to be for normal operations. They are used for CPC/SE configuration tasks or for service. For more information refer to the Support Element Operations Guide of your processor.

CI Differences to 3270-Based Console Devices

Compared to 3270 display devices, CI does not provide 3270 data stream related features such as extended color or program function key support. In case of a SE outage, the CI for all CPC LPARs is affected. The CI becomes available again, once an alternate SE is activated as the primary, or the primary SE is reactivated. In a channel-attached 3270 operator console environment, failing consoles can be backed up by using multiple operator consoles over different channel paths.

CI Performance Factors

The CPC’s microcode must handle the CI message requests from all its LPARs concurrently. Depending on the number of LPARs and the number of messages that are sent by each operating system over CI, upcoming workload peaks can influence the overall CI performance. This also applies to a SE/HMC, when a varying number of applications have to be serviced, by sending a varying number of CI message events. On the SE side, CI is lower in priority than time-critical SE tasks such as power and thermal management, and when the SE is busy with those tasks, CI can be slowed down. the activation of an LPAR can affect the CI performance of adjacent LPARs on the same CPC. See also “Testing CI Performance for SNMP Connections” on page 240.
Network Dependencies

CI-based automation with ProcOps depends on the availability of a Internet Protocol network infrastructure. The connection between the SE/HMC and a SA z/OS ProcOps FP system requires this. If a network element, such as the IP stack on the ProcOps FP system, is not available, CI-based automation cannot work. This also applies if LAN routers or bridges that are used to interconnect the CPC Hardware LAN with the customer Business LAN have configuration or connection problems, or fail.

For CI over BCPii connections, the following dependencies apply:

As long as all participating system images are running on the same CPC, no external network elements are involved. For SA z/OS managed systems, located on different CPCs of a CPC Hardware LAN, at least one HMC is involved as network element for internal routing purposes. The routing HMC and the routing mechanism are transparent to the BCPii protocol. If multiple HMCs in a CPC HW LAN are configured for routing, each of them can potentially be used for that purpose.

IP Stack Considerations

The SA z/OS ProcOps SNMP (TCP/IP) transport requires an IP stack to be active on the ProcOps FP system. The BCPii transport does not have this requirement. SA z/OS ProcOps supports multiple IP stacks on the FP system on a SE/HMC connection level. You can therefore predefine the IP stack to be used for a specific SE/HMC connection with the SA z/OS customization dialog. If you do not define an IP stack name, the system default stack is used.

Adjusting the Receive Buffer size of the ProcOps FP IP stack is an efficient way to prevent CI events from getting lost. See “ProcOps SNMP Sessions” for more details about lost events. SA z/OS ProcOps uses the Receive Buffer size value that is specified in the configuration file of the IP stack. With a larger Receive Buffer size, more CI event data can be queued to the ProcOps FP system IP stack before a Receive Buffer full condition occurs and a negative response must be returned to the SE/HMC.

ProcOps SNMP Sessions

When an SNMP (TCP/IP) connection is established to a SE/HMC, ProcOps uses the session parameter: HWMCA_TOLERATE_LOST_EVENTS. This setting makes sure that a session is not terminated by the console application (HWMCA) if the IP stack of the SE/HMC can no longer send events (CI or others) due to a negative send response returned from the ProcOps FP IP stack. In this case the event is discarded, but the session remains operational. Without this parameter, the session would terminate, the events would be discarded, and the session would have to be restarted. For more information about the session parameters refer to System z Application Programming Interfaces.

OS Message Format Support with ProcOps/BCPii

With SA z/OS, the CI message ID and message text are the only supported parts of an OS message in ProcOps/BCPii. Available CI attributes, like date and time or system names which can prefix a message line, are not supported. They may however be present in the CI window of the HMC. Similarly, display attributes, such as held message, priority message, prompt indicators or audible alarm
Using the Hardware Integrated Console

indicators, are ignored when the OS message event data is collected by SA z/OS. The unsupported CI message attributes; date, time, system name and unsupported display attributes; held message, priority message, and the audible alarm may be OS-specific. The common CI format of the operating system environments identified by the SA z/OS hardware interfaces apply to: z/OS, z/VM, z/VSE®, z/TPF, Linux on System z, Coupling Facility Control Code (CFCC), and stand-alone utilities such as SADUMP or the Device Support Facility ICKDSF.

Automating Multi-Line z/OS Messages

Care must be taken when automating z/OS multi-line messages, displayed on CI. Internal z/OS message attributes which identify the different parts of a multi-line message are not available with CI; it can be difficult to identify them explicitly. Parts of a multi-line message are: Header line, one or more Data lines, and End of message line. With the internal message data format of a multi-line message, available over the z/OS subsystem interface (SSI), you can explicitly access these multi-line message parts. ProcOps/BCPii connections to the HMC/SE are always external connections which cannot register to the z/OS SSI. With ProcOps/BCPii CI multi-line messages are only made available as a number of single message lines in the order that they are displayed on CI.

Limiting the Number of z/OS IPL Messages Displayed on CI

As part of the z/OS Load parameter specification, the initialization message suppression indicator (IMSI) can be chosen to control the suppression of messages and system prompts during initialization. The IMSI character tells the system whether to perform the following actions during system initialization:

- Display most informational messages
- Prompt for system parameters
- Prompt for the name of the master catalog

See the section “Loading the System Software” in z/OS MVS System Commands for a table that shows the possible values for the IMSI character. The values indicate all possible combinations of the actions that are listed.

Whenever possible, it is recommended that you suppress the display of informational messages to reduce the total number of messages at IPL time. If you plan for z/OS IPL automation do not use informational messages as automation action triggers. Choose only messages that cannot be suppressed, in addition to action or decision operator prompts.

Recommended z/OS Console Settings for CI Usage with SA z/OS

Although not a 3270 console device, z/OS supports certain console characteristics for this facility. In the z/OS literature it is referred to as a system console. Because the system console is a special facility, z/OS allows you to activate and to deactivate its usage. This is done with the z/OS console commands V CN(*),ACTIVATE and V CN(*),DEACTIVATE, entered at the HMC or by automation software.

Once activated, z/OS calls this ‘the console is in Problem Determination mode’. Operators or automation software can use it to get command responses and unsolicited messages. The amount of unsolicited messages sent to the z/OS system console (CI) can be controlled by setting its z/OS routing codes.
Using the Hardware Integrated Console

You can specify the AUTOACT group keyword in the CONSOLxx member of the PARMLIB. With an AUTOACT group, the ACTIVATE, DEACTIVATE of the system console can be done automatically.

If you have automation routines to issue commands on the CI after IPL is complete, make sure that the allowed routing codes for the system console are limited. Issue command V CN(*),ROUT=NONE on the CI to achieve this. This setting makes sure that you receive only the command responses, job start/stop information, and z/OS priority messages. For more information about system console (CI) and AUTOACT usage refer to z/OS MVS Planning: Operations.

Using CI in a z/OS Sysplex Environment

In a sysplex environment you can set the message scope for the system console to cover multiple or all systems of the sysplex. Do not do this if you use SA z/OS ProcOps or GDPS to monitor and control the systems. The scope must be limited to the system, to which the system console is attached.

The z/OS ROUTE command allows you to forward operator commands from the System Console (CI) of one system in a Sysplex to another system in the same Sysplex for execution. The command response is then returned to the System Console where the ROUTE command was entered. In a SA z/OS environment, do not use the ROUTE command in your CI communication-based automation. Instead, you should establish a connection to the CI of each system and address each target system directly.

The reason for this restriction is the fact that the SA z/OS Hardware interface automatically prefixes CI messages with the processor (dot) LPAR name of the CI, where the message is displayed. For a ROUTE command response, however, this may not be the system location where the response came from.

Running with the z/OS System Console Deactivated

In deactivated mode, the z/OS System Console (CI) does not allow you to issue regular operator commands. Unsolicited z/OS messages are not displayed, with exception of z/OS priority messages. In addition you can:

• Send a message to the System Console from TSO or another z/OS consoles (MCS/SMCS/EMCS), using the system console’s z/OS console name as destination,
• Respond to pending system requests (reply numbers). Care must be taken when doing this because no response messages are displayed. In deactivated mode you can also not issue a z/OS D R command to determine the pending requests.

z/OS Health Checker Considerations

The Health Checker MVS component allows monitoring of certain active settings for the System Console (CI) and to issue exception notification messages if they deviate from predefined best practices settings. Together with many other checks of the system environment the z/OS Health Checker can help to recognize potential system problems or even to prevent system outages.

If you have z/OS system images controlled remotely with the SA z/OS hardware interfaces and you have their System Consoles (CI) running in PD mode, you have
Using the Hardware Integrated Console

to decide if this is really considered to be an exception in case the IBMCNZ Syscons checking is active. For more information about Health Checking refer to *IBM Health Checker for z/OS: User’s Guide*.

CI Security with SA z/OS

You can control the usage of CI with SA z/OS by restricting the user access to the processors hardware and LPARs. SA z/OS users without the required permission are not able to issue Hardware interface commands either directly with ProcOps or indirectly using a GDPS command which issues hardware interface commands internally. For more information see “Controlling Access to the Processor Hardware Functions” on page 184.

Note: Regardless of restricting the CI access with SA z/OS, some operating systems that use CI as a console facility restrict console usage by requesting an operator to log in first. If you perform such a login with SA z/OS, for example using the ProcOps ISQSEND API command, password information is not protected.

Testing CI Performance for SNMP Connections

Sending a specified number of predefined (pattern) messages to the integrated console using a message per second rate of your choice is the basic logic to determine the overall CI message throughput and performance of a SA z/OS SNMP connection to a SE or HMC.

Once the messages arrive at the ProcOps FP system, they are written to the NetView log. You can determine if OS message events are lost by controlling the message sequence numbers.

In the example shown in **Figure 20** the ISQ999I message sequence number is 00004. The test case was started for a total of 00010 messages. In the ProcOps FP Netlog you should find all messages from 00001 to 00010. If one or more messages are missing, this indicates that message events were lost on the connection.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ISQ999I 12:24:01 Test Message 0004 of 00010 *** 1234567890$%/ (</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>)=? qwertzuiop_QWERTZUIOP* <em>ProcOps-SYSCONS</em> asdfgh+120</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 20. ISQ999I Test Message Pattern Example**

Two REXX program utilities, ISQWTO3 and ISQTSND3 are delivered with the SA z/OS sample library SINGSAMP as members INGEI005 and INGEI006.

Both programs require specifying the total number of messages to be produced on the integrated console (CI) per call. The second parameter can be used to specify the message per second rate that the utility should try to achieve. For installation and usage information refer to the utility source members in the SINGSAMP library.
Using the Hardware Integrated Console

ISQWTO3 is the utility implementation for NetView environments; ISQTSND3 is a TSO implementation, if a NetView/SA z/OS environment is not available on the z/OS system to be tested.

Run the programs with different combinations of total message numbers and message per second rates. This allows you to emulate different CI message load situations.

Warning!: The usage of these utilities can produce many messages in the system log of the targeted system and the NetView log of the ProcOps FP system.

Summary: Managing CI Performance for SA z/OS

Bear in mind the following recommendations:
1. Follow the recommendations in this chapter to reduce the number of CI messages.
2. If possible, do not use CI alone to monitor the control a system completely. Limit its usage to system initialization and recovery situations.
3. Avoid issuing commands over the CI that may return a large amount of output.
4. For SNMP connections, consider using separate IP stacks with tailored Receive Buffer sizes to cover lost message event situations.
5. Use the ISQWTO3 and ISQTSND3 utilities from the SA z/OS sample library to test peak message load situations and how they affect CI performance.
Using the Hardware Integrated Console
Appendix E. Migration Information

This appendix provides information about migrating to SA z/OS 3.5 from SA z/OS 3.4 or SA z/OS 3.3. The actions that are required depend on which release you are migrating from.

- "Migration Steps to SA z/OS 3.5"
- "Migration Notes and Advice when Migrating to SA z/OS 3.5" on page 247
- "Migration Notes and Advice when Migrating from SA z/OS 3.3" on page 247
- "Coexistence of SA z/OS 3.5 with Previous Releases" on page 248

Migration Steps to SA z/OS 3.5

Complete the following steps to migrate to SA z/OS 3.5:

1. Install the compatibility APAR OA45711 (SA z/OS 3.3 and SA z/OS 3.4) before migrating to SA z/OS 3.5. Open the customization dialog before converting to a SA z/OS 3.5 policy database in step 2. This APAR also enables you to use a SA z/OS 3.5-built configuration file on a system running SA z/OS 3.3 or SA z/OS 3.4 in a mixed environment.

2. Make a copy of your V3.n policy database and edit it with the SA z/OS 3.5 customization dialog. This converts it to a V3.4 policy database. For more information, see "Conversion Function" in IBM Tivoli System Automation for z/OS Defining Automation Policy.

3. Read through the following sections before migrating to SA z/OS 3.5:
   - If you are migrating from SA z/OS 3.3, "Migration Notes and Advice when Migrating from SA z/OS 3.3" on page 247, and afterwards "Migration Notes and Advice when Migrating to SA z/OS 3.5"
   - If you are migrating from SA z/OS 3.4, "Migration Notes and Advice when Migrating to SA z/OS 3.5"

4. Build the configuration files from the policy database. For more information, see "Building and Distributing Configuration Files" in IBM Tivoli System Automation for z/OS Defining Automation Policy.

5. Load the build files on the designated system. For the first load of the new and converted build files a NetView recycle is required. For more information, see "Step 18B: Distribute System Operations Configuration Files" on page 122 and the chapter "Building and Distributing Configuration Files" in IBM Tivoli System Automation for z/OS Defining Automation Policy.

Migration Notes and Advice when Migrating to SA z/OS 3.5

This section contains details of various aspects of migration that you should be aware of. Make sure that you read through this section before migrating to SA z/OS 3.5:

- Post SMP/E Steps
- Miscellaneous

Post SMP/E Steps

You must review the following standard installation steps and, if necessary, carry them out:

1. "Step 4A: Update IEAAPFx" on page 82
2. “Step 4B: Update SCHEDxx” on page 82
3. “Step 4D: Update LPALSTxx” on page 83
4. “Step 4E: Update LNKLSTxx” on page 84
5. “Step 5: Configure SYS1.PROCLIB Members” on page 86
6. “Step 6F: Add the REXX Function Packages to DSIRXPRM” on page 95
7. “Step 10B: Configuring HSAPRMMxx” on page 107
8. “Step 13A: Allocate Libraries for the Dialogs” on page 111
9. “Step 17: Compile SA z/OS REXX Procedures” on page 121 (if necessary)
10. “Step 23: Check for Required IPL” on page 130
11. “Step 36: Configuring I/O Operations” on page 156 (if you need to specify communication via IP)

AT / MRT / MPF Migration Notes

- The ProcOps Automation Table ISQMSG01, which has been shipped by System Automation as a sample AT, is merged into INGMSGSA.
- REXX error messages are now run through the AT. There is the following entry in INGMSG01 to avoid all REXX SAY and TRACE output flooding the AT, but allowing REXX error messages to be trapped in the AT:
  ```plaintext
  IF HDRMTYPE='C' & MSGID /SM790000 'IRX' .THEN;
  ```
- MPF entries are generated for messages with Ignore message id characters = LEADING or BOTH. On panel Automation Table entry Conditions (AOFGMATS) there is a field Ignore message id characters. When the values LEADING or BOTH are specified then for the AT a dot is generated in front of the message id, for example IF MSGID = . 'TCA3103'. Nevertheless the message is put into the MPFLSTxx member as it is specified. So in this example, the message id TCA3103 would be put into the MPF member.
- If MRT default AUTOMATE = YES is specified then now also the messages from INGMSGSA are added to MRT and MPF. So if a customer had added these messages already himself and has selected this option, then he needs to cleanup at least his MRT because there each message id should appear only once.
- The following messages are moved from INGMSGSA to +SA_PREDEFINED_MSGS so that installation specific changes for the AT entry can be applied in the usual way via modification of the predefined data:
  - IEE303I
  - EVJ120I
  - EQQE037I
  - EQQE107I
  - EQQW079W
  - EQQZ7201I

File Update

Some specifications for Message Automation have changed, and while most downlevel specifications are accepted, for the following a manual conversion is required:

- **Ignore/Suppress selection : SUPPRESS**
  This specification needs to be replaced by the following:
  ```plaintext
  Action     -AT
  Ignore for AT :YES
  ```
Action -MRT
Ignore for MRT :YES
Action -MPF
MPF message parameter :SUP(YES),AUTO(NO)

- **MRT do not automate :YES**
  This specification needs to be replaced by the following:
  Action -MRT
  MRT automate :NP

- **Application Status : CAPTURE**
  This specification needs to be replaced by the following:
  AT Capture message selection : SELECTED

Some specifications for Entry Type Processor (PRO) for NEW and UPD have changed. Downlevel specifications are not accepted. Use File Update to create a file in the new format and update the NEW and UPD sections accordingly.

**Miscellaneous**

- Customization Dialog, Policy Activity Log. The recommended record length of the policy activity log is changed to 400. New log data sets will honor this value. It is advised to migrate existing log data sets and thereby to increase the record length from 250 to 400.

- With NetView 6.1, which is a prerequisite of SA z/OS 3.5., System Automation's REXX function will be loaded automatically. Therefore drop your modified DSIRXPRM module from your used defined load library. In addition to this, consult [Step 6F: Add the REXX Function Packages to DSIRXPRM](#) on page 95 in Chapter 10, “Traditional SA z/OS Configuration,” on page 73.

- NetView task DSIRQJOB has been added to the *BASE policy and will be controlled by SA z/OS now.
  The NetView task DSIRQJOB (part of the infrastructure for the NetView SUBMIT and ALLOCATE command) has been added to the *BASE policy. If you intend to control its availability through SA z/OS, then Import the APL DSIRQJOB APL from the *BASE best practices policy and remove the TASK.DSIRQJOB.INIT=Y statement from your CNMSTGEN definitions.

- With SA z/OS 3.5, the System Automation SETTIMER is activated by default. Remove SETTIMER activation definitions from CNMCMDU:
  CMDDEF.EZLE600A.CMDSYN=TIMER,TIMERS,TIMR,SETTIMER
  CMDDEF.AOFRAATA.CMDSYN=AOFRAATA

- Removal of SA z/OS communication task parameter member INGXKSYS
  Parameter member INGXKSYS for GDPS controlling systems has been removed. Its content was merged into the existing member INGXINIT.

- Removal of pseudo messages ACORESTART and INGTIMER.
  During the initial conversation the commands defined within the pseudo message ACORESTART will be moved to new startup phase REFRESHSTART in the policy STARTUP. The commands defined within pseudo message INGTIMER will be moved to new startup phase ANYSTART in the policy STARTUP. You may consider merging your POSTSTART and REFRESHSTART command definition into ANYSTART.

- Header layout of INGLIST, INGFILT, INGIMS command changed.
  The column name 'subtype' changed to 'subcategory' in the header line of these commands. This also implies that the start position of all subsequent columns after 'subcategory' changed. Note that it is recommended to use the INGDATA command in automation scripts instead of INGLIST.
- Layout INGINFO and DISPINFO command changed.
  The data field 'subtype' changed to 'subcategory' in the output of these commands changed.

- Layout of DISPGW command changed.
  The column 'In/Outbound' was split into the columns 'In Status' to reflect the inbound status and 'Out Status' to reflect the outbound status in the header line of the command.

- Attributes requisites for OUTDSN output data set.
  OUTDSN is a valid parameter of several SA z/OS commands. The usage of OUTDSN requires that the specified output data set exists already. The recommended record format (RECFM) is VB and recommended record length (LRECL) is 1024. It is advised to migrate existing output data sets and thereby to adapt the data set attributes appropriately.

- Status Display Facility (SDF) definitions in the NetView style sheet.
  For the definition of the systems participating in the SDF common AAO stem variable AOF_AAO_SDFROOT.List is introduced. Please convert the AAO variable AOF_AAO_SDFROOT_LIST into the stem form. Therefore consult Step 22 Configure the Status Display Facility (SDF) on page 129 in Chapter 10, "Traditional SA z/OS Configuration," on page 73. Note that AAO AOF_AAO_SDFROOT_LIST is still valid for compatibility reasons in a mixed environment but will be dropped in a future release.

- Monitor Config Refresh
  A new status descriptor INGCFG is available and the INGPTOP panel is updated under SDF for this function.

- Batch Command Interface ; INGRCRDX ; AT Checking ; Command Receiver
  With SA z/OS 3.5, sending commands with SERVER=* is done 'authorized only'. The same is true for relational data services from TSO via INGRCRDX and AT Checking by the customization dialog.
  Therefore it is necessary to install the TSO authorized command INGPAUTH. For details, see Step 15B: Install SA Provided Authorized TSO Command INGPAUTH on page 118 and the chapter "Command Receiver" in IBM Tivoli System Automation for z/OS Customizing and Programming.

- Policy Database Security
  For the data set of the Policy Database, a discrete or generic SAF profile is required. If a user has only READ access to the data set then every request to open the PDB is switched to BROWSE for the PDB.

- Output format of Policy report changed
  In case you have report post processing in place, be ready to adapt it to the new format.

- End-to-End Adapter
  If facility class BPX.DEAMON is defined in RACF you must ensure that SINGMOD1 and maybe other libraries are program-controlled. For details, see the chapter "Operating the end-to-end automation adapter" in IBM Tivoli System Automation for z/OS End-to-End Adapter.

  Authorization checking has been enhanced. If specific RACF profiles are defined, authorization checking has been tightened. For details, see the chapter "Security" in IBM Tivoli System Automation for z/OS End-to-End Adapter.

  The end-to-end adapter start script is now read-only. There is no need any more to copy and change the script. Instead copy and change the new properties file ingadapter.properties. For details, see the chapter 'Installing the end-to-end automation adapter' in IBM Tivoli System Automation for z/OS End-to-End Adapter.
Looping Address Space Suppression and ITM

The security model for SOAP servers has been enhanced so that the userid and password can be specified through the customization dialogs. If you have installed Looping Address Space Suppression under SA z/OS 3.4 then you will have defined the userid and password it uses via INGPW. The documentation for V3R5 instructs you to specify the userid in the customization dialogs along with a password of SAFPW which instructs the agent to consult INGPW for the password.

The previously defined method for V3R4 will start work, however we would recommend that you switch over to using the V3R5 method. To do this, you need to add the userid and SAFPW as the password to your SOAP server definition and then use INGPW to delete the USER SOAP entry.

With z/OS 2.1 Health Checker address space (HZSPROC) is automatically started by the system at IPL.

With z/OS 2.1 Health Checker address space is automatically started by the system under JES. This may prevent your JES address space termination at system shutdown. Consider controlling the Health Checker through SA z/OS and import the APL HSZPROC from the *IBMCOMP best practices policy.

Migration Notes and Advice when Migrating from SA z/OS 3.3

- The Inactive Monitor Health status is no longer available for the INGMON command. Clean up any PDB or Automation Table entries that have an entry INGMON STATUS=INACTIVE. Conversion will remove the INACTIVE status in all cases where INACTIVE was selected by the MESSAGES/USER DATA policy (line command AS).
- New columns are added and the order of columns is changed for the command DISPGW.
- The IMS State/Action Tables (ISA) Entry Type has been withdrawn from the Product Automation Entry Type.
- The Timeout Settings Entry Type (TMO) has been withdrawn. This entry type defined the NetView globals: WAITTIME, XDOMTIME and up to 4 user globals in the form xxxxTIME together with specific timeout values. These definitions can also be done in the NetView member CNMSTYLE.
- Message variables are now shown in mixed case where applicable. The consequence for the DISPACF command is that data is displayed as entered in the original policy.
- The INGLIST command linemode has a new layout starting with the "Health" column. It is recommended to use the INGDATA command in automation scripts.
- When running in a mixed environment, install compatibility APAR OA37376 (prerequisite is other known compatibility APARs OA34478 and OA33659).
- SDF Statuses APGG, APGY, APGB, and APGR are obsolete and replaced by APG_OK, APG_ERR, AOG_DOWN and APG_DOWN.
- Beginning with V3.4 message variables are shown in mixed case where applicable. The consequence is that the DISPACF command shows the data as specified in the policy.
- The default type of APG changed from SYSPLEX to SYSTEM. Clients who use 'update via file' to create new APGs of type SYSPLEX need to define explicitly the APG Type to SYSPLEX in the 'NEW APG' block.
- The Batch command receiver has been enhanced to enable concurrent usage of the command receiver. Commands submitted from batch jobs may be processed in parallel. For more information, refer to the chapter "Command Receivers" in
IBM Tivoli System Automation for z/OS Customizing and Programming. The REXX module EVJRYCMD was added as AOFRYCMD to the ING.SINGTREX library. For migration purposes, EVJRYCMD also resides in ING.SINGNREX library. It is recommended to use AOFRYCMD in the future and to adapt the automation environment accordingly.

- Specific messages for SDF are now added with the prefix AOFS. Some AOF-prefixed messages are now withdrawn and replaced. Refer to the "What is New in SA z/OS 3.4" section of the manual (Version 3 Release 4) for a full illustration of message changes from prefix AOF to AOFS.

- The 'Automation Name' for APGs is set to the APG's entry name by default. The automation name was undefined before. If you used the option 'Update via file' to create new APGs with a defaulted 'Automation Name', you now need to set explicitly 'Automation Name' to blank in the 'NEW APG' block. If this is not done, the APG may build a resource that is visible to operators.

- The name of the subsystem interface router task is set to the default NetView name CNMCSSIR. When using NetView 6.1 it cannot be adapted any longer. If you are not using NetView 6.1 then you still have the option to set the SSINAME in the NetView stylesheet according to your needs. There is no need to adapt the SSName in AOFMSGSY any longer.

- The TSO REXX function package INGTXFPG has been introduced. For full details, please refer to "Step 15: Configure Function Packages for TSO" on page 117.

- The default for the ALL2CONS parameter of the DISPSTAT command changed from NO to YES. The offset layout of DISPSTAT in line mode has changed for this release.

---

Coexistence of SA z/OS 3.5 with Previous Releases

It is not expected that you will cut over all your systems at the same time from previous releases to SA z/OS 3.5. This means that you may be running different releases at the same time.

SA z/OS 3.5 systems can coexist with SA z/OS 3.4 and SA z/OS 3.3 systems in the same sysplex. [Figure 21 on page 249] illustrates this: it shows a sysplex with three automated systems and a separate automation manager (and its secondary).
Any policy database created by a previous version of the customization dialog (that is, earlier than SA z/OS 3.5) is automatically converted into the SA z/OS 3.5 format when the policy database is opened the first time using the SA z/OS 3.5 customization dialog.

The automation configuration files that are built by the SA z/OS 3.5 customization dialog can be used by any automation agent running either SA z/OS 3.5, SA z/OS 3.4, or SA z/OS 3.3.
The NetView automation table (AT) that is created by the SA z/OS 3.5 customization dialog can be used by automation agents running either SA z/OS 3.4 or SA z/OS 3.3, but the automation table INGMSGSA is required for compatibility with SA z/OS 3.3.

In a sysplex (that is, the same XCF group) automation agents running SA z/OS 3.5, SA z/OS 3.4, or SA z/OS 3.3 can communicate with an SA z/OS 3.5 automation manager. The communication is via XCF. The automation agents communicate with each other via XCF.
Appendix F. Ensemble Hardware Management Console Setup

Setting up the Hardware Management Console for use with System Automation for z/OS

In order to exploit the Web Services API of the zEnterprise System Hardware Management Console (HMC), the following setup actions are required:

1. A user must be defined with the appropriate management scope and task roles to access objects and perform actions at the HMC
2. The Web Services API must be enabled in general and the user defined in step 1 must be enabled to access this interface.

These actions are described in the following subsections in more detail. For a comprehensive reference about management scope and task roles as well as for information about console actions to administrate the HMC environment, refer to the System z Hardware Management Console Operations Guide Version 2.11.11 or later as well as to zEnterprise System Hardware Management Console Operations Guide for Ensembles Version 2.11.12 or later.

Defining a user

To define a new user, login at the HMC with the pre-defined user ACSADMIN or with a user that has equivalent authorization to define a new user.

1. Select the User Profiles task.
2. Add a new user:
   a. Select the type of Authentication. For Local Authentication, a password must be specified. If you plan to allow SA z/OS to maintain the password in the VSAM file for the SAFPW user predefined value in the zEnterprise Ensemble SA z/OS customization dialogs, the password value must be 4-8 characters long. If you select LDAP Server as the means for authentication, the server managing the directory that lists this user must be selected or defined first.
   b. Select the Managed Resource Roles that determine to which objects access is permitted for this user. For system management functions such as monitoring, discovery and availability management, the assumption is the user has access to all resources in the scope of the ensemble managed by this HMC. For this, select the following pre-defined roles. If you want to limit access to certain resources only, you need to have defined corresponding roles yourself:
   c. Select the Task Roles that determine which tasks are permitted on the Managed Resources selected above. Select the following pre-defined roles or equivalent roles that you have created for this HMC:
   d. Select other User Properties. Make sure you select Allow access to management interfaces as this enables the user to use the Web Services API.

Enable Web Services API

To enable the Web Services API, login at the HMC with the pre-defined user ACSADMIN or with a user that has equivalent authorization to customize API settings.

1. Select the Customize API Settings task.
2. Select WEB Services and either enable ALL or just specific IP-addresses that are allowed to connect to this HMC.

3. Make sure the user SA z/OS uses to logon to the HMC is selected under User Access Control. Users are selected automatically, if the Allow access to management interfaces user property was set for this user (see 2d on page 251).

Getting the Hardware Management Console certificate

The communication over secure HTTP requires that all data is encrypted using a secret key. For key exchange, the HMC sends its certificate to the client who can then validate it and when trusted, the keys can be exchanged.

To allow SA z/OS to validate the certificate, its truststore must contain a copy of the public part of the server certificate or it must have a copy of the public part of the Certificate Authority’s (CA) certificate. If a server’s certificate is not found in the truststore but the certificate of the CA that signed the server’s certificate is, then the validation can still be performed.

For self-signed certificates, or for certificates that are signed by a CA that is not in SA z/OS’s truststore it is necessary to first obtain a copy of the certificate (its public part). You can do this with your browser by typing in the web address of the HMC into the address field of your browser.

If this is the first access of the HMC for the current web browser session, you can receive a certificate error. In this case, follow the instructions provided by the browser to view and export the certificate. You might have to authenticate with an administrator userid and password before the browser allows you to export the certificate. As an example, this process is outlined for the Firefox browser:

1. Point your browser to the HMC by entering the hostname or the IP-address of the HMC into the URL input field.
2. If the certificate cannot be validated, a warning popup window appears with title This Connection is Untrusted. Click on I Understand the Risks and then press the Add Exception... button.
3. The Add Security Exception dialog is displayed.
4. Press button Get Certificate. This allows the browser to get the certificate and the View... button will be enabled.
5. Press button View... to open the Certificate Viewer dialog.
6. Verify who issued the certificate and to whom it was issued. If OK, press button Details followed by Export to save the certificate on your disk.
7. The certificate is stored in text format and can now be copied to the machine where SA z/OS is running and imported into SA z/OS’s truststore.

Firewall considerations

When the Web Services API is enabled, the HMC API HTTP server listens for SSL-based socket connections on TCP port 6794. The HMC is enabled for both the SSL version 3 and TLS version 1 protocols on this SSL port. It does not accept non-SSL connections.

As part of the Web Services API, the HMC also provides an integrated JMS message broker based on Apache ActiveMQ Version 5.2.0. This message broker is active on the HMC whenever the Web Services API is enabled.
When active, the integrated broker listens for client connections using the following transports supported by ActiveMQ:

- STOMP (Streaming Text Oriented Messaging Protocol) flowing over SSL connections, listening port 61612.

The broker is enabled for the SSL version 3 and TLS version 1 protocols on these SSL ports.

The listening ports listed above for the API and for the message broker are fixed port numbers and are not subject to customer reconfiguration.

If you have firewalls between SA z/OS and the HMC, you need to contact your network administrator to set up firewall rules that enable communication over these ports across firewalls.
Appendix G. Syntax for HSAPRM00

Notes:
1. A sample member called HSAPRM00 is provided in the SINGSAMP sample library.
2. Records starting with a '*' in column 1 are treated as comments. Each parameter must be specified on a single line. Trailing comments are not supported.

ARMWAIT=nnn
BLOCKOMVS={YES|NO}
BUILDTIMEOUT={ss|180}
CFGDSN=<configuration file data set name>
COMM=CFE
DELAY={ss|0}
DIAGDUPMSG={nnnn|0}
DIAGINFO=dsname
GRPID={xx| ' '}
IOINTERVAL={n|0}
LEOPT={any}
LIFECYCLE={500|nnnn};MY.AGENT.DATA.SET
LOGSTREAM={YES|NO|GRPID}
NUMQTHDS={n|0}
OVRDELETEDELAY={dd|0}
PREF={n|0}
PREMPT={YES|NO}
START={COLD|HOT|WARM}
STOPDELAY={ss|30}
TAKEOVERFILE=name
TAKEOVERTIMEOUT={nn|12}
WLMQUERYINTERVAL={n|0}

ARMWAIT
Maximum number of seconds the automation manager waits for ARM being up during automation manager initialization. Not specified or 0 specified does not cause the AM to wait.
A value from 0-999 seconds may be specified.

BLOCKOMVS
This parameter allows you to specify whether the automation manager blocks OMVS shutdown as long as the automation manager is active.

YES If BLOCKOMVS=YES is specified, at the automation manager's initialization time, it adds a shutdown block to OMVS. Thus OMVS does not terminate as long as the automation manager is active, even if this is requested by the operator. OMVS is stopped only when the automation manager is stopped with the AM stop command.

Notes:
1. A STOP,DEFER causes the automation manager to terminate when all agents connected to it have terminated. Then the stop command for OMVS will get through.
2. For BLOCKOMVS=YES the automation manager must be UID(0).
3. For BLOCKOMVS=YES to work effectively, the stop command for OMVS must be issued as "F OMVS,SHUTDOWN".
Syntax for HSAPRM00

NO  If BLOCKOMVS=NO is specified and OMVS shuts down, the automation manager abends due to cancellation by OMVS.

Note:
1. You should not use STOP, DEFER when BLOCKOMVS=NO is specified as it will cause unpredictable results.

BUILDTIMEOUT
May be used to specify a time limit for the completion of the data structure build process that is used during a COLD or WARM start of the primary automation manager. You can specify a value from 0–180 seconds. A value of 180 (3 minutes) is assumed if omitted. A specification of 0 suppresses timing of the data structure build process.

CFGDSN
The CFGDSN value is used only on a COLD start, and may be overridden by an initialization prompt response. On other start types, the default CFGDSN is the one that was in use when automation was last active.

Specify the name of the control data set that contains the SA z/OS configuration that is read by the SA z/OS automation agent and automation manager.

The name can be a fully qualified data set name or a generation data group (GDG) name (either a GDG base name which defaults to generation level 0, or a GDG base name with a level qualifier, for example(-1)).

COMM
This parameter specifies that the automation manager will use XCF for communication with the automation agents. In this case, the takeover file provides the persistent storage medium for holding the current resource states and settings across automation manager sessions.

Using XCF for communication has the following risks:
• All work items travelling to, queued in, or processed by the automation manager are lost when the automation manager terminates abnormally.
• Orders for the automation agents can be broken because some orders could already have been sent at the time when the automation manager terminated abnormally.
• A warm start is required when an irrecoverable I/O error occurs while reading from or writing to the takeover file.

DELAY
Is the number of seconds to be used as a default delay prior to determining the operational mode when the automation manager instance is started. The delay option can be used when you IPL several systems concurrently and want to ensure that the primary or secondary automation manager is started on a particular system.

Note that the DELAY parameter applies only to the IPL of a system, whereas the PREF parameter applies only in the case of a takeover.

A delay value from 0–999 seconds may be specified. A value of 0 (no delay) will be assumed if it is omitted.

This value may be overridden on an individual instance basis by the start command parameter.
This parameter will be ignored when the automation manager instance is started by Automatic Restart Manager or with the specification of TYPE=HOT.

**DIAGDUPMSG**
This is the number of message buffer IDs that are validated before send and after receive. This is for diagnostic purposes. A value for \textit{nnnnn} may be chosen between 0 (no validation) and 99999. The default is 0 and performance decreases with larger values.

**DIAGINFO**
Specifies that the automation manager starts work item recording from the beginning. \textit{dsname} is the name of the data set that will hold the work items. The data set must be a sequential file. It must exist and must be catalogued.

\textbf{Note:} The data set name is accepted without checking if the data set exists or if it is accessed by another user.

**GRPID**
Specifies the 2-character suffix that composes the XCF group name that is used by the automation manager and the various agents when communicating among each other.

The value must be the same as specified for GRPID in the corresponding member INGXINIT.

**IOINTERVAL**
This defines the interval that is used to buffer any I/O to the takeover file. The value can be from 0 to 10 seconds. The default is 0 which means that no buffering is done. The maximum is 10 seconds. At the end of the interval any deferred I/O is done. The recommended value is 3.

**LEOPT**
May be used to pass run-time options to the run-time environment.

\textbf{1. Options forced by the Automation Manager.}

The following LE runtime options are set by the Automation Manager during initialization:

\begin{itemize}
  \item \texttt{ALL31(ON)} \texttt{POSIX(ON)}
\end{itemize}

\textbf{Note:} These options must not be overwritten by installation default settings (CEEOPT) with the NONOVR attribute.

\textbf{2. Default options set by the Automation Manager during initialization.}

The following LE runtime options are set by the Automation Manager during initialization:

\begin{itemize}
  \item \texttt{ANYHEAP(3M,1M,ANYWHERE,FREE)}
  \item \texttt{DEPTHCONDLMT(4)}
  \item \texttt{ERRCOUNT(0)}
  \item \texttt{HEAP(100M,10M,ANYWHERE,KEEP)}
  \item \texttt{STACK(64K,64K,ANYWHERE,KEEP)}
  \item \texttt{STORAGE(NONE,NONE,NONE,128K)}
\end{itemize}

\textbf{Note:} You may override these options.

\textbf{3. The recommended LE Options.}

The following LE options are recommended for the System Automation Manager:

\begin{itemize}
\end{itemize}
The following options can be used to gather diagnostic and storage usage information, but should be removed when no longer needed: RPTSTG(ON) RPTOPTS(ON)

The LE options below should be tuned using the LE storage reporting facility RPTSTG(ON). The initial value for HEAP storage can be calculated using the following formula:

\[
\text{heapsize} = 16 \text{ MB} + \text{nnn} - 8\text{K}
\]

where \( \text{nnn} \) is the number of resources and resource groups.

ANYHEAP(3M, 1M, ANYWHERE, FREE)
HEAP(100M, 10M, ANYWHERE, KEEP)
HEAPPOLS(ON, 40, 64, 2, 104, 2, 312, 2, 624, 1, 2024, 1)
STACK(64K, 64K, ANYWHERE, KEEP)

The following option is used to direct output created as a result of specifying RPTOPTS(ON) or RPTSTG(ON). It is also used to direct diagnostic messages written to CEEMSG and CEEMOUT by the Automation Manager:

MSGFILE(SYSOUT, FBA, 121, 0, NOENQ)

The storage options for below the line heap need to be tuned.

Notes:

- If an LEOPT=keyword is present in HSAPRM00, it replaces any LEOPT that may have been specified as an input parameter through JCL.
- When specifying options in HSAPRMxx you may have tuned LEOPT statements on multiple lines, but the total length of all of the options cannot exceed 4096 characters.

Sample LEOPTS statements are supplied in sample member HSAPRM00.

**LIFECYCLE=nnnn;dataset**

This parameter allows you to prepare for Life Cycle Recording in order to debug automation manager-related problems. Normally, SA z/OS Service will advise when Life Cycle Recording should be enabled. Specify the following:

\( \text{nnnn} \) Defines the size of the data space in number of megabytes (1 through 2097). A value of 500 is recommended and is sufficient in most situations.

\( \text{dataset} \) Specifies the fully-qualified DSN to be used when offloading the dataspace to disk.

**Note:** \( \text{nnnn} \) and \( \text{dataset} \) must be separated by a semicolon without intervening blanks. The total length of \( \text{nnnn}; \text{dataset} \) can be a maximum of 44 bytes.

**LOGSTREAM**

The parameter defines if the automation manager establishes a connection to the system logger at initialization time. The default is YES which causes the automation manager to connect to the following log streams:

- HSA.WORKITEM.HISTORY
- HSA.MESSAGE.LOG
Syntax for HSAPRM00

You may specify GRPID instead of YES to connect to a different set of log streams:

- HSA.GRP.xx.WORKITEM.HISTORY
- HSA.GRP.xx.MESSAGE.LOG

where xx represents the value of the keyword GRPID.
Then you may separate the log streams for each subplex. Note that if you specify GRPID but the value of the keyword GRPID is blank, the automation manager returns to the default value YES.
If NO is specified, no access to any SA related log stream is established and subsequently no data is written into them. No work item history besides that shown in the INGINFO command is available and no detailed information or warning or error messages are available for problem determination.

Note: Both values, LOGSTREAM and GRPID, must be the same as in the DSIPARM member INGXINIT that is used to start the related NetView agent(s).

NUMQTHDS
The NUMQTHDS parameter controls the number of query threads. This value limits the amount of parallel query activity that can be performed. If not specified, a default value of 3 will be used. A maximum of 9 query threads may be specified.

OVRDELETEDELAY
Is the number of days that a schedule override should be retained before being automatically deleted. A value of 0 days indicates that schedule overrides are not to be automatically deleted and is the default if no value is specified. A maximum of 366 days may be specified.

PREF
Specifies the preference given to the instance of the automation manager when determining which of the secondary automation managers (SAMs) should become the primary automation manager.

The value can range from 0 through 15, where 0 is the highest preference. The SAM will only participate in the escalation process when there is no other SAM active with a higher preference. The default is 0.

Note that the PREF parameter applies only in the case of a takeover, whereas the DELAY parameter applies only to the IPL of a system.

PROMPT
Specifying YES lets you overwrite the CFGDSN parameter (the name of the automation manager configuration file). Message HSAM1302A is issued and waits for a response. You can now specify the keyword/value pair:

```
CFGDSN=<fully.qualified.data.set.name>
```

Alternatively you can use a null or 'U' response to indicate that no override values are to be applied.

START
Defines the start mode of the automation manager. During initialization, the automation manager retrieves input from:

1. The CFGDSN parameter
2. Schedule overrides
3. The persistent data store (votes, triggers, resource states)
The following table shows where the automation manager retrieves initialization data for the possible values for the START parameter:

<table>
<thead>
<tr>
<th>COLD</th>
<th>WARM</th>
<th>HOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The name of automation manager configuration file is taken from PARMLIB, the START command, or via the PROMPT=YES option.</td>
<td>The last value that was used is taken</td>
</tr>
<tr>
<td>2</td>
<td>Deleted</td>
<td>Taken from the last run</td>
</tr>
<tr>
<td>3</td>
<td>Deleted</td>
<td>Deleted</td>
</tr>
</tbody>
</table>

Recommendation:
- Use COLD for the very first time, or when the schedule override file should be cleared.
- Use WARM if the automation policy has changed, that is, the automation manager configuration file has been rebuilt.
- Use HOT in any other case.

The start mode does not affect the secondary automation managers. However, the secondary automation manager reads the CFGDSN parameter from the original HSAPRMxx when the SAM was started. Any changes that you make to the HSAPRMxx are not reflected in a takeover with a cold start. If you want to perform a cold start with a modified HSAPRMxx you must first stop all your SAMs and then restart them.

The START parameter can also be specified in the automation manager JCL. If the HSAPRM00 values are to be used, the START= parameter must be removed from the JCL.

**STOPDELAY**
Is the number of seconds to be used when an MVS F <jobname>,STOP,DEFER command is entered for the primary automation manager. This delay will be invoked only if one or more secondary automation managers are active and ready when the command is received. Specify a value in the range 0–999 seconds. The recommended value is 30 seconds.

**TAKEOVERFILE**
This defines the data set name of the takeover file. It must be fully qualified.

**TAKEOVERTIMEOUT**
The value, \(nn\), may range from 1 to 600 seconds. The default is 12 seconds.

If the (secondary) automation manager performs a takeover, or an automation manager is started HOT, it will wait for specified seconds before the takeover is done from the takeover file. This delay may be required in order to allow VSAM to perform its cleanup activities on the takeover file.

**WLMQUERYINTERVAL**
This specifies the time in minutes between queries of WLM by the automation manager, as used for resource aware application move. The default is 0, which means that no querying of WLM is done. The valid range for WLMQUERYINTERVAL is from 0 to 600 minutes (that is, 10 hours).
Note: Enabling Resource Aware Application Move will impact the automation manager performance.
Syntax for HSAPRM00
Appendix H. INGDLG Command

The INGDLG command allocates required DD names and invokes the ISPF dialog. Its syntax is:

```
/SM590000/SM590000
INGDLG
ADMIN
SELECT( IOCONNECT )
ALLOCATE( NO )
/SM590000
```

```
/DDname( DSname )
SYSEXEC( DSname )
AOFPRINT( DSname )
(SYSOUT( class ))
/SM590000
```

```
/HLQ( high level qualifier )
LLQ( none )
```

```
/INITSEL( entrypoint )
```

```
/SM590000/SM630000
```

The parameters of the INGDLG command are:

**SELECT**
- Enables you to select either ADMIN or IOCONNECT. If the SELECT keyword is not specified, SELECT (ADMIN) is the default.
  - **ADMIN** Enables the selection of automation policy dialogs. This is the default.
  - **IOCONNECT** Enables the selection of I/O operations command dialogs.

**ALLOCATE**
- Controls defining DD names. If ALLOCATE is not specified, ALLOCATE (YES) is the default.
  - **YES** Allocates the necessary libraries according to the specifications in the HLQ and LLQ parameters.
    - If DDname AOFTABL is specified as an additional parameter, that data set is also allocated for ISPTLIB.
    - Furthermore, to avoid enqueue situations for multiple users, the name of the ISPF profile data set is obtained and allocated as the first data set of the table input library.
  - **NO** Does not perform any allocation of data sets. The libraries needed for the customization dialog need to be allocated before invoking INGDLG.

**DDname(DSname)**
- The DSname is the fully-qualified data set name that is to be associated...
with DD name that is specified. The name is not extended with any prefixes or suffixes that are defined using the HLQ and LLQ parameters.

For example, the following specification allocates the data set ING.CUSTOM.AOFTABL to the DD name AOFTABL:
AOFTABL(ING.CUSTOM.AOFTABL)

SYSEXEC(DSname DSname DSname ...)
For the DD name SYSEXEC multiple data set names are supported:
SYSEXEC(DSname DSname DSname ...)

This results in the following command:
TSO ALLOC ALTLIB ACTIVATE APPLICATION(EXEC)
    DATASET(DSname DSname DSname ...) UNCOND

AOFPRINT
For the DD name AOFPRINT, DSname is a fully-qualified data set name and the following syntax is valid:
AOFPRINT(SYSOUT(class))

Where class is a valid output class, creating a DD statement with
SYSOUT=class. In this case, the output is placed into the JES output class
class.

HLQ Enables you to change the high level qualifier (HLQ) of the SMP/E data sets, which is currently ING, to a HLQ of your choice. If you do not specify this parameter, ING is retained as the default.

LLQ Enables you to establish a suffix for default data set names. The default is none.

INITSEL
This parameter can be used to provide a user-selected entry point to the customization dialog. If this keyword is specified, you do not see the Customization Dialog Primary Menu as the first panel when invoking the customization dialog. INITSEL provides a fast path to some other panel, for example, the Entry Name Selection panel for a frequently used entry type. Valid values are those that you can specify as a fast path in the customization dialog, for example:
- To reach the APPC application:
  INITSEL(=APL; S APPC)
- To reach application group CICS_APG:
  INITSEL(=APG; S CICS_APG)
- To reach the Entry Name Selection panel for Applications:
  INITSEL(=APL;)

fastpath
Any words that are not the reserved keywords. The fastpath words are passed as parameters to I/O operations dialogs, if selected.

Return codes for this routine are:
0 No errors encountered
4 ISPF is not active
8 Error in data set allocation
12 Error in data set deallocation or a failed allocation
Appendix I. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user’s responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Websites are provided for convenience only and do not in any manner serve as an endorsement of those Websites. The materials at those Websites are not part of the materials for this IBM product and use of those Websites is at your own risk.
IBM may use or distribute any of the information you supply in any way it
believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose
of enabling: (i) the exchange of information between independently created
programs and other programs (including this one) and (ii) the mutual use of the
information which has been exchanged, should contact:

IBM Deutschland Research & Development GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany

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

The licensed program described in this document and all licensed material
available for it are provided by IBM under terms of the IBM Customer Agreement,
IBM International Program License Agreement or any equivalent agreement
between us.

This information contains examples of data and reports used in daily business
operations. To illustrate them as completely as possible, the examples include the
names of individuals, companies, brands, and products. All of these names are
fictitious and any similarity to the names and addresses used by an actual business
enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which
illustrate programming techniques on various operating platforms. You may copy,
modify, and distribute these sample programs in any form without payment to
IBM, for the purposes of developing, using, marketing or distributing application
programs conforming to the application programming interface for the operating
platform for which the sample programs are written. These examples have not
been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or
imply reliability, serviceability, or function of these programs. The sample
programs are provided "AS IS", without warranty of any kind. IBM shall not be
liable for any damages arising out of your use of the sample programs.

If you are viewing this information softcopy, the photographs and color
illustrations may not appear.

Programming Interface Information

This publication documents information that is not intended to be used as a
programming interface of IBM Tivoli System Automation for z/OS.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of
International Business Machines Corp., registered in many jurisdictions worldwide.
Other product and service names might be trademarks of IBM or other companies.
A current list of IBM trademarks is available on the Web at www.ibm.com/legal/
copytrade.shtml.
Java and all Java-based trademarks and logos are trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.
Glossary

This glossary includes terms and definitions from:
- The Information Technology Vocabulary developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.

The following cross-references are used in this glossary:
- **Contrast with.** This refers to a term that has an opposed or substantively different meaning.
- **Deprecated term for.** This indicates that the term should not be used. It refers to a preferred term, which is defined in its proper place in the glossary.
- **See.** This refers the reader to multiple-word terms in which this term appears.
- **See also.** This refers the reader to terms that have a related, but not synonymous, meaning.
- **Synonym for.** This indicates that the term has the same meaning as a preferred term, which is defined in the glossary.
- **Synonymous with.** This is a backward reference from a defined term to all other terms that have the same meaning.

A

**ACF** See automation configuration file

**ACF/NCP**
Advanced Communications Function for

**ACF/VTAM**

**active monitoring**
In SA z/OS automation control file, the acquiring of resource status information by soliciting such information at regular, user-defined intervals. See also passive monitoring.

**adapter**
Hardware card that enables a device, such as a workstation, to communicate with another device, such as a monitor, a printer, or some other I/O device.

**adjacent hosts**
Systems connected in a peer relationship using adjacent NetView sessions for purposes of monitoring and control.

**adjacent NetView**
In SA z/OS, the system defined as the communication path between two SA z/OS systems that do not have a direct link. An adjacent NetView is used for message forwarding and as a communication link between two SA z/OS systems. For example, the adjacent NetView is used when sending responses from a focal point to a remote system.

**Advanced Communications Function (ACF)**
A group of IBM licensed programs (principally VTAM, TCAM, NCP, and SSP) that use the concepts of Systems Network Architecture (SNA), including distribution of function and resource sharing.

**advanced program-to-program communication (APPC)**
A set of inter-program communication services that support cooperative transaction processing in a Systems
Network Architecture (SNA) network. APPC is the implementation, on a given system, of SNA’s logical unit type 6.2.

Advanced Workload Analysis Reporter (zAware)
IBM analytics appliance which requires an exclusive user partition as runtime environment. The appliance is a combination of a LPAR running in a special mode, an embedded OS (Linux on System Z) and an application layer. Customers can use the appliance to monitor the console message streams of other LPARs running in the same System z cluster and create trend reports. Exploiting zAware and these trend reports can help to better predict OS outages or performance degradations and initiate proactive clusters.

alert
In SNA, a record sent to a system problem management focal point or to a collection point to communicate the existence of an alert condition.

In NetView, a high-priority event that warrants immediate attention. A database record is generated for certain event types that are defined by user-constructed filters.

alert condition
A problem or impending problem for which some or all of the process of problem determination, diagnosis, and resolution is expected to require action at a control point.

alert focal-point system
See NPDA focal point system.

alert threshold
An application or volume service value that determines the level at which SA z/OS changes the associated icon in the graphical interface to the alert color. SA z/OS may also issue an alert. See warning threshold.

AMC
See Automation Manager Configuration.

American Standard Code for Information Interchange (ASCII)
A standard code used for information exchange among data processing systems, data communication systems, and associated equipment. ASCII uses a coded character set consisting of 7-bit coded characters (8-bit including parity check). The ASCII set consists of control characters and graphic characters. See also Extended Binary Coded Decimal Interchange Code.

APF
See authorized program facility.

API
See application programming interface.

APPC
See advanced program-to-program communication.

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

Application entry
A construct, created with the customization dialogs, used to represent and contain policy for an application.

application group
A named set of applications. An application group is part of an SA z/OS enterprise definition and is used for monitoring purposes.

application program
A program written for or by a user that applies to the user’s work, such as a program that does inventory or payroll.

A program used to connect and communicate with stations in a network, enabling users to perform application-oriented activities.

application programming interface (API)
An interface that allows an application program that is written in a high-level language to use specific data or functions of the operating system or another program.

ApplicationGroup entry
A construct, created with the customization dialogs, used to represent and contain policy for an application group.

ARM
See automatic restart management.

ASCB
Address space control block.

ASCB status
An application status derived by SA z/OS running a routine (the ASCB
checker) that searches the z/OS address space control blocks (ASCBs) for address spaces with a particular job name. The job name used by the ASCB checker is the job name defined in the customization dialog for the application.

ASCII See American Standard Code for Information Interchange

ASF See automation status file

authorized program facility (APF)
A facility that permits identification of programs that are authorized to use restricted functions.

automated console operations (ACO)
The use of an automated procedure to replace or simplify the action that an operator takes from a console in response to system or network events.

automated function
SA z/OS automated functions are automation operators, NetView autotasks that are assigned to perform specific automation functions. However, SA z/OS defines its own synonyms, or automated function names, for the NetView autotasks, and these function names are referred to in the sample policy databases provided by SA z/OS. For example, the automation operator AUTBASE corresponds to the SA z/OS automated function BASEOPER.

automatic restart management (ARM)
A z/OS recovery function that improves the availability of specified subsystems and applications by automatically restarting them under certain circumstances. Automatic restart management is a function of the Cross-System Coupling Facility (XCF) component of z/OS.

automatic restart management element name
In MVS 5.2 or later, z/OS automatic restart management requires the specification of a unique sixteen character name for each address space that registers with it. All automatic restart management policy is defined in terms of the element name, including SA z/OS’s interface with it.

automation
The automatic initiation of actions in response to detected conditions or events.

SA z/OS provides automation for z/OS applications, z/OS components, and remote systems that run z/OS. SA z/OS also provides tools that can be used to develop additional automation.

automation agent
In SA z/OS, the automation function is split up between the automation manager and the automation agents. The observing, reacting and doing parts are located within the NetView address space, and are known as the automation agents. The automation agents are responsible for:

- Recovery processing
- Message processing
- Active monitoring; they propagate status changes to the automation manager

automation configuration file
The SA z/OS customization dialogs must be used to build the automation configuration file. It consists of:

- The automation manager configuration file (AMC)
- The NetView automation table (AT)
- The NetView message revision table (MRT)
- The MPFLSTxx member

automation control file (ACF)
In SA z/OS, a file that contains system-level automation policy information. There is one master automation control file for each NetView system that SA z/OS is installed on. Additional policy information and all resource status information is contained in the policy database (PDB). The SA z/OS customization dialogs must be used to build the automation control files. They must not be edited manually.

automation flags
In SA z/OS, the automation policy settings that determine the operator functions that are automated for a resource and the times during which automation is active. When SA z/OS is running, automation is controlled by automation flag policy settings and override settings (if any) entered by the operator. Automation flags are set using the customization dialogs.
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. See also *automation configuration file*.

**Automation NetView**
In SA z/OS the NetView that performs routine operator tasks with command procedures or uses other ways of automating system and network management, issuing automatic responses to messages and management services units.

**automation operator**
NetView automation operators are NetView autotasks that are assigned to perform specific automation functions. See also *automated function*. NetView automation operators may receive messages and process automation procedures. There are no logged-on users associated with automation operators. Each automation operator is an operating system task and runs concurrently with other NetView tasks. An automation operator could be set up to handle JES2 messages that schedule automation procedures, and an automation statement could route such messages to the automation operator. Similar to *operator station task*. SA z/OS message monitor tasks and target control tasks are automation operators.

**automation policy**
The policy information governing automation for individual systems. This includes automation for applications, z/OS subsystems, z/OS data sets, and z/OS components.

**automation policy settings**
The automation policy information contained in the automation control file. This information is entered using the customization dialogs. You can display or modify these settings using the customization dialogs.

**automation procedure**
A sequence of commands, packaged as a NetView command list or a command processor written in a high-level language. An automation procedure performs automation functions and runs under NetView.

**automation routines**
In SA z/OS, a set of self-contained automation routines that can be called from the NetView automation table, or from user-written automation procedures.

**automation status file (ASF)**
In SA z/OS, a file containing status information for each automated subsystem, component or data set. This information is used by SA z/OS automation when taking action or when determining what action to take. In Release 2 and above of AOC/MVS, status information is also maintained in the operational information base.

**automation table (AT)**
See *NetView automation table*.

**autotask**
A NetView automation task that receives messages and processes automation procedures. There are no logged-on users associated with autotasks. Each autotask is an operating system task and runs concurrently with other NetView tasks. An autotask could be set up to handle JES2 messages that schedule automation procedures, and an automation statement could route such messages to the autotasks. Similar to *operator station task*. SA z/OS message monitor tasks and target control tasks are autotasks. Also called *automation operator*.

**available**
In VTAM programs, pertaining to a
logical unit that is active, connected, enabled, and not at its session limit.

B

**Base Control Program (BCP)**
A program that provides essential services for the MVS and z/OS operating systems. The program includes functions that manage system resources. These functions include input/output, dispatch units of work, and the z/OS UNIX System Services kernel. See also [Multiple Virtual Storage](#) and [z/OS](#).

**basic mode**
A central processor mode that does not use logical partitioning. Contrast with [logically partitioned mode](#).

**BCP** See [Base Control Program](#).

**BCP Internal Interface**
Processor function of System z processor families. It allows for communication between basic control programs such as z/OS and the processor support element in order to exchange information or to perform processor control functions. Programs using this function can perform hardware operations such as ACTIVATE or SYSTEM RESET.

**beaconing**
The repeated transmission of a frame or messages (beacon) by a console or workstation upon detection of a line break or outage.

**blade**
A hardware unit that provides application-specific services and components. The consistent size and shape (or form factor) of each blade allows it to fit in a BladeCenter chassis.

**BladeCenter chassis**
A modular chassis that can contain multiple blades, allowing the individual blades to share resources such as management, switch, power, and blower modules.

**BookManager®**
An IBM product that lets users view softcopy documents on their workstations.

C

**central processor (CP)**
The part of the computer that contains the sequencing and processing facilities for instruction execution, initial program load (IPL), and other machine operations.

**central processor complex (CPC)**
A physical collection of hardware that consists of central storage, (one or more) central processors, (one or more) timers, and (one or more) channels.

**central site**
In a distributed data processing network, the central site is usually defined as the focal point for alerts, application design, and remote system management tasks such as problem management.

**CFR/CFS and ISC/ISR**
I/O operations can display and return data about integrated system channels (ISC) connected to a coupling facility and coupling facility receiver (CFR) channels and coupling facility sender (CFS) channels.

**channel**
A path along which signals can be sent; for example, data channel, output channel. See also [link](#).

**channel path identifier**
A system-unique value assigned to each channel path.

**channel-attached**
Attached directly by I/O channels to a host processor (for example, a channel-attached device).

**CHPID**
In SA z/OS, channel path ID; the address of a channel.

**CHPID port**
A label that describes the system name, logical partitions, and channel paths.

**CI** See [console integration](#).

**CICS/VS**
Customer Information Control System for Virtual Storage. See [Customer Information Control System](#).

**CLIST** See [command list](#).

**clone**
A set of definitions for application...
instances that are derived from a basic application definition by substituting a number of different system-specific values into the basic definition.

**clone ID**
A generic means of handling system-specific values such as the MVS SYSCLONE or the VTAM subarea number. Clone IDs can be substituted into application definitions and commands to customize a basic application definition for the system that it is to be instantiated on.

**CNC**
A channel path that transfers data between a host system image and an ESCON control unit. It can be point-to-point or switchable.

**command**
A request for the performance of an operation or the execution of a particular program.

**command facility**
The component of NetView that is a base for command processors that can monitor, control, automate, and improve the operation of a network. The successor to NCCF.

**command list (CLIST)**
A list of commands and statements, written in the NetView command list language or the REXX language, designed to perform a specific function for the user. In its simplest form, a command list is a list of commands. More complex command lists incorporate variable substitution and conditional logic, making the command list more like a conventional program. Command lists are typically interpreted rather than being compiled.

In SA z/OS, REXX command lists that can be used for automation procedures.

**command procedure**
In NetView, either a command list or a command processor.

**command processor**
A module designed to perform a specific function. Command processors, which can be written in assembler or a high-level language (HLL), are issued as commands.

**command processor control block**
An I/O operations internal control block that contains information about the command being processed.

**Command Tree/2**
An OS/2-based program that helps you build commands on an OS/2 window, then routes the commands to the destination you specify (such as a 3270 session, a file, a command line, or an application program). It provides the capability for operators to build commands and route them to a specified destination.

**common commands**
The SA z/OS subset of the CPC operations management commands.

**Common User Access (CUA) architecture**
Guidelines for the dialog between a human and a workstation or terminal.

**communication controller**
A type of communication control unit whose operations are controlled by one or more programs stored and executed in the unit or by a program executed in a processor to which the controller is connected. It manages the details of line control and the routing of data through a network.

**communication line**
Deprecated term for telecommunication line.

**connectivity view**
In SA z/OS, a display that uses graphic images for I/O devices and lines to show how they are connected.

**console automation**
The process of having NetView facilities provide the console input usually handled by the operator.

**console connection**
In SA z/OS, the 3270 or ASCII (serial) connection between a PS/2 computer and a target system. Through this connection, the workstation appears (to the target system) to be a console.

**console integration (CI)**
A hardware facility that if supported by an operating system, allows operating system messages to be transferred through an internal hardware interface for
display on a system console. Conversely, it allows operating system commands entered at a system console to be transferred through an internal hardware interface to the operating system for processing.

**consoles**

Workstations and 3270-type devices that manage your enterprise.

**Control units**

Hardware units that control I/O operations for one or more devices. You can view information about control units through I/O operations, and can start or stop data going to them by blocking and unblocking ports.

**controller**

A unit that controls I/O operations for one or more devices.

**converted mode (CVC)**

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

**couple data set**

A data set that is created through the XCF couple data set format utility and, depending on its designated type, is shared by some or all of the z/OS systems in a sysplex. See also sysplex couple data set and XCF couple data set.

**coupling facility**

The hardware element that provides high-speed caching, list processing, and locking functions in a sysplex.

**CPU**

Central processing unit. Deprecated term for processor.

**cross-system coupling facility (XCF)**

A component of z/OS that provides functions to support cooperation between authorized programs running within a sysplex.

**CTC**

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

**Customer Information Control System (CICS)**

A general-purpose transactional program that controls online communication between terminal users and a database for a large number of end users on a real-time basis.

**customization dialogs**

The customization dialogs are an ISPF application. They are used to customize the enterprise policy, like, for example, the enterprise resources and the relationships between resources, or the automation policy for systems in the enterprise. How to use these dialogs is described in IBM Tivoli System Automation for z/OS Customizing and Programming.

**CVC**

See converted mode.

**D**

**DataPower X150z**

See IBM Websphere DataPower Integration Appliance X150 for zEnterprise (DataPower X150z).

**DASD**

See direct access storage device.

**data services task (DST)**

The NetView subtask that gathers, records, and manages data in a VSAM file or a network device that contains network management information.

**data set**

The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

**data set members**

Members of partitioned data sets that are individually named elements of a larger file that can be retrieved by name.

**DBCS**

See double-byte character set.
DCCF  See disabled console communication facility.

DCF  See Document Composition Facility.

DELAY Report
An RMF report that shows the activity of each job in the system and the hardware and software resources that are delaying each job.

device  A piece of equipment. Devices can be workstations, printers, disk drives, tape units, remote systems or communications controllers. You can see information about all devices attached to a particular switch, and control paths and jobs to devices.

DEVR Report
An RMF report that presents information about the activity of I/O devices that are delaying jobs.

dialog  Interactive 3270 panels.

direct access storage device (DASD)
A device that allows storage to be directly accessed, such as a disk drive.

disabled console communication facility (DCCF)
A z/OS component that provides limited-function console communication during system recovery situations.

disk operating system (DOS)
An operating system for computer systems that use disks and diskettes for auxiliary storage of programs and data. Software for a personal computer that controls the processing of programs. For the IBM Personal Computer, the full name is Personal Computer Disk Operating System (PCDOS).

display
To present information for viewing, usually on the screen of a workstation or on a hardcopy device.

Deprecated term for panel.

distribution manager
The component of the NetView program that enables the host system to use, send, and delete files and programs in a network of computers.

Document Composition Facility (DCF)
An IBM licensed program used to format input to a printer.

domain
An access method and its application programs, communication controllers, connecting lines, modems, and attached workstations.

In SNA, a system services control point (SSCP) and the physical units (PUs), logical units (LUs), links, link stations, and associated resources that the SSCP can control with activation requests and deactivation requests.

double-byte character set (DBCS)
A character set, such as Kanji, in which each character is represented by a 2-byte code.

DP enterprise
Data processing enterprise.

DSIPARM
This file is a collection of members of NetView's customization.

DST  Data Services Task.

E

EBCDIC
See Extended Binary Coded Decimal Interchange Code.

ECB  See event control block.

EMCS  Extended multiple console support. See also multiple console support.

ensemble
A collection of one or more zEnterprise nodes (including any attached zBX) that are managed as a single logical virtualized system by the Unified Resource Manager, through the Hardware Management Console.

ensemble member
A zEnterprise node that has been added to an ensemble.

enterprise
The composite of all operational entities, functions, and resources that form the total business concern and that require an information system.

enterprise monitoring
Enterprise monitoring is used by SA z/OS to update the NetView Management Console (NMC) resource status information that is stored in the Resource Object Data Manager (RODM).
Resource status information is acquired by enterprise monitoring of the Resource Measurement Facility (RMF) Monitor III service information at user-defined intervals. SA z/OS stores this information in its operational information base, where it is used to update the information presented to the operator in graphic displays.

**Enterprise Systems Architecture (ESA)**
A hardware architecture that reduces the effort required for managing data sets and extends addressability for system, subsystem, and application functions.

**entries**
Resources, such as processors, entered on panels.

**entry type**
Resources, such as processors or applications, used for automation and monitoring.

**environment**
Data processing enterprise.

**error threshold**
An automation policy setting that specifies when SA z/OS should stop trying to restart or recover an application, subsystem or component, or offload a data set.

**ESA** See [Enterprise Systems Architecture](#).

**event**
In NetView, a record indicating irregularities of operation in physical elements of a network.

An occurrence of significance to a task; for example, the completion of an asynchronous operation, such as an input/output operation.

Events are part of a trigger condition, such that if all events of a trigger condition have occurred, a startup or shutdown of an application is performed.

**event control block (ECB)**
A control block used to represent the status of an event.

**exception condition**
An occurrence on a system that is a deviation from normal operation. SA z/OS monitoring highlights exception conditions and allows an SA z/OS enterprise to be managed by exception.

**Extended Binary Coded Decimal Interchange Code (EBCDIC)**
A coded character set of 256 8-bit characters developed for the representation of textual data. See also [American Standard Code for Information Interchange](#).

**extended recovery facility (XRF)**
A facility that minimizes the effect of failures in z/OS, VTAM, the host processor, or high availability applications during sessions between high availability applications and designated terminals. This facility provides an alternate subsystem to take over sessions from the failing subsystem.

**F**

**fallback system**
See [secondary system](#).

**field**
A collection of bytes within a record that are logically related and are processed as a unit.

**file manager commands**
A set of SA z/OS commands that read data from or write data to the automation control file or the operational information base. These commands are useful in the development of automation that uses SA z/OS facilities.

**focal point**
In NetView, the focal-point domain is the central host domain. It is the central control point for any management services element containing control of the network management data.

**focal point system**
A system that can administer, manage, or control one or more target systems. There are a number of different focal point systems associated with IBM automation products.

**NMC focal point system**
The NMC focal point system is a NetView system with an attached workstation server and LAN that gathers information about the state of the network. This focal point system uses RODM to store the data it collects in the data model. The information stored in RODM can be accessed from any LAN-connected workstation with NetView Management Console installed.
NPDA focal point system. This is a NetView system that collects all the NPDA alerts that are generated within your enterprise. It is supported by NetView. If you have SA z/OS installed the NPDA focal point system must be the same as your NMC focal point system. The NPDA focal point system is also known as the alert focal point system.

SA z/OS Processor Operations focal point system. This is a NetView system that has SA z/OS host code installed. The SA z/OS Processor Operations focal point system receives messages from the systems and operator consoles of the machines that it controls. It provides full systems and operations console function for its target systems. It can be used to IPL these systems. Note that some restrictions apply to the Hardware Management Console for an S/390 microprocessor cluster.

SA z/OS SDF focal point system. The SA z/OS SDF focal point system is an SA z/OS NetView system that collects status information from other SA z/OS NetViews within your enterprise.

Status focal point system. In NetView, the system to which STATMON, VTAM and 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.

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.

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.

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 (HMC)
A user interface through which data center personnel configure, control, monitor, and manage System z hardware and software resources. The HMC communicates with each central processor complex (CPC) through the Support Element. On an IBM zEnterprise 196 (z196), using the Unified Resource Manager on the HMCs or Support Elements, personnel can also create and manage an ensemble.

Hardware Management Console Application (HWMCA)
A direct-manipulation object-oriented graphical user interface that provides a single point of control and single system image for hardware elements. The HWMCA provides grouping support, aggregated and real-time system status.
using colors, consolidated hardware messages support, consolidated service messages support, and hardware commands targeted at a single system, multiple systems, or a group of systems.

**heartbeat**

In SA z/OS, a function that monitors the validity of the status forwarding path between remote systems and the NMC focal point, and monitors the availability of remote z/OS systems, to ensure that status information displayed on the SA z/OS workstation is current.

**help panel**

An online panel that tells you how to use a command or another aspect of a product.

**hierarchy**

In the NetView program, the resource types, display types, and data types that make up the organization, or levels, in a network.

**high-level language (HLL)**

A programming language that provides some level of abstraction from assembler language and independence from a particular type of machine. For the NetView program, the high-level languages are PL/I and C.

**IBM blade**

A customer-acquired, customer-installed select blade to be managed by IBM zEnterprise Unified Resource Manager. One example of an IBM blade is a POWER7 blade.

**IBM Smart Analyzer for DB2 for z/OS**

An optimizer that processes certain types of data warehouse queries for DB2 for z/OS.

**IBM System z Application Assist Processor (zAAP)**

A specialized processor that provides a Java execution environment, which enables Java-based web applications to be integrated with core z/OS business applications and backend database systems.

**IBM System z Integrated Information Processor (zIIP)**

A specialized processor that provides computing capacity for selected data and transaction processing workloads and for selected network encryption workloads.

**IBM Websphere DataPower Integration Appliance X150 for zEnterprise (DataPower X150z)**

A purpose-built appliance that simplifies, helps secure, and optimizes XML and Web services processing.

**IBM Enterprise 196 (z196)**

The newest generation of System z family of servers built on a new processor chip, with enhanced memory function and capacity, security, and on demand enhancements to support existing mainframe workloads and large scale consolidation.

**IBM zEnterprise BladeCenter Extension (zBX)**

A heterogeneous hardware infrastructure that consists of a BladeCenter chassis attached to an IBM zEnterprise 196 (z196). A BladeCenter chassis can contain IBM blades or optimizers.

**IBM zEnterprise BladeCenter Extension (zBX) blade**

Generic name for all blade types supported in an IBM zEnterprise BladeCenter Extension (zBX). This term includes IBM blades and optimizers.
IBM zEnterprise System (zEnterprise)
A heterogeneous hardware infrastructure that can consist of an IBM zEnterprise 196 (z196) and an attached IBM zEnterprise BladeCenter Extension (zBX) Model 002, managed as a single logical virtualized system by the Unified Resource Manager.

IBM zEnterprise Unified Resource Manager
Licensed Internal Code (LIC), also known as firmware, that is part of the Hardware Management Console. The Unified Resource Manager provides energy monitoring and management, goal-oriented policy management, increased security, virtual networking, and data management for the physical and logical resources of a given ensemble.

I/O operations
The part of SA z/OS that provides you with a single point of logical control for managing connectivity in your active I/O configurations. I/O operations takes an active role in detecting unusual conditions and lets you view and change paths between a processor and an I/O device, using dynamic switching (the ESCON director). Also known as I/O Ops.

I/O Ops
See I/O operations.

I/O resource number
Combination of channel path identifier (CHPID), device number, etc. See Internal token.

images
A grouping of processors and I/O devices that you define. You can define a single-image mode that allows a multiprocessor system to function as one central processor image.

IMS
See Information Management System.

IMS/VS
See Information Management System/Virtual Storage.

inbound
In SA z/OS, messages sent to the focal-point system from the PC or target system.

inbound gateway operator
The automation operator that receives incoming messages, commands, and responses from the outbound gateway operator at the sending system. The inbound gateway operator handles communications with other systems using a gateway session.

Information Management System (IMS)
Any of several system environments available with a database manager and transaction processing that are capable of managing complex databases and terminal networks.

Information Management System/Virtual Storage (IMS/VS)
A database/data communication (DB/DC) system that can manage complex databases and networks. Synonymous with Information Management System.

INGEIO PROC
The I/O operations default procedure name. It is part of the SYS1.PROCLIB.

initial microprogram load
The action of loading microprograms into computer storage.

initial program load (IPL)
The initialization procedure that causes an operating system to commence operation.

The process by which a configuration image is loaded into storage at the beginning of a workday or after a system malfunction.

The process of loading system programs and preparing a system to run jobs.

initialize automation
SA z/OS-provided automation that issues the correct z/OS start command for each subsystem when SA z/OS is initialized. The automation ensures that subsystems are started in the order specified in the automation control files and that prerequisite applications are functional.

input/output configuration data set (IOCDS)
A configuration definition built by the I/O configuration program (IOCP) and stored on disk files associated with the processor controller.

input/output support processor (IOSP)
The hardware unit that provides I/O support functions for the primary support processor and maintenance support functions for the processor controller.
Interactive System Productivity Facility (ISPF)
An IBM licensed program that serves as a full-screen editor and dialog manager. Used for writing application programs, it provides a means of generating standard screen panels and interactive dialogs between the application programmer and the terminal user. See also Time Sharing Option.

interested operator list
The list of operators who are to receive messages from a specific target system.

internal token
A logical token (LTOk); name by which the I/O resource or object is known; stored in IODF.

IOCDS
See input/output configuration data set.

IOSP
See input/output support processor.

IPL
See initial program load.

ISPF
See Interactive System Productivity Facility.

ISPF console
You log on to ISPF from this 3270-type console to use the runtime panels for I/O operations and SA z/OS customization panels.

issuing host
The base program that you enter a command for processing with. See primary host.

J
JCL
See job control language.

JES
See job entry subsystem.

JES2
An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system. In an installation with more than one processor, each JES2 processor independently controls its job input, scheduling, and output processing. See also job entry subsystem and JES3.

JES3
An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system. In complexes that have several loosely coupled processing units, the JES3 program manages processors so that the global processor exercises centralized control over the local processors and distributes jobs to them using a common job queue. See also job entry subsystem and JES2.

job
A set of data that completely defines a unit of work for a computer. A job usually includes all necessary computer programs, linkages, files, and instructions to the operating system.

An address space.

job control language (JCL)
A problem-oriented language designed to express statements in a job that are used to identify the job or describe its requirements to an operating system.

job entry subsystem (JES)
An IBM licensed program that receives jobs into the system and processes all output data that is produced by jobs. In SA z/OS publications, JES refers to JES2 or JES3, unless otherwise stated. See also JES2 and JES3.

K
Kanji
An ideographic character set used in Japanese. See also double-byte character set.

L
LAN
See local area network.

line mode
A form of screen presentation in which the information is presented a line at a time in the message area of the terminal screen. Contrast with full-screen mode.

link
In SNA, the combination of the link connection and the link stations joining network nodes; for example, a System/370 channel and its associated protocols, a serial-by-bit connection under the control of synchronous data link control (SDLC). See synchronous data link control.

In SA z/OS, link connection is the physical medium of transmission.

link-attached
Describes devices that are physically connected by a telecommunication line. Contrast with channel-attached.
Linux on System z
UNIX-like open source operating system conceived by Linus Torvalds and developed across the internet.

local
Pertaining to a device accessed directly without use of a telecommunication line. Synonymous with channel-attached.

local area network (LAN)
A network in which a set of devices is connected for communication. They can be connected to a larger network. See also token ring.
A network that connects several devices in a limited area (such as a single building or campus) and that can be connected to a larger network.

logical partition (LP)
A subset of the processor hardware that is defined to support an operating system. See also logically partitioned mode.

logical switch number (LSN)
Assigned with the switch parameter of the CHPID macro of the IOCP.

logical token (LTOK)
Resource number of an object in the IODF.

logical unit (LU)
In SNA, a port through which an end user accesses the SNA network and the functions provided by system services control points (SSCPs). An LU can support at least two sessions, one with an SSCP and one with another LU, and may be capable of supporting many sessions with other LUs. See also physical unit and system services control point.

logical unit 6.2 (LU 6.2)
A type of logical unit that supports general communications between programs in a distributed processing environment. LU 6.2 is characterized by:
• A peer relationship between session partners
• Efficient use of a session for multiple transactions
• A comprehensive end-to-end error processing
• A generic application program interface (API) consisting of structured verbs that are mapped to a product implementation

Synonym for advanced program-to-program communication.

logically partitioned (LPAR) mode
A central processor mode that enables an operator to allocate system processor hardware resources among several logical partitions. Contrast with basic mode.

LOGR
The sysplex logger.

LP
See logical partition.

LPAR
See logically partitioned mode.

LSN
See logical switch number.

LU
See logical unit.

LU 6.2
See logical unit 6.2.

LU 6.2 session
A session initiated by VTAM on behalf of an LU 6.2 application program, or a session initiated by a remote LU in which the application program specifies that VTAM is to control the session by using the APPCCMD macro. See logical unit 6.2.

LU-LU session
In SNA, a session between two logical units (LUs) in an SNA network. It provides communication between two end users, or between an end user and an LU services component.

M

MAT
Deprecated term for NetView automation table.

MCA
See Micro Channel architecture.

MCS
See multiple console support.

member
A specific function (one or more modules or routines) of a multisystem application that is defined to XCF and assigned to a group by the multisystem application. A member resides on one system in the sysplex and can use XCF services to communicate (send and receive data) with other members of the same group.

message automation table (MAT)
Deprecated term for NetView automation table.

message class
A number that SA z/OS associates with a message to control routing of the message. During automated operations, the classes associated with each message
issued by SA z/OS are compared to the classes assigned to each notification operator. Any operator with a class matching one of the message’s classes receives the message.

**message forwarding**
The SA z/OS process of sending messages generated at an SA z/OS target system to the SA z/OS focal-point system.

**message group**
Several messages that are displayed together as a unit.

**message monitor task**
A task that starts and is associated with a number of communications tasks. Message monitor tasks receive inbound messages from a communications task, determine the originating target system, and route the messages to the appropriate target control tasks.

**message processing facility (MPF)**
A z/OS table that screens all messages sent to the z/OS console. The MPF compares these messages with a customer-defined list of messages (based on this message list, messages are automated and/or suppressed from z/OS console display), and marks messages to automate or suppress. Messages are then broadcast on the subsystem interface (SSI).

**message suppression**
The ability to restrict the amount of message traffic displayed on the z/OS console.

**Micro Channel architecture**
The rules that define how subsystems and adapters use the Micro Channel bus in a computer. The architecture defines the services that each subsystem can or must provide.

**microprocessor**
A processor implemented on one or a small number of chips.

**migration**
Installation of a new version or release of a program to replace an earlier version or release.

**MP**
Multiprocessor.

**MPF**
See message processing facility.

**MPFLSTxx**
The MPFLST member that is built by SA z/OS.

**multi-MVS environment**
Physical processing system that is capable of operating more than one MVS image. See also MVS image.

**multiple console support (MCS)**
A feature of MVS that permits selective message routing to multiple consoles.

**Multiple Virtual Storage (MVS)**
An IBM operating system that accesses multiple address spaces in virtual storage. The predecessor of z/OS.

**multiprocessor (MP)**
A CPC that can be physically partitioned to form two operating processor complexes.

**multisystem application**
An application program that has various functions distributed across z/OS images in a multisystem environment.

**multisystem environment**
An environment in which two or more systems reside on one or more processors. Or one or more processors can communicate with programs on the other systems.

**MVS**
See Multiple Virtual Storage.

**MVS image**
A single occurrence of the MVS operating system that has the ability to process work. See also multi-MVS environment and single-MVS environment.

**MVS/ESA**
Multiple Virtual Storage/Enterprise Systems Architecture. See z/OS.

**MVS/JES2**
Multiple Virtual Storage/Job Entry System 2. A z/OS subsystem that receives jobs into the system, converts them to an internal format, selects them for execution, processes their output, and purges them from the system. In an installation with more than one processor, each JES2 processor independently controls its job input, scheduling, and output processing.

**N**
See network addressable unit.

**NAU**
See network addressable unit.
See network accessible unit.

NCCF  See Network Communications Control Facility.

NCP  See network control program (general term).
See Network Control Program (an IBM licensed program). Its full name is Advanced Communications Function for the Network Control Program. Synonymous with ACF/NCP.

NCP/token ring interconnection
A function used by ACF/NCP to support token ring-attached SNA devices. NTRI also provides translation from token ring-attached SNA devices (PUs) to switched (dial-up) devices.

NetView
An IBM licensed program used to monitor a network, manage it, and diagnose network problems. NetView consists of a command facility that includes a presentation service, command processors, automation based on command lists, and a transaction processing structure on which the session monitor, hardware monitor, and terminal access facility (TAF) network management applications are built.

NetView (NCCF) console
A 3270-type console for NetView commands and runtime panels for system operations and processor operations.

NetView automation procedures
A sequence of commands, packaged as a NetView command list or a command processor written in a high-level language. An automation procedure performs automation functions and runs under the NetView program.

NetView automation table (AT)
A table against which the NetView program compares incoming messages. A match with an entry triggers the specified response. SA z/OS entries in the NetView automation table trigger an SA z/OS response to target system conditions. Formerly known as the message automation table (MAT).

NetView command list language
An interpretive language unique to NetView that is used to write command lists.

NetView Graphic Monitor Facility (NGMF)
Deprecated term for NetView Management Console.

NetView hardware monitor
The component of NetView that helps identify network problems, such as hardware, software, and microcode, from a central control point using interactive display techniques. Formerly called network problem determination application.

NetView log
The log that NetView records events relating to NetView and SA z/OS activities in.

NetView Management Console (NMN)
A function of the NetView program that provides a graphic, topological presentation of a network that is controlled by the NetView program. It provides the operator different views of a network, multiple levels of graphical detail, and dynamic resource status of the network. This function consists of a series of graphic windows that allows you to manage the network interactively. Formerly known as the NetView Graphic Monitor Facility (NGMF).

NetView message table
See NetView automation table.

NetView paths via logical unit (LU 6.2)
A type of network-accessible port (VTAM connection) that enables end users to gain access to SNA network resources and communicate with each other. LU 6.2 permits communication between processor operations and the workstation. See logical unit 6.2.

NetView-NetView task (NNT)
The task that a cross-domain NetView operator session runs under. Each NetView program must have a NetView-NetView task to establish one NNT session. See also operator station task.

NetView-NetView task session
A session between two NetView programs that runs under a NetView-NetView task. In SA z/OS, NetView-NetView task
sessions are used for communication between focal point and remote systems.

**network**
An interconnected group of nodes.

In data processing, a user application network. See SNA network.

**network accessible unit (NAU)**
In SNA networking, any device on the network that has a network address, including a logical unit (LU), physical unit (PU), control point (CP), or system services control point (SSCP). It is the origin or the destination of information transmitted by the path control network. Synonymous with network addressable unit.

**network addressable unit (NAU)**
Synonym for network accessible unit.

**Network Communications Control Facility (NCCF)**
The operations control facility for the network. NCCF consists of a presentation service, command processors, automation based on command lists, and a transaction processing structure on which the network management applications NLDN and NPDA are built. NCCF is a precursor to the NetView command facility.

**Network Control Program (NCP)**
An IBM licensed program that provides communication controller support for single-domain, multiple-domain, and interconnected network capability. Its full name is Advanced Communications Function for the Network Control Program.

**network control program (NCP)**
A program that controls the operation of a communication controller.

A program used for requests and responses exchanged between physical units in a network for data flow control.

**Network Problem Determination Application (NPDA)**
An NCCF application that helps you identify network problems, such as hardware, software, and microcode, from a central control point using interactive display methods. The alert manager for the network. The precursor of the NetView hardware monitor.

**Networking NetView**
In SA z/OS the NetView that performs network management functions, such as managing the configuration of a network. In SA z/OS it is common to also route alerts to the Networking NetView.

**NGMF**
Deprecated term for NetView Management Console.

**NGMF focal-point system**
Deprecated term for NMC focal point system.

**NIP**
See nucleus initialization program.

**NMC focal point system**
See focal point system.

**NMC workstation**
The NMC workstation is the primary way to dynamically monitor SA z/OS systems. From the windows, you see messages, monitor status, view trends, and react to changes before they cause problems for end users. You can use multiple windows to monitor multiple views of the system.

**NNT**
See NetView-NetView task.

**notification message**
An SA z/OS message sent to a human notification operator to provide information about significant automation actions. Notification messages are defined using the customization dialogs.

**notification operator**
A NetView console operator who is authorized to receive SA z/OS notification messages. Authorization is made through the customization dialogs.

**NPDA**
See Network Problem Determination Application.

**NPDA focal-point system**
See focal point system.

**NTRI**
See NCP/token ring interconnection.

**nucleus initialization program (NIP)**
The program that initializes the resident control program; it allows the operator to...
request last-minute changes to certain options specified during system generation.

**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.

**OPC/A**

See Operations Planning and Control/Advanced.

**Operations Planning and Control/Enterprise Systems Architecture (OPC/ESA)**

A set of IBM licensed programs that automate, plan, and control batch workload. OPC/ESA analyzes system and workload status and submits jobs accordingly. The successor to OPC/A.

**Operator**

A person who keeps a system running.

A person or program responsible for managing activities controlled by a given piece of software such as z/OS, the NetView program, or IMS.

A person who operates a device.

In a language statement, the lexical entity that indicates the action to be performed on operands.

**Operator console**

A functional unit containing devices that are used for communications between a computer operator and a computer. (T)

A display console used for communication between the operator and the system, used primarily to specify information concerning application programs and I/O operations and to monitor system operation.

In SA z/OS, a console that displays output from and sends input to the operating system (z/OS, LINUX, VM, VSE). Also called operating system console. In the SA z/OS operator commands and configuration dialogs, OC is used to designate a target system operator console.

**Operator station task (OST)**

The NetView task that establishes and maintains the online session with the network operator. There is one operator station task for each network operator who logs on to the NetView program.

**Operator view**

A set of group, system, and resource definitions that are associated together for monitoring purposes. An operator view appears as a graphic display in the graphical interface showing the status of the defined groups, systems, and resources.

**OperatorView entry**

A construct, created with the
customization dialogs, used to represent and contain policy for an operator view.

**optimizer**
A special-purpose hardware component or appliance that can perform a limited set of specific functions with optimized performance when compared to a general-purpose processor. Because of its limited set of functions, an optimizer is an integrated part of a processing environment, rather than a stand-alone unit. One example of an optimizer is the IBM Smart Analytics Optimizer for DB2 for z/OS.

**OS**  
See [operating system](#).

**OSA**  
See [Open Systems Adapter](#).

**OST**  
See [operator station task](#).

**outbound**
In SA z/OS, messages or commands from the focal-point system to the target system.

**outbound gateway operator**
The automation operator that establishes connections to other systems. The outbound gateway operator handles communications with other systems through a gateway session. The automation operator sends messages, commands, and responses to the inbound gateway operator at the receiving system.

**P**

**page**  
The portion of a panel that is shown on a display surface at one time.

To transfer instructions, data, or both between real storage and external page or auxiliary storage.

**panel**  
A formatted display of information that appears on a terminal screen. Panels are full-screen 3270-type displays with a monospaced font, limited color and graphics.

By using SA z/OS panels you can see status, type commands on a command line using a keyboard, configure your system, and passthru to other consoles. See also [help panel](#).

In computer graphics, a display image that defines the locations and characteristics of display fields on a display surface. Contrast with [screen](#).

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

**parameter**
A variable that is given a constant value for a specified application and that may represent an application, for example.

An item in a menu for which the user specifies a value or for which the system provides a value when the menu is interpreted.

Data passed to a program or procedure by a user or another program, specifically as an operand in a language statement, as an item in a menu, or as a shared data structure.

**partition**
A fixed-size division of storage.

In VSE, a division of the virtual address area that is available for program processing.

On an IBM Personal Computer fixed disk, one of four possible storage areas of variable size; one can be accessed by DOS, and each of the others may be assigned to another operating system.

**partitionable CPC**
A CPC that can be divided into 2 independent CPCs. See also [physical partition](#), [single-image mode](#), [MIP](#), and [side](#).

**partitioned data set (PDS)**
A data set in direct access storage that is divided into partitions, called *members*, each of which can contain a program, part of a program, or data.

**passive monitoring**
In SA z/OS, the receiving of unsolicited messages from z/OS systems and their resources. These messages can prompt updates to resource status displays. See also [active monitoring](#).

**PCE**  
A processor controller. Also known as the support processor or service processor in some processor families.

**PDB**  
See [policy database](#).

**PDS**  
See [partitioned data set](#).

---

Glossary 287
physical partition
Part of a CPC that operates as a CPC in its own right, with its own copy of the operating system.

physical unit (PU)
In SNA, the component that manages and monitors the resources (such as attached links and adjacent link stations) of a node, as requested by a system services control point (SSCP) through an SSCP-PU session. An SSCP activates a session with the physical unit to indirectly manage, through the PU, resources of the node such as attached links.

physically partitioned (PP) configuration
A mode of operation that allows a multiprocessor (MP) system to function as two or more independent CPCs having separate power, utilities, and maintenance boundaries. Contrast with single-image mode.

PLEXID group
PLEXID group or "extended XCF communication group" is a term used in conjunction with a sysplex. The PLEXID group includes System Automation Agents for a subset of a sysplex or for the entire sysplex. It is used to provide XCF communication beyond the SAplex boundaries. For a detailed description, refer to "Defining the Extended XCF Communication Group" in IBM Tivoli System Automation for z/OS Planning and Installation.

POI See program operator interface

policy
The automation and monitoring specifications for an SA z/OS enterprise. See IBM Tivoli System Automation for z/OS Defining Automation Policy.

policy database
The automation definitions (automation policy) that the automation programmer specifies using the customization dialog is stored in the policy database. Also known as the PDB. See also automation policy

POR See power-on reset

port
System hardware that the I/O devices are attached to.
In an ESCON switch, a port is an addressable connection. The switch routes data through the ports to the channel or control unit. Each port has a name that can be entered into a switch matrix, and you can use commands to change the switch configuration.

An access point (for example, a logical unit) for data entry or exit.
A functional unit of a node that data can enter or leave a data network through.
In data communication, that part of a data processor that is dedicated to a single data channel for the purpose of receiving data from or transmitting data to one or more external, remote devices.

power-on reset (POR)
A function that re-initializes all the hardware in a CPC and loads the internal code that enables the CPC to load and run an operating system. See initial microprogram load.

PP See physical partition

PPI See program to program interface

PPT See primary POI task

PR/SM See Processor Resource/Systems Manager

primary host
The base program that you enter a command for processing at.

primary POI task (PPT)
The NetView subtask that processes all unsolicited messages received from the VTAM program operator interface (POI) and delivers them to the controlling operator or to the command processor. The PPT also processes the initial command specified to execute when NetView is initialized and timer request commands scheduled to execute under the PPT.

primary system
A system is a primary system for an application if the application is normally meant to be running there. SA z/OS starts the application on all the primary systems defined for it.

problem determination
The process of determining the source of a problem; for example, a program component, machine failure, telecommunication facilities, user or contractor-installed programs or
equipment, environment failure such as a power loss, or user error.

**processor**
A device for processing data from programmed instructions. It may be part of another unit.

In a computer, the part that interprets and executes instructions. Two typical components of a processor are a control unit and an arithmetic logic unit.

**processor controller**
Hardware that provides support and diagnostic functions for the central processors.

**processor operations**
The part of SA z/OS that monitors and controls processor (hardware) operations. Processor operations provides a connection from a focal-point system to a target system. Through NetView on the focal-point system, processor operations automates operator and system consoles for monitoring and recovering target systems. Also known as ProcOps.

**Processor Resource/Systems Manager (PR/SM)**
The feature that allows the processor to use several operating system images simultaneously and provides logical partitioning capability. See also logically partitioned mode.

**ProcOps**
See processor operations.

**ProcOps Service Machine (PSM)**
The PSM is a CMS user on a VM host system. It runs a CMS multitasking application that serves as "virtual hardware" for ProcOps. ProcOps communicates via the PSM with the VM guest systems that are defined as target systems within ProcOps.

**product automation**
Automation integrated into the base of SA z/OS for the products CICS, DB2, IMS, TWS (formerly called features).

**program operator interface (POI)**
A NetView facility for receiving VTAM messages.

**program to program interface (PPI)**
A NetView function that allows user programs to send or receive data buffers from other user programs and to send alerts to the NetView hardware monitor from system and application programs.

**protocol**
In SNA, the meanings of, and the sequencing rules for, requests and responses used for managing the network, transferring data, and synchronizing the states of network components.

**proxy resource**
A resource defined like an entry type APL representing a processor operations target system.

**PSM**
See ProcOps Service Machine.

**PU**
See physical unit.

**R**

**RACF**
See Resource Access Control Facility.

**remote system**
A system that receives resource status information from an SA z/OS focal-point system. An SA z/OS remote system is defined as part of the same SA z/OS enterprise as the SA z/OS focal-point system to which it is related.

**requester**
A workstation from that user can log on to a domain from, that is, to the servers belonging to the domain, and use network resources. Users can access the shared resources and use the processing capability of the servers, thus reducing hardware investment.

**resource**
Any facility of the computing system or operating system required by a job or task, and including main storage, input/output devices, the processing unit, data sets, and control or processing programs.

In NetView, any hardware or software that provides function to the network.

In SA z/OS, any z/OS application, z/OS component, job, device, or target system capable of being monitored or automated through SA z/OS.

**Resource Access Control Facility (RACF)**
A program that can provide data security for all your resources. RACF protects data from accidental or deliberate unauthorized disclosure, modification, or destruction.
resource group
A physically partitionable portion of a processor. Also known as a side.

Resource Measurement Facility (RMF)
A feature of z/OS that measures selected areas of system activity and presents the data collected in the format of printed reports, System Management Facility (SMF) records, or display reports.

Resource Object Data Manager (RODM)
In NetView for z/OS, a component that provides an in-memory cache for maintaining real-time data in an address space that is accessible by multiple applications. RODM also allows an application to query an object and receive a rapid response and act on it.

resource token
A unique internal identifier of an ESCON resource or resource number of the object in the IODF.

restart automation
Automation provided by SA z/OS that monitors subsystems to ensure that they are running. If a subsystem fails, SA z/OS attempts to restart it according to the policy in the automation configuration file.

Restructured Extended Executor (REXX)
A general-purpose, high-level, programming language, particularly suitable for EXEC procedures or programs for personal computing, used to write command lists.

return code
A code returned from a program used to influence the issuing of subsequent instructions.

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

RMF See Resource Measurement Facility
RODM See Resource Object Data Manager
S
SAF See Security Authorization Facility

SA IOM
See System Automation for Integrated Operations Management

SAplex
SAplex or "SA z/OS Subplex" is a term used in conjunction with a sysplex. In fact, a SAplex is a subset of a sysplex. However, it can also be a sysplex. For a detailed description, refer to "Using SA z/OS Subplexes" in IBM Tivoli System Automation for z/OS Planning and Installation.

SA z/OS
See System Automation for z/OS

SA z/OS customization dialogs
An ISPF application through which the SA z/OS policy administrator defines policy for individual z/OS systems and builds automation control data and RODM load function files.

SA z/OS customization focal point system
See focal point system

SA z/OS data model
The set of objects, classes and entity relationships necessary to support the function of SA z/OS and the NetView automation platform.

SA z/OS enterprise
The group of systems and resources defined in the customization dialogs under one enterprise name. An SA z/OS enterprise consists of connected z/OS systems running SA z/OS.

SA z/OS focal point system
See focal point system

SA z/OS policy
The description of the systems and resources that make up an SA z/OS enterprise, together with their monitoring and automation definitions.

SA z/OS policy administrator
The member of the operations staff who is responsible for defining SA z/OS policy.

SA z/OS satellite
If you are running two NetViews on 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 panel.

**screen handler**
In SA z/OS, software that interprets all data to and from a full-screen image of a target system. The interpretation depends on the format of the data on the full-screen image. Every processor and operating system has its own format for the full-screen image. A screen handler controls one PS/2 connection to a target system.

**SDF**
See status display facility.

**SDLCC**
See synchronous data link control.

**SDSF**
See System Display and Search Facility.

**secondary system**
A system is a secondary system for an application if it is defined to automation on that system, but the application is not normally meant to be running there. Secondary systems are systems to which an application can be moved in the event that one or more of its primary systems are unavailable. SA z/OS does not start the application on its secondary systems.

**Security Authorization Facility (SAF)**
An MVS interface with which programs can communicate with an external security manager, such as RACF.

**server**
A server is a workstation that shares resources, which include directories, printers, serial devices, and computing powers.

**service language command (SLC)**
The line-oriented command language of processor controllers or service processors.

**service period**
Service periods allow the users to schedule the availability of applications. A service period is a set of time intervals (service windows), during which an application should be active.

**service processor (SVP)**
The name given to a processor controller on smaller System/370 processors.

**service threshold**
An SA z/OS policy setting that determines when to notify the operator of deteriorating service for a resource. See also alert threshold and warning threshold.

**session**
In SNA, a logical connection between two network addressable units (NAUs) that can be activated, tailored to provide various protocols, and deactivated, as requested. Each session is uniquely identified in a transmission header by a pair of network addresses identifying the origin and destination NAUs of any transmissions exchanged during the session.

**session monitor**
The component of the NetView program that collects and correlates session-related data and provides online access to this information. The successor to NLDM.

**shutdown automation**
SA z/OS-provided automation that manages the shutdown process for subsystems by issuing shutdown commands and responding to prompts for additional information.

**side**
A part of a partitionable CPC that can run as a physical partition and is typically referred to as the A-side or the B-side.

**Simple Network Management Protocol (SNMP)**
A set of protocols for monitoring systems and devices in complex networks. Information about managed devices is defined and stored in a Management Information Base (MIB).

**single image**
A processor system capable of being physically partitioned that has not been physically partitioned. Single-image systems can be target hardware processors.
single-MVS environment
An environment that supports one MVS image. See also MVS image.

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 See service language command

SMP/E See System Modification Program/Extended

SNA See Systems Network Architecture

SNA network
In SNA, the part of a user-application network that conforms to the formats and protocols of systems network architecture. It enables reliable transfer of data among end users and provides protocols for controlling the resources of various network configurations. The SNA network consists of network addressable units (NAUs), boundary function components, and the path control network.

SNMP See Simple Network Management Protocol

solicited message
An SA z/OS message that directly responds to a command. Contrast with unsolicited message

SSCP See system services control point

SSI See subsystem interface

start automation
SA z/OS-provided automation that manages and completes the startup process for subsystems. During this process, SA z/OS replies to prompts for additional information, ensures that the startup process completes within specified time limits, notifies the operator of problems, if necessary, and brings subsystems to an UP (or ready) state.

startup
The point in time that a subsystem or application is started.

status
The measure of the condition or availability of the resource.

status display facility (SDF)
The system operations part of SA z/OS that displays status of resources such as applications, gateways, and write-to-operator messages (WTORs) on dynamic color-coded panels. SDF shows spool usage problems and resource data from multiple systems.

status focal-point system
See focal point system

steady state automation
The routine monitoring, both for presence and performance, of subsystems, applications, volumes and systems. Steady state automation may respond to messages, performance exceptions and discrepancies between its model of the system and reality.

structure
A construct used by z/OS to map and manage storage on a coupling facility.

subgroup
A named set of systems. A subgroup is part of an SA z/OS enterprise definition and is used for monitoring purposes.

SubGroup entry
A construct, created with the customization dialogs, used to represent and contain policy for a subgroup.

subplex See SAplex

subsystem
A secondary or subordinate system, usually capable of operating independently of, or asynchronously with, a controlling system.

In SA z/OS, an z/OS application or subsystem defined to SA z/OS.

subsystem interface (SSI)
The z/OS interface over which all messages sent to the z/OS console are broadcast.

support element
A hardware unit that provides communications, monitoring, and diagnostic functions to a central processor complex (CPC).

support processor
Another name given to a processor
controller on smaller System/370 processors. See service processor.

SVP See service processor.

**switch identifier**
The switch device number (swchdevn), the logical switch number (LSN) and the switch name.

**switches**
ESCON directors are electronic units with ports that dynamically switch to route data to I/O devices. The switches are controlled by I/O operations commands that you enter on a workstation.

**symbolic destination name (SDN)**
Used locally at the workstation to relate to the VTAM application name.

**synchronous data link control (SDLC)**
A discipline for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or nonswitched links. The configuration of the link connection may be point-to-point, multipoint, or loop. SDLC conforms to subsets of the Advanced Data Communication Control Procedures (ADCCP) of the American National Standards Institute and High-Level Data Link Control (HDLC) of the International Standards Organization.

**SYSINFO Report**
An RMF report that presents an overview of the system, its workload, and the total number of jobs using resources or delayed for resources.

**SysOps**
See system operations.

**sysplex**
A set of z/OS systems communicating and cooperating with each other through certain multisystem hardware components (coupling devices and timers) and software services (couple data sets).

In a sysplex, z/OS provides the coupling services that handle the messages, data, and status for the parts of a multisystem application that has its workload spread across two or more of the connected processors, sysplex timers, coupling facilities, and couple data sets (which contains policy and states for automation).

A Parallel Sysplex is a sysplex that includes a coupling facility.

**sysplex application group**
A sysplex application group is a grouping of applications that can run on any system in a sysplex.

**sysplex couple data set**
A couple data set that contains sysplex-wide data about systems, groups, and members that use XCF services. All z/OS systems in a sysplex must have connectivity to the sysplex couple data set. See also couple data set.

**Sysplex Timer**
An IBM unit that synchronizes the time-of-day (TOD) clocks in multiple processors or processor sides. External Time Reference (ETR) is the z/OS generic name for the IBM Sysplex Timer (9037).

**system**
In SA z/OS, system means a focal point system (z/OS) or a target system (MVS, VM, VSE, LINUX, or CF).

**System Automation for Integrated Operations Management**
An outboard automation solution for secure remote access to mainframe/distributed systems. Tivoli System Automation for Integrated Operations Management, previously Tivoli AF/REMOTE, allows users to manage mainframe and distributed systems from any location.

The full name for SA IOM.

**System Automation for OS/390**
The full name for SA OS/390, the predecessor to System Automation for z/OS.

**System Automation for z/OS**
The full name for SA z/OS.

**system console**
A console, usually having a keyboard and a display screen, that is used by an operator to control and communicate with a system.

A logical device used for the operation and control of hardware functions (for
example, IPL, alter/display, and reconfiguration). The system console can be assigned to any of the physical displays attached to a processor controller or support processor.

In SA z/OS, the hardware system console for processor controllers or service processors of processors connected using SA z/OS. In the SA z/OS operator commands and configuration dialogs, SC is used to designate the system console for a target hardware processor.

**System Display and Search Facility (SDSF)**

An IBM licensed program that provides information about jobs, queues, and printers running under JES2 on a series of panels. Under SA z/OS you can select SDSF from a pull-down menu to see the resources' status, view the z/OS system log, see WTOR messages, and see active jobs on the system.

**System Entry**

A construct, created with the customization dialogs, used to represent and contain policy for a system.

**System Modification Program/Extended (SMP/E)**

An IBM licensed program that facilitates the process of installing and servicing an z/OS system.

**System Operations**

The part of SA z/OS that monitors and controls system operations applications and subsystems such as NetView, SDSF, JES, RMF, TSO, RODM, ACF/VTAM, CICS, IMS, and OPC. Also known as SysOps.

**System Services Control Point (SSCP)**

In SNA, the focal point within an SNA network for managing the configuration, coordinating network operator and problem determination requests, and providing directory support and other session services for end users of the network. Multiple SSCPs, cooperating as peers, can divide the network into domains of control, with each SSCP having a hierarchical control relationship to the physical units and logical units within its domain.

**System/390 Microprocessor Cluster**

A configuration that consists of central processor complexes (CPCs) and may have one or more integrated coupling facilities.

**Systems Network Architecture (SNA)**

The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks.

**T**

**TAF** See [terminal access facility](#).

**Target** A processor or system monitored and controlled by a focal-point system.

**Target Control Task**

In SA z/OS, target control tasks process commands and send data to target systems and workstations through communications tasks. A target control task (a NetView autotask) is assigned to a target system when the target system is initialized.

**Target Hardware**

In SA z/OS, the physical hardware on which a target system runs. It can be a single-image or physically partitioned processor. Contrast with [target system](#).

**Target System**

In a distributed system environment, a system that is monitored and controlled by the focal-point system. Multiple target systems can be controlled by a single focal-point system.

In SA z/OS, a computer system attached to the focal-point system for monitoring and control. The definition of a target system includes how remote sessions are established, what hardware is used, and what operating system is used.

**Task** A basic unit of work to be accomplished by a computer.

In the NetView environment, an operator station task (logged-on operator), automation operator (autotask), application task, or user task. A NetView task performs work in the NetView environment. All SA z/OS tasks are [NetView tasks](#). See also [message monitor task](#) and [target control task](#).
telecommunication line
Any physical medium, such as a wire or microwave beam, that is used to transmit data.

terminal access facility (TAF)
A NetView function that allows you to log onto multiple applications either on your system or other systems. You can define TAF sessions in the SA z/OS customization panels so you don't have to set them up each time you want to use them.

In NetView, a facility that allows a network operator to control a number of subsystems. In a full-screen or operator control session, operators can control any combination of subsystems simultaneously.

terminal emulation
The capability of a microcomputer or personal computer to operate as if it were a particular type of terminal linked to a processing unit to access data.

threshold
A value that determines the point at which SA z/OS automation performs a predefined action. See alert threshold, warning threshold, and error threshold.

time of day (TOD)
Typically refers to the time-of-day clock.

Time Sharing Option (TSO)
An optional configuration of the operating system that provides conversational timesharing from remote stations. It is an interactive service on z/OS, MVS/ESA, and MVS/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 instruction that issues a command or command processor (list of commands) at a specified time or time interval.

Tivoli Workload Scheduler (TWS)
A family of IBM licensed products that plan, execute and track jobs on several platforms and environments. The successor to OPC/A.

TOD
Time of day.

token ring
A network with a ring topology that passes tokens from one attaching device to another; for example, the IBM Token-Ring Network product.

TP
See transaction program.

transaction program
In the VTAM program, a program that performs services related to the processing of a transaction. One or more transaction programs may operate within a VTAM application program that is using the VTAM application program interface (API). In that situation, the transaction program would request services from the applications program using protocols defined by that application program. The application program, in turn, could request services from the VTAM program by issuing the APPCCMD macro instruction.

transitional automation
The actions involved in starting and stopping subsystems and applications that have been defined to SA z/OS. This can include issuing commands and responding to messages.

translating host
Role played by a host that turns a resource number into a token during a unification process.

trigger
Triggers, in combination with events and service periods, are used to control the starting and stopping of applications in a single system or a parallel sysplex.

TSO
See Time Sharing Option.

TSO console
From this 3270-type console you are logged onto TSO or ISPF to use the runtime panels for I/O operations and SA z/OS customization panels.

TWS
See Tivoli Workload Scheduler.

U
unit control block (UCB)
A control block in common storage that describes the characteristics of a particular I/O device on the operating system and that is used for allocating devices and controlling I/O operations.

unsolicited message
An SA z/OS message that is not a direct response to a command.

uniform resource identifier (URI)
A uniform resource identifier is a string of characters used to identify a name of a web resource. Such identification enables interaction with representations of the web resource over the internet, using specific protocols.

user task
An application of the NetView program defined in a NetView TASK definition statement.

Using An RMF Monitor III definition. Jobs getting service from hardware resources (processors or devices) are using these resources. The use of a resource by an address space can vary from 0% to 100% where 0% indicates no use during a Range period, and 100% indicates that the address space was found using the resource in every sample during that period.

Virtual Server Image
A package containing metadata that describes the system requirements, virtual storage drives, and any goals and constraints for the virtual machine (for example, isolation and availability). The Open Virtual Machine Format (OVF) is a Distributed Management Task Force (DMTF) standard that describes a packaging format for virtual server images.

Virtual Server Image Capture
The ability to store metadata and disk images of an existing virtual server. The metadata describes the virtual server storage, network needs, goals and constraints. The captured information is stored as a virtual server image that can be referenced and used to create and deploy other similar images.

Virtual Server Image Clone
The ability to create an identical copy (clone) of a virtual server image that can be used to create a new similar virtual server.

Virtual Storage Extended (VSE)
A system that consists of a basic operating system (VSE/Advanced Functions), and any IBM supplied and user-written programs required to meet the data processing needs of a user. VSE and the hardware that it controls form a complete computing system. Its current version is called VSE/ESA.

Virtual Telecommunications Access Method (VTAM)
An IBM licensed program that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability. Its full name is Advanced Communications Function for the Virtual Telecommunications Access Method. Synonymous with ACF/VTAM.

VM Second Level Systems Support
With this function, Processor Operations is able to control VM second level
systems (VM guest systems) in the same way that it controls systems running on real hardware.

**VM/ESA**
Virtual Machine/Enterprise Systems Architecture. Its current version is called **z/VM**.

**volume**
A direct access storage device (DASD) volume or a tape volume that serves a system in an SA **z/OS enterprise**.

**VSE**
See **Virtual Storage Extended**.

**VTAM**
See **Virtual Telecommunications Access Method**.

**W**

**warning threshold**
An application or volume service value that determines the level at which SA **z/OS** changes the associated icon in the graphical interface to the warning color. See **alert threshold**.

**workstation**
In SA **z/OS** workstation means the **graphic workstation** that an operator uses for day-to-day operations.

**write-to-operator (WTO)**
A request to send a message to an operator at the z/OS operator console. This request is made by an application and is handled by the WTO processor, which is part of the z/OS supervisor program.

**write-to-operator-with-reply (WTOR)**
A request to send a message to an operator at the z/OS operator console that requires a response from the operator. This request is made by an application and is handled by the WTO processor, which is part of the z/OS supervisor program.

**WTO**
See **write-to-operator**.

**WTOR**
See **write-to-operator-with-reply**.

**WWV**
The US National Institute of Standards and Technology (NIST) radio station that provides standard time information. A second station, known as WWVB, provides standard time information at a different frequency.

**X**

**XCF**
See **cross-system coupling facility**.

**XCF couple data set**
The name for the sysplex couple data set prior to MVS/ESA System Product Version 5 Release 1. See also **sysplex couple data set**.

**XCF group**
A set of related members that a multisystem application defines to XCF. A member is a specific function, or instance, of the application. A member resides on one system and can communicate with other members of the same group across the sysplex.

**XRF**
See **extended recovery facility**.

**Z**

**z/OS**
An IBM mainframe operating system that uses 64-bit real storage. See also **Base Control Program**.

**z/OS component**
A part of **z/OS** that performs a specific **z/OS** function. In SA **z/OS**, component refers to entities that are managed by SA **z/OS automation**.

**z/OS subsystem**
Software products that augment the **z/OS** operating system. JES and TSO/E are examples of **z/OS subsystems**. SA **z/OS** includes automation for some **z/OS subsystems**.

**z/OS system**
A **z/OS** image together with its associated hardware, which collectively are often referred to simply as a system, or **z/OS system**.

**z196**
See **IBM Enterprise 196 (z196)**.

**zAAP**
See **IBM System z Application Assist Processor (zAAP)**.

**zBX**
See **IBM zEnterprise BladeCenter Extension (zBX)**.

**zBX blade**
See **IBM zEnterprise BladeCenter Extension (zBX) blade**.

**zCPC**
The physical collection of main storage, central processors, timers, and channels within a zEnterprise mainframe. Although this collection of hardware resources is part of the larger **zEnterprise central**
processor complex, you can apply energy management policies to zCPC that are different from those that you apply to any attached IBM zEnterprise BladeCenter Extension (zBX) or blades. See also central processor complex.

zIIP  See IBM System z Integrated Information Processor (zIIP)

zEnterprise  See IBM zEnterprise System (zEnterprise)
Index

Special characters
, INGPW command 181

A
access
  APPC 173
  data sets, granting 174
  HOM interface 174
  IBM Tivoli Monitoring products, controlling 179
  IPL information 175
  JES2 spool output data sets 176
  OMEGAMON monitors, controlling 180
  processor hardware functions, controlling 183
  restricting, INGCF 177
  restricting, INGJLM 178
  restricting, INGPLEX 177
  spare Couple Data Sets 175
  spare local page data sets 176
  user-defined Couple Data Sets 176
  XCF utilities 174
access authorization levels for I/O operations RACF profiles 189
accessibility xiii
APF
  availability demands 35
  connections 37
  alert filtering 88
alert handler, user-defined, and alert notification
  enabling 121
  introducing 29
  sample alert handler 121
alert notification
  configure 118
  configuring global initialization file 120
  configuring NetView message adapter service 120
  enabling via EIF events 119
  enabling via SA IOM peer-to-peer protocol 119
  enabling via user-defined alert handler 121
  enabling via XML 120
  infrastructure 27
  installation considerations 27
  integration with EIF events 28
  integration with SA IOM 28
  integration with trouble ticket 28
  integration with user-defined alert handler 29
  introduction 27
  starting event/automation service 119
  alert notification for IBM SA z/OS configuring 58
alerts
  NPDA setup 88
  allocation requirements
    REXX environments 20
  ALLOCOUT automation manager startup procedure 79
  alternate focal point 35
  alternate focal point for HTTP connections 35
  alternate focal point for SNMP connections 35
  ANCHOR statement 216
  AOPCOM sample 130
  AOFIN 112
  AOFINIT 129
  AOFIPBD DD statement 113
  AOFMSGST 138
  AOFMSGSY 23, 91
  AOFOPFGW 94
  AOPRINT DD statement 112
  AOFSTAT NetView startup procedure 86
  AOFSTAT NetView startup procedure 78
  AOFTREE 129
  AOFSTS 132
  AOFUT2 DD names 112
  AOFxxxx DD names 112
  APF authorization
    IEAAPFx member 131
    API enabling for SE, 2.9 and later 98
APPC
  access 173
  APPN definitions for VTAM 126
  ARM instrumentation of the automation manager 107
  authorization of started procedures 162
  AUTO1 sample automation operator 23
  AUTO2
    sample automation operator 23
    updating NetView style sheet 89
  Automatic Restart Manager 128
  enabling the automation manager for 107
  automation
    automating product startups 130
  automation agent
    communication with automation manager 25
  automation control file 122
  migrating 122
  automation manager
    communication with automation agent 25
    considerations 23
    initialization 107
    installing 23
    recovery concept 25
    security 108

automation manager (continued)
  startup procedure 86
  storage requirements 24
automation manager configuration
  file 122
  automation manager start procedure 131
  automation manager startup procedure
    ALLOCOUT 79
    CEEDUMP 79
    HSAFCGIN 79
    HSARVR 79
    HSAPLIB 79
    SYSPUT 79
    SYSPRINT 79
    TRACETO 79
    TRACET1 79
  Automation NetView 212
  automation operator AUTO2, update
    NetView style sheet 89
automation policy
  customizing 122
  automation table 173
  autotask operator IDs 143
  autotasks begin 32
  autotasks start 36

B
back-end checking 168
Backup Support Element 35
Base SA z/OS
  configuring 55
  basic mode 16
  BCP internal interface 16
  understanding 18
  BCP internal interface considerations 35
  BINDIR 199
  BLDVIEWS cards 147
  BLDVIEWS statement 217
  BLOCKOMVS parameter 255, 257
  BPXPRMxx member 84
  building the configuration files 68
  Build_TIMEOUT parameter 256

C
CEEDUMP automation manager startup procedure 79
CFGDSN parameter 256
CHPID ports, naming suggestions 44
cloning on z/OS systems 43
CNMCMDU member 89
CNMSTYLE 23
COMM parameter 256
commands
  DISPAUTO 132
  DISPFLGS 132
  DISPSTAT 132
  COMMNDXX 130

© Copyright IBM Corp. 1996, 2014
299
communication established by XCF 25
communication link processor operations 35
communication links HTTP 16
communications links 16
BCP internal interface 16
I/O operations 39
NetView RMTCMD function 16
SNMP 16
TCP/IP 16
compiling SA z/OS REXX Procedures 121
component trace 108
Configuration Assistant preparing 56
configuration of NMC workstation 197
configuration of SA z/OS workstation components 197
configuration options file 58
configuring DSIPARM 88
I/O operations, optional 156
NetView 88
SDF 129
USS Automation 133
configuring SA z/OS 55
connections alternate focal point system 37
focal point system 37
target system 38
connectivity system operations 31
console workplace 2.9 and later
CPC object definitions on the HMC 98
enabling the HMC API 96
enabling the SE API 98
preparing the master HMC 97
preparing the SE 98
setting the community name 96, 98
setting the SE cross partition flags 100
console workplace, identifying 95
control files 122
control unit description 17
control unit ports naming suggestions 44
controlling access to IBM Tivoli Monitoring products 179
to OMEGAMON monitors 180
to processor hardware functions 183
Couple Data Sets spare, access to 175
user-defined, access to 176
coupling facilities description 15
CPC controlling using an HMC, 2.9 and later 96
object definitions on the HMC, 2.9 and later 98
creating a basic PDB 62
cross partition flags setting for SE, 2.9 and later 100
cross-domain definitions for VTAM 124
CU See control unit 17
customization dialog data sets allocating 112
customization of SA z/OS automating product startups 130
configuration of ISPF dialogs 111
SYS1.PARMLIB members 82
VTAM 123
customizing automation policy 122
D DASD description 17
data sets allocating non-shareable 77
granting access to 174
ISPRWRK 112
DD names AOFIN 112
AOFUT2 112
restricted 112
DD statements AOFIPDB 113
AOFPRINT 112
defining APPN for VTAM 126
consoles 149
cross-domain definitions for VTAM 124
TCP/IP host alias 126
VTAM 124
DELAY parameter 256
devices description 17
DHFRPL and the CICS Automation library 150
DFSAS000 exit 151
DIAGDUPMSG INGXINIT parameter 89
DIAGDUMPMSG parameter 257
DIAGINFO parameter 257
dialogs allocate libraries 111
dynamic allocation 112
disability xii
DISPAUTO command 132
DISPFLGS command 132
DISPSTAT command 132
DS66INIT 143
DSICLD
data set name inserted in DD concatenation 138
DSICRTTD 144
DSIMSG 138
DSIOPF 33, 143, 144, 173
DSIPARM 89, 94
configuring 88
customizing 33
data set name inserted in DD concatenation 138
DSIPARM (continued)
DS66INIT 143
DSICRTTD 144
DSIOPF 143, 144
DSIPARM 145
DSICLD
data set name inserted in DD concatenation 138
DSIPARM 145
DSICLD
data set name inserted in DD concatenation 138
E EIF events and alert notification configuring global initialization file 120
configuring NetView message adapter service 120
enabling 119
introducing 28
starting event/automation service 119
EMCS restrictions and limitations 228
setting up 228
Enabling Capacity Change API
Requests 100
Ensemble Hardware Management Console 251
Ensemble HMC communication preparing 101
EQQMLIB library 152
EQQMLIB library 152
ESCON Director naming conventions 44
event/automation service, starting for alert notification 119
extended multiple console support 228
external writer of component trace startup procedure 86
F filtering of NetView alerts 88
firewall information, updating 101
focal point alternate system 35
using services 34
verification of installation 132
focal point DSIPARM 138
focal point startup 138
focal point system 31
alternate 35, 37
connections 37
connections to the target system 38
hardware connections for processor operations 37
front-end checking 168
Function Packages for TSO install function packages 117
functional hardware prerequisites 4
functional prerequisites 5
G gateway sessions 32
ISPF dialog invocation (continued) using TSO logon 115
ISPF dialogs for customization 111
ISPTABL 111
ISPWKR data sets 112
ISQMSG01 94
ISQMSGU1 94

J
JES 85
JES spool output data sets access to 176
JES3INxx 85

K
keyboard xiii

L
LEOPT 257
LIFECYCLE parameter 258
LNKLSTxx load module prefixes in 21 updating 84
local page data sets spare, access to 176
LOCATION statement 216
LOGSTREAM
INGXINIT parameter 89
LOGSTREAM parameter 258
LPALSTxx load module prefixes in 21 updating 83
LPAR mode 16

M
mandatory prerequisites 4
master HMC preparing, 2.9 and later 97
MAXFILEPROC 126
MAXSOCKETS 126
member IEAAPFx 82
SCHEDxx 82
message forwarding path 32
monitoring agent, installing TEP support for 138
monitors, OMEGAMON controlling access to 180
MPFSTSTA 83
MPFSTTxx 83
multiple NetViews 31, 227
MultiSystem Manager 139

N
naming conventions ESCON 44
processor operations 44
NETCONV sessions, NetView style sheet 89
Netcool/OMNibus configuring 204
NetView command authorization for OMEGAMON 180
granting access to data sets 174
Kanji support, update NetView style sheet for 89
running two on one z/OS system 227
security 173
style sheet 23
automation operator AUTO2 89
NETCONV sessions 89
resource discovery 89
tower statements 89
NetView alerts 88
NetView application startup procedure 86
NetView Graphic Monitor Facility Host Subsystem 212
NetView Management Console 13, 17 configuration 211
NetView message adapter service, configuring for alert notification 120
NetView RMTCMD function 16
NetView startup procedure AOFSTAT 78
HSAIPL 78
INGDUMP 78
NetView stylesheet 142
NetView subsystem interface startup procedure 86
NetView to NetView 16
Network Security Program (NetSP) 192
Networking NetView 138, 212
Networking NetView DSIPARM 138
Networking NetView startup 138
NMC 13, 17, 197 configuration 211
preparing the focal point system 139
NMC client 197
installation 201
NMC Server 197
installation 199
NMC workstation installation preparation 212
NMC workstation configuration 197
non-shareable data sets allocating 77
NPDA setup for alerts 88
NUMQTHDS parameter 259

O
OMEGAMON password management 181
security, NetView command authorization 180
OMEGAMON monitors controlling access to 180
OMVS segment 24
operating systems supported operating systems 6
operator definition file 173
operator terminals 6
OPTION statement 218
OS/390 Automatic Restart Manager 128
OVRDELETEDELAY parameter 259

P
PAM 25
Parallel Sysplex description 15
partitioning logical 16
Password Data Store 80
password management
OMEGAMON 181
peer-to-peer protocol, SA IOM enabling 119
introducing 28
physical path completion 37
planning considerations, REXX 20
considerations, z/OS 21
hardware interfaces 18
processor operations connections 37
planning installation 13
policy database 62
policy databases, converting 122
PPIBQOL INGXINIT parameter 89
PREF parameter 259
prefixes
load module 21
members 21
REXX parts 21
preparing Ensemble HMC communication 101
Hardware Management Console, console workplace 2.9 and later 96
hardware, the 95
Support Element, console workplace 2.9 and later 98
prerequisites functional 5
functional hardware 4
mandatory 4
primary automation manager 25
primary focal point 143
processor hardware functions controlling access to 183
processor operations 13
adding to the ISPF menu 114
adding to the ISPF startup procedure 112
BCP internal interface, understanding 18
configure NetView 94
connections, planning 37
control file 81
HTTP Interface, understanding 18
naming conventions 44, 47
SNMP interface, understanding 19
TCP/IP interface, understanding 20
processor operations communication link 35
ProcOps 3, 13
PROCOPS statement 216
profile, RACF, for I/O operations 188
Program List Table Definitions 149
RACF profile
access authorization levels 189
assigning authorization 188
assigning authorization by access level 188
for I/O operations 188
TCP/IP port authorization 189
using specific profiles 189
reconvert I/O operations panels 116
recovery
performed by XCF 25
takeover file 25
recovery scenarios 26
recovery task 36
relational data services 148
Required Control Region Parameters
specifying 151
requirements
hardware 3
software 4
resource discovery, NetView style sheet 89
restart Automatic Restart Manager
enabled subsystems 128
restricting access
INGCF 177
INGJLM 178
INGPLEX 177
restrictions to z/OS system names 43
REXX environments, allocation
requirements 20
planning considerations 20
procedures, compilation 121
REXX environments 116
RMTCMD 16
RMTCMD security 128
RODM
access information 173
authorization 227
RODMVIEW 227
running two NetViews on one z/OS system 227
SA z/OS satellite
installation verification 228
SA z/OS satellite installation 138
SA z/OS topology manager 139
configure INGTOPOF file 146
configure RODM 145
DSIPARM,DSIOPF 143, 144
operator profiles 144
SA z/OS topology manager, modify DSIPARM for 142
SAF-based security product 173
safe switching 39
SAM 25
sample
AOFCOM 130
INGEJES3 85
INGEMPF 83
INGEREXC 121
INGEREXR 121
INGSCHE 82
sample alert handler for alert notification 121
sample INGTOPOF file 224
sample library
SINGSAMP 51
sample user exits 147
satellite 227
installation verification 228
satellite installation 138
SCHEDx member 82
SDF, configuring 129
SDFROOT 129
SE
API, enabling for 2.9 and later 98
community name, setting for 2.9 and later 98
cross partition flags, setting for 2.9 and later 100
preparing, console workplace 2.9 and later 98
secondary automation manager 25
secondary focal point 143
security 161
back-end checking 168
commands 165
focal point system and target system 173
front-end checking 168
OMEGAMON, NetView command authorization 180
operators 165
roles 163
stylesheet options 172
use of commands cross system 166
use of commands from TSO or Batch 167
security considerations 108
security definition 128
SETTIMER 89
setting up
 Ensemble Hardware Management Console 251
setup (NPDA) for alerts 88
shortcut keys xiii
SINGNPRM 89
SINGSAMP 51, 138
HSADEFA 107
SINGSAMP (continued)
HSAPRM00 107
INGEAMSA 131
.sample exits 147
SIT or startup overrides 148
SMFPRMxx member 85
SMP/E 51
SNMP 16
SNMP interface
understanding 19
SOAP requests 179
software requirements 4
SOMAXCONN 126
spare Couple Data Sets
access to 175
spare local page data sets
access to 176
specifying
 Required Control Region Parameters 151
SRFILTER 88
SSI startup procedure 86
START parameter 259
start SA z/OS for the first time 59
starting the customization dialog 61
startup
automation manager 131
system operations 130
startup procedure
automation manager 86
starting procedure, ISPF
adding processor operations to 112
status display facility 129
STC-user
granting access to data sets 174
STEPLIB
data set name inserted in DD concatenation 138
STOPDELAY parameter 260
storage requirements
automation manager 24
style sheet, NetView 89
subplex
requirements for 21
using 21
subsystem interface startup procedure 86
Support Element
API, enabling for 2.9 and later 98
community name, setting for 2.9 and later 98
cross partition flags, setting for 2.9 and later 100
preparing, console workplace 2.9 and later 98
supported hardware 5
operator terminals 6
supported operating systems 6
switch ports, reasons for naming 44
syntax
HSAPRM00 255
INGTOPOF 215
SYSL1.NUCLEUS 82
SYSL1.PARMLIB
customization of members 82
SYSL1.PARMLIB member
configuring 58