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

Editions
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 or technical newsletters.

This edition replaces SC34-2647-00.

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
- vii

## Tables
- ix

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

## Dotted decimal syntax diagrams
- xiii

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

## Notices
- xvii
  - Website Disclaimer
  - xvii
  - Programming Interface Information
  - xvii
  - Trademarks
  - xviii

## About This Book
- xix
  - Who Should Use This Book
  - xix
  - Where to Find More Information
  - xix
    - The System Automation for z/OS Library
    - xix
    - Related Product Information
    - xx
    - Related Online Information
    - xx
    - Summary of Changes for SC34-2718-00
    - xx
      - New Information
      - xx

## Part 1. Introduction and Concepts

### Chapter 1. Introducing SA z/OS
- 3
  - Overview
  - 3
  - Component Description
  - 4
    - System Operations
    - 4
    - Processor Operations
    - 4
    - I/O Operations
    - 4
  - Hardware Overview
  - 4
    - SA z/OS Resources
    - 6
    - Channels
    - 8
    - Processors and Systems
    - 10
    - I/O Devices
    - 10

### Chapter 2. Concepts
- 11
  - Automation Concepts
  - 11
    - Overview of Automation Logic
    - 11
    - How Requests Become Goals
    - 17
    - How Request Propagation Works
    - 17
    - Relationship Support
    - 18
    - How the Automation Manager and Automation Agents Cooperate in a Sysplex
    - 21
    - Locked Votes and Requests
    - 22
    - Goal Driven Automation
    - 22
    - Automatic Calculation of Schedules (Service Periods)
    - 24
    - Event and Trigger Support
    - 24
    - Error Thresholds
    - 26
    - Groups
    - 27
      - Group Types
      - 27
      - Relationships
      - 28
      - Runmodes
      - 28
      - Operating Groups
      - 28
      - Moving MOVE Groups
      - 28
    - Controlling Application Move and Server Management
      - 28
      - Move Functions
      - 29
      - Controlling Application Move with Preference Values
      - 30
      - SA z/OS Automation Flags
      - 39
      - Automation Agent Flags
      - 40
      - The Automation Manager Global Automation Flag
      - 41
      - Setting Automation Flags
      - 42
      - Relationship Between Automation Flags and Resource Status
      - 42
      - Automation Flags and Minor Resource Flags
      - 43
      - Statuses Supplied by the Automation Manager
        - Observed Status
        - 44
        - Automation Status
        - 46
        - Desired Status
        - 46
        - Startability Status
        - 47
        - Compound Status
        - 47
        - Health Status
        - 48
      - Application Statuses Supplied by Automation Agents
        - Application Statuses Supplied by Automation Agents
        - 48
        - Automation Agent Statuses
        - 48
        - Application Monitor Status
        - 56
        - Automatic Restart Manager Statuses
        - 56
        - Mapping the Automation Agent Statuses to the Automation Manager Observed Status
        - 57
        - Monitor Statuses Supplied by Automation Agents
        - 59
        - SA z/OS Processes Involved in Status Determination
        - 61
      - Using SA z/OS for Monitoring
        - Resynchronization
        - 62
      - Inform List concepts
        - 63
      - Inform SDF, NMC
        - 64
      - Inform ITM
        - 64
      - Inform IOM, EIF, USR, TTT
        - 64
      - Inform SMF
        - 65
      - Automated Function Names/OPERATORS Policy Item
        - 66
      - Structure of AOFWRKxx Operators
        - 68

### Chapter 3. Starting and Stopping
- 69
  - SA z/OS
  - 69
    - Starting, Stopping, and Maintaining the Automation Manager
    - 69
    - Starting the Automation Manager: START (S)
    - 69
### Chapter 4. Refreshing Automation Policy

- **Other Ways to Refresh Automation Policy**
  - Refreshing an Automation Manager Configuration
  - Identifying Current Configuration Files
  - For an Automation Agent
  - For the Primary Automation Manager

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

<table>
<thead>
<tr>
<th>Chapter 5. Getting Started</th>
<th>81</th>
</tr>
</thead>
<tbody>
<tr>
<td>Issuing Commands</td>
<td>81</td>
</tr>
<tr>
<td>Using Command Dialogs</td>
<td>82</td>
</tr>
<tr>
<td>Command Logging</td>
<td>82</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 6. How to Get Resource Information Sysplexwide</th>
<th>85</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using INGLIST to View Resources</td>
<td>85</td>
</tr>
<tr>
<td>Specifying Filters with INGFILT</td>
<td>86</td>
</tr>
<tr>
<td>Using INGINFO to View a Selected Resource</td>
<td>87</td>
</tr>
<tr>
<td>Displaying Application Group Information</td>
<td>87</td>
</tr>
<tr>
<td>Displaying Monitor Information</td>
<td>87</td>
</tr>
<tr>
<td>Displaying Relationship Information with INGRELS</td>
<td>87</td>
</tr>
<tr>
<td>Displaying the Automation Agent View with DISPSTAT</td>
<td>88</td>
</tr>
<tr>
<td>Specifying Filters with DISPSFLT</td>
<td>89</td>
</tr>
<tr>
<td>Using DISPINFO to Display Detailed Information</td>
<td>90</td>
</tr>
<tr>
<td>Using DISPTREE to Display Dependency Information</td>
<td>90</td>
</tr>
<tr>
<td>Displaying Statistical Information</td>
<td>91</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 7. How to Monitor and Control Resources</th>
<th>93</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting and Stopping Resources</td>
<td>94</td>
</tr>
<tr>
<td>Using INGREQ</td>
<td>94</td>
</tr>
<tr>
<td>Using INGVOTE</td>
<td>95</td>
</tr>
<tr>
<td>Using INGHIST</td>
<td>96</td>
</tr>
<tr>
<td>Application Pacing</td>
<td>96</td>
</tr>
<tr>
<td>Monitoring for IPL Completion</td>
<td>98</td>
</tr>
<tr>
<td>Using Runmodes</td>
<td>99</td>
</tr>
<tr>
<td>Defining Runmodes and Runtokens</td>
<td>100</td>
</tr>
<tr>
<td>Assigning Runtokens</td>
<td>101</td>
</tr>
<tr>
<td>Displaying Valid Runmodes</td>
<td>101</td>
</tr>
<tr>
<td>Displaying Runtokens</td>
<td>101</td>
</tr>
<tr>
<td>Displaying Runtoken Qualifications</td>
<td>102</td>
</tr>
<tr>
<td>Displaying Runmode Requests or Votes</td>
<td>103</td>
</tr>
<tr>
<td>Setting Runmodes at SA z/OS Initialization</td>
<td>104</td>
</tr>
<tr>
<td>Time</td>
<td>104</td>
</tr>
<tr>
<td>Setting Runmodes or Forcing a Runmode Qualification using INGRUN</td>
<td>104</td>
</tr>
<tr>
<td>Using UP Status Delay</td>
<td>105</td>
</tr>
<tr>
<td>Using Schedules</td>
<td>105</td>
</tr>
<tr>
<td>Schedule Override</td>
<td>106</td>
</tr>
<tr>
<td>Resource Override</td>
<td>106</td>
</tr>
<tr>
<td>Displaying and Setting Trigger Information</td>
<td>107</td>
</tr>
<tr>
<td>Displaying and Setting Events</td>
<td>107</td>
</tr>
<tr>
<td>Determining Automation Agent Flag Values</td>
<td>108</td>
</tr>
<tr>
<td><strong>Displaying Actual and Effective Automation Agent Flags</strong></td>
<td>109</td>
</tr>
<tr>
<td>Displaying Automation Agent Flag Override Settings</td>
<td>109</td>
</tr>
<tr>
<td>Setting Automation On or Off: INGAUTO</td>
<td>109</td>
</tr>
<tr>
<td>Working with Application Groups</td>
<td>110</td>
</tr>
<tr>
<td>Using INGGROUP</td>
<td>111</td>
</tr>
<tr>
<td>Moving Sysplex Application Groups</td>
<td>114</td>
</tr>
<tr>
<td>Setting Timers</td>
<td>115</td>
</tr>
<tr>
<td>Interacting with the Automation Manager</td>
<td>115</td>
</tr>
<tr>
<td>Changing the Automation Agent Status</td>
<td>116</td>
</tr>
<tr>
<td>Monitoring of Resources</td>
<td>116</td>
</tr>
<tr>
<td>Message Capturing</td>
<td>116</td>
</tr>
<tr>
<td>Monitoring Messages</td>
<td>119</td>
</tr>
<tr>
<td>Monitoring Hardware</td>
<td>122</td>
</tr>
<tr>
<td>Monitoring Health Performance</td>
<td>123</td>
</tr>
<tr>
<td>Monitoring OMEGAMON Sessions</td>
<td>125</td>
</tr>
<tr>
<td>Monitoring Processor Status</td>
<td>128</td>
</tr>
<tr>
<td>Monitoring with the Status Display Facility</td>
<td>131</td>
</tr>
<tr>
<td>Setting Up SDF</td>
<td>132</td>
</tr>
<tr>
<td>Interpreting Colors in SDF and DISPSTAT</td>
<td>133</td>
</tr>
<tr>
<td>How SDF Works</td>
<td>133</td>
</tr>
<tr>
<td>Accessing SDF</td>
<td>134</td>
</tr>
<tr>
<td>Using SDF</td>
<td>134</td>
</tr>
<tr>
<td>How to Check Resource Status Conditions</td>
<td>136</td>
</tr>
<tr>
<td>Deleting SDF Records</td>
<td>137</td>
</tr>
<tr>
<td>Config Refresh Monitoring</td>
<td>138</td>
</tr>
<tr>
<td>Working with the Looping Address Space Monitor</td>
<td>140</td>
</tr>
<tr>
<td>Monitoring and Controlling a Sysplex</td>
<td>142</td>
</tr>
<tr>
<td>How to Work with INGPLEX</td>
<td>142</td>
</tr>
<tr>
<td>How to Work with INGSTR</td>
<td>159</td>
</tr>
<tr>
<td>Monitoring and Controlling Guest Machines and Logical Partitions</td>
<td>160</td>
</tr>
<tr>
<td>VM Second Level Systems Support</td>
<td>160</td>
</tr>
<tr>
<td>LPAR Management Functions</td>
<td>168</td>
</tr>
<tr>
<td>Monitoring and controlling the resources of a Blade Extension (zBX) in a zEnterprise ensemble environment</td>
<td>171</td>
</tr>
<tr>
<td>Preparation Work</td>
<td>171</td>
</tr>
<tr>
<td>zBX Resource Monitoring and Controlling</td>
<td>171</td>
</tr>
<tr>
<td>Using the SA z/OS Discovery Library Adapter</td>
<td>172</td>
</tr>
<tr>
<td>Overview</td>
<td>173</td>
</tr>
<tr>
<td>Using the SA z/OS Discovery Library Adapter</td>
<td>174</td>
</tr>
<tr>
<td>How SA z/OS Uses Error Thresholds</td>
<td>174</td>
</tr>
<tr>
<td>Displaying Threshold Settings and Occurrences</td>
<td>174</td>
</tr>
<tr>
<td>How to Set or Change Error Thresholds</td>
<td>175</td>
</tr>
<tr>
<td>How to Reset Occurrences</td>
<td>176</td>
</tr>
<tr>
<td>Setting Up Thresholds Using the Customization Dialog</td>
<td>176</td>
</tr>
</tbody>
</table>
### Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Handling Jobs That Are Not Controlled by SA z/OS</td>
<td>177</td>
</tr>
<tr>
<td>Communicating with Other Systems</td>
<td>178</td>
</tr>
<tr>
<td>Sending Commands within a Sysplex</td>
<td>178</td>
</tr>
<tr>
<td>Sending Commands to a Linux Target System</td>
<td>180</td>
</tr>
<tr>
<td>Terminal Access Facility Full-Screen Sessions</td>
<td>180</td>
</tr>
<tr>
<td>Gateway Sessions</td>
<td>182</td>
</tr>
<tr>
<td>Chapter 8. Solving Problems with Resources</td>
<td>183</td>
</tr>
<tr>
<td>If Resources Do Not Start</td>
<td>183</td>
</tr>
<tr>
<td>If Resources Do Not Stop</td>
<td>184</td>
</tr>
<tr>
<td>Availability Scenario</td>
<td>184</td>
</tr>
<tr>
<td>Chapter 9. Using I/O Operations</td>
<td>195</td>
</tr>
<tr>
<td>Making Connectivity Changes</td>
<td>196</td>
</tr>
<tr>
<td>Display Commands To Obtain Data About I/O Resources</td>
<td>196</td>
</tr>
<tr>
<td>Connectivity Commands That Affect a Switch</td>
<td>196</td>
</tr>
<tr>
<td>Display and Utility Commands Related to Connectivity Commands</td>
<td>197</td>
</tr>
<tr>
<td>Utility Commands for General Processing</td>
<td>197</td>
</tr>
<tr>
<td>Using the I/O Operations ISPF Dialog</td>
<td>198</td>
</tr>
<tr>
<td>Using the z/OS Operator Console for I/O Operations Commands</td>
<td>198</td>
</tr>
<tr>
<td>Part 3. Using SA z/OS on Workstations</td>
<td>201</td>
</tr>
<tr>
<td>Chapter 10. Using the NetView Management Console for SA z/OS</td>
<td>203</td>
</tr>
<tr>
<td>Introduction</td>
<td>203</td>
</tr>
<tr>
<td>Multiple Focal Point Support</td>
<td>204</td>
</tr>
<tr>
<td>Starting the NMC</td>
<td>204</td>
</tr>
<tr>
<td>Monitoring SA z/OS on the NMC</td>
<td>205</td>
</tr>
<tr>
<td>Regular Resources</td>
<td>205</td>
</tr>
<tr>
<td>Minor Resources, Anchors and Tokens</td>
<td>206</td>
</tr>
<tr>
<td>Properties of SA z/OS Objects on the NMC</td>
<td>209</td>
</tr>
<tr>
<td>Dynamic Behavior of the SA z/OS Topology Manager</td>
<td>211</td>
</tr>
<tr>
<td>SA z/OS Topology Manager Initialization: INGTOPO INIT</td>
<td>211</td>
</tr>
<tr>
<td>System Detection</td>
<td>212</td>
</tr>
<tr>
<td>The Heartbeat Function</td>
<td>212</td>
</tr>
<tr>
<td>Takeover when a System Goes Down</td>
<td>212</td>
</tr>
<tr>
<td>Issuing Commands from the NMC</td>
<td>212</td>
</tr>
<tr>
<td>Direct Commands</td>
<td>212</td>
</tr>
<tr>
<td>Dialog Commands</td>
<td>213</td>
</tr>
<tr>
<td>Interactive Commands</td>
<td>213</td>
</tr>
<tr>
<td>The INGRTCMD Command</td>
<td>215</td>
</tr>
<tr>
<td>Creating Views</td>
<td>216</td>
</tr>
<tr>
<td>Network Views and Exception Views</td>
<td>216</td>
</tr>
<tr>
<td>Names</td>
<td>217</td>
</tr>
<tr>
<td>SA z/OS Topology Manager Commands</td>
<td>219</td>
</tr>
<tr>
<td>Processor Operations on the NMC</td>
<td>219</td>
</tr>
<tr>
<td>Monitoring Sysplex Resources</td>
<td>220</td>
</tr>
<tr>
<td>Problem Management</td>
<td>220</td>
</tr>
<tr>
<td>Exception Views</td>
<td>220</td>
</tr>
<tr>
<td>Marked and Unmarked Resources</td>
<td>221</td>
</tr>
<tr>
<td>Updating Operator Data</td>
<td>221</td>
</tr>
<tr>
<td>Problem Management of Minor Resources</td>
<td>221</td>
</tr>
<tr>
<td>Part 4. Appendixes</td>
<td>223</td>
</tr>
<tr>
<td>Appendix A. Automation Initialization Options</td>
<td>225</td>
</tr>
<tr>
<td>Appendix B. Using the I/O Operations ISPF Dialog</td>
<td>227</td>
</tr>
<tr>
<td>Getting Started</td>
<td>227</td>
</tr>
<tr>
<td>Starting with the I/O Operations ISPF Main Menu</td>
<td>227</td>
</tr>
<tr>
<td>The Modify Menu</td>
<td>229</td>
</tr>
<tr>
<td>A Sample Way to Modify a Switch Matrix</td>
<td>230</td>
</tr>
<tr>
<td>The View Menu</td>
<td>231</td>
</tr>
<tr>
<td>The Options Menu</td>
<td>234</td>
</tr>
<tr>
<td>The Help Menu</td>
<td>235</td>
</tr>
<tr>
<td>Appendix C. Problem Determination</td>
<td>237</td>
</tr>
<tr>
<td>Maintaining System Status during a Status</td>
<td>237</td>
</tr>
<tr>
<td>Debugging and Tracing</td>
<td>237</td>
</tr>
<tr>
<td>Automation Manager State Trace Table</td>
<td>237</td>
</tr>
<tr>
<td>Using Trace Services for the Automation Manager</td>
<td>238</td>
</tr>
<tr>
<td>Using AOCTRACE</td>
<td>239</td>
</tr>
<tr>
<td>How to Use a Log with INGHIST</td>
<td>240</td>
</tr>
<tr>
<td>How to Use the Diagnostic Option of INGAMS</td>
<td>240</td>
</tr>
<tr>
<td>Tracing and Debugging for ProcOps and the BCP Internal Interface</td>
<td>247</td>
</tr>
<tr>
<td>Using Trace Services for I/O Operations</td>
<td>249</td>
</tr>
<tr>
<td>Collecting the available BCP session status messages</td>
<td>251</td>
</tr>
<tr>
<td>Appendix D. Hardware Interfaces: Additional Recovery Information</td>
<td>253</td>
</tr>
<tr>
<td>BCP Internal Interface Recovery Information</td>
<td>253</td>
</tr>
<tr>
<td>BCP Internal Interface Recovery Actions</td>
<td>253</td>
</tr>
<tr>
<td>How to Switch the IP Address for Target Hardware</td>
<td>254</td>
</tr>
<tr>
<td>Used by Processor Operations</td>
<td>254</td>
</tr>
<tr>
<td>How to Terminate BCP Internal Interface</td>
<td>255</td>
</tr>
<tr>
<td>Connections or Clear Hardware Control Blocks after a Connection Failure</td>
<td>255</td>
</tr>
<tr>
<td>Recovery of Processor Operations Target System</td>
<td>255</td>
</tr>
<tr>
<td>Connections in the event of Hardware Management Console Outage</td>
<td>255</td>
</tr>
<tr>
<td>Glossary</td>
<td>257</td>
</tr>
<tr>
<td>Index</td>
<td>287</td>
</tr>
<tr>
<td>Figures</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>1. A Basic Hardware Configuration</td>
<td>5</td>
</tr>
<tr>
<td>2. Relationships and Conditions</td>
<td>18</td>
</tr>
<tr>
<td>3. The HASPARENT Relationship</td>
<td>19</td>
</tr>
<tr>
<td>4. Relationship in a Sysplex</td>
<td>19</td>
</tr>
<tr>
<td>5. Relationship between Applications and Application Groups</td>
<td>20</td>
</tr>
<tr>
<td>6. Conflicting Relationships and Requests</td>
<td>21</td>
</tr>
<tr>
<td>7. Automation Manager and Automation Agents Cooperation in a Sysplex</td>
<td>22</td>
</tr>
<tr>
<td>8. Example of Aggregating Applications into Application Groups</td>
<td>23</td>
</tr>
<tr>
<td>9. How a Trigger Uses Events</td>
<td>25</td>
</tr>
<tr>
<td>10. Triggers, Events, and Conditions</td>
<td>26</td>
</tr>
<tr>
<td>11. Non-Disruptive and Disruptive Restoration</td>
<td>34</td>
</tr>
<tr>
<td>12. Resource Aware Application Move Example Sysplex</td>
<td>38</td>
</tr>
<tr>
<td>13. Resource Aware Application Move</td>
<td>39</td>
</tr>
<tr>
<td>14. Automation Flag Span of Control</td>
<td>40</td>
</tr>
<tr>
<td>15. Observed Status Transitions</td>
<td>45</td>
</tr>
<tr>
<td>16. Automation Status Transitions</td>
<td>46</td>
</tr>
<tr>
<td>17. Transitions for a Normal Subsystem</td>
<td>55</td>
</tr>
<tr>
<td>18. Transitions for an Abending Subsystem</td>
<td>55</td>
</tr>
<tr>
<td>19. DISPSYS Command Dialog</td>
<td>76</td>
</tr>
<tr>
<td>20. INGAMS Command Dialog</td>
<td>77</td>
</tr>
<tr>
<td>21. Entry Type Panel Selection</td>
<td>97</td>
</tr>
<tr>
<td>22. INGPAC Command Dialog Panel</td>
<td>98</td>
</tr>
<tr>
<td>23. DISPSYS output for Runmodes</td>
<td>101</td>
</tr>
<tr>
<td>24. INGINFO output for Runtokens</td>
<td>102</td>
</tr>
<tr>
<td>25. INGLIST output for Runmode qualifications</td>
<td>102</td>
</tr>
<tr>
<td>26. INGVOTE output for Runmode requests or votes</td>
<td>103</td>
</tr>
<tr>
<td>27. Detail display for INGVOTE command results</td>
<td>103</td>
</tr>
<tr>
<td>28. INGRUN command dialog (for SET request type)</td>
<td>104</td>
</tr>
<tr>
<td>29. INGRUN command dialog (for ADD request type)</td>
<td>105</td>
</tr>
<tr>
<td>30. Natures of Application Groups</td>
<td>111</td>
</tr>
<tr>
<td>31. ISQXDST Command on NetView Console</td>
<td>123</td>
</tr>
<tr>
<td>32. ProcOps Target Status Summary Panel</td>
<td>129</td>
</tr>
<tr>
<td>33. Example SDF Hierarchy for KEY1.</td>
<td>134</td>
</tr>
<tr>
<td>34. Example SDF System Summary Panel</td>
<td>135</td>
</tr>
<tr>
<td>35. Example SDF Detail Status Display Panel for HUGC2APL0</td>
<td>136</td>
</tr>
<tr>
<td>36. Delete Confirmation Panel for an SDF Record</td>
<td>138</td>
</tr>
<tr>
<td>37. INGPTOP Panel</td>
<td>139</td>
</tr>
<tr>
<td>38. INGPCFG Panel</td>
<td>139</td>
</tr>
<tr>
<td>40. Processor Operations Status Summary Panel (ISQESUM)</td>
<td>162</td>
</tr>
<tr>
<td>41. Processor Operations Target Hardware</td>
<td>163</td>
</tr>
<tr>
<td>42. Processor Operations Target Hardware Summary Panel (ISQEHARP) for Non-PSMs</td>
<td>163</td>
</tr>
<tr>
<td>43. Processor Operations Target System Summary Panel (ISQETARG)</td>
<td>164</td>
</tr>
<tr>
<td>44. Processor Operations PSMs/Guests Summary Panel (ISQETRGG)</td>
<td>164</td>
</tr>
<tr>
<td>45. Processor Operations PATH Detail Panel (ISQEPMVC)</td>
<td>165</td>
</tr>
<tr>
<td>46. SA z/OS Discovery Library Adapter</td>
<td>173</td>
</tr>
<tr>
<td>47. Sending a Command to a Linux Target System</td>
<td>180</td>
</tr>
<tr>
<td>48. TAF Fullscreen Display Panel</td>
<td>181</td>
</tr>
<tr>
<td>49. Scenario: INGLIST Output</td>
<td>185</td>
</tr>
<tr>
<td>50. Scenario: INGVOTE Output</td>
<td>186</td>
</tr>
<tr>
<td>51. Scenario: Refreshed INGLIST Output</td>
<td>186</td>
</tr>
<tr>
<td>52. Scenario: INGINFO Output</td>
<td>187</td>
</tr>
<tr>
<td>53. Scenario: INGLIST Output for MSCSK Resources</td>
<td>188</td>
</tr>
<tr>
<td>54. Scenario: INGVOTE Output for MSCSK2</td>
<td>188</td>
</tr>
<tr>
<td>55. Scenario: INGCHEC DS for MSCSK2</td>
<td>189</td>
</tr>
<tr>
<td>56. Scenario: Schedule Override for MSCSK2</td>
<td>189</td>
</tr>
<tr>
<td>57. Scenario: Refreshed INGLIST Output</td>
<td>190</td>
</tr>
<tr>
<td>58. Scenario: Checking Automation Flags with INGLIST</td>
<td>190</td>
</tr>
<tr>
<td>59. Scenario: Checking Triggers with INGLIST</td>
<td>191</td>
</tr>
<tr>
<td>60. Scenario: DISPTRG Command Dialog</td>
<td>192</td>
</tr>
<tr>
<td>61. Scenario: Trigger Condition List</td>
<td>192</td>
</tr>
<tr>
<td>62. Scenario: Verifying Automation Status</td>
<td>193</td>
</tr>
<tr>
<td>63. SA z/OS-NMC Communication Overview</td>
<td>204</td>
</tr>
<tr>
<td>64. Tivoli NetView for z/OS Management Console</td>
<td>205</td>
</tr>
<tr>
<td>65. Network View for Regular Resources</td>
<td>206</td>
</tr>
<tr>
<td>66. Subview with Applications of an Application Group</td>
<td>206</td>
</tr>
<tr>
<td>67. Anchor View</td>
<td>208</td>
</tr>
<tr>
<td>68. View with 'Lost&amp;Found'-Anchors</td>
<td>209</td>
</tr>
<tr>
<td>69. Resource Properties Window</td>
<td>211</td>
</tr>
<tr>
<td>70. Parameter Window for INGTHRES</td>
<td>213</td>
</tr>
<tr>
<td>71. NetView Console with INGREQ Panel</td>
<td>214</td>
</tr>
<tr>
<td>72. Actualized View</td>
<td>215</td>
</tr>
<tr>
<td>73. Processor Operations View</td>
<td>219</td>
</tr>
<tr>
<td>74. Exception View</td>
<td>221</td>
</tr>
<tr>
<td>75. Sample AOF767I Message</td>
<td>226</td>
</tr>
<tr>
<td>76. I/O Operations ISPF Main Menu</td>
<td>228</td>
</tr>
<tr>
<td>77. INGAMS Diagnostic Functions Panel</td>
<td>240</td>
</tr>
<tr>
<td>78. INGAMS Command Dialog, Show Details</td>
<td>241</td>
</tr>
<tr>
<td>79. INGAMS Work Item Statistics Display</td>
<td>242</td>
</tr>
<tr>
<td>80. INGAMS Work Item History Display</td>
<td>243</td>
</tr>
</tbody>
</table>
# Tables

1. System Automation for z/OS Library xix  
2. Hierarchy of Requests 14  
3. Preference Values and Behavior of Members 31  
4. Automation Agent Statuses 49  
5. Regular Monitoring: Automation Status Determination 56  
6. Automation Agent to Automation Manager Status Mapping 57  
7. Monitor Statuses 59  
8. Monitor Status Mapping 59  
9. SA z/OS Mapping of NetView Automation Operator to SA z/OS Automated Function 66  
10. Operator Tasks and Related Commands 93  
11. Setting Timers 115  
12. Code Entry Meanings 118  
13. Displaying Threshold Settings and Occurrences 175  
14. SA z/OS Trace Activation Options 238
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 z/OS ISPF User’s Guide Vol 1 for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.
Dotted decimal syntax diagrams

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

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

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

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

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

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

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

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

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

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

Notes:
1. If a dotted decimal number has an asterisk (*) next to it and there is only one item with that dotted decimal number, you can repeat that same item more than once.
2. If a dotted decimal number has an asterisk next to it and several items have that dotted decimal number, you can use more than one item from the list, but you cannot use the items more than once each. In the previous example, you could write HOST STATE, but you could not write HOST HOST.
3. The * symbol is equivalent to a loop-back line in a railroad syntax diagram.

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

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

Use one of the following methods to send us your comments:

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

Include the following information:
- Your name and address
- Your email address
- Your telephone or fax number
- The publication title and order number:
  IBM Tivoli System Automation for z/OS V3R50 User's Guide
  SC34-2718-00
- 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/.
Notices

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

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

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

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

Website Disclaimer

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

Programming Interface Information

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

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

- AIX
- BookManager
- CICS
- DB2
- ESCON
- eServer
- FICON
- GDPS
- IBM
- IMS
- MVS
- NetView
- OMEGAMON
- OS/390
- Parallel Sysplex
- PR/SM
- Processor Resource/Systems Manager
- RACF
- S/390
- Sysplex Timer
- System z
- System/390
- Tivoli
- Tivoli Enterprise Console
- VM/ESA
- VTAM
- z/OS
- z/VM
- z/VSE
- zSeries

The following terms are trademarks of other companies:

- Linux is a registered trademark of Linus Torvalds.
- Microsoft and Windows are registered trademarks of the Microsoft Corporation.
- UNIX is a registered trademark of The Open Group.
About This Book

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

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

Who Should Use This Book

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

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

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

Where to Find More Information

The System Automation for z/OS Library

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

Table 1. 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 and Operator’s Reference Guide</td>
<td>SC34-2749</td>
</tr>
<tr>
<td>IBM Tivoli System Automation for z/OS End-to-End Automation Adapter</td>
<td>SC34-2750</td>
</tr>
<tr>
<td>IBM Tivoli System Automation for z/OS Monitoring Agent Configuration and User’s Guide</td>
<td>SC34-2751</td>
</tr>
</tbody>
</table>
SA z/OS Home Page:
For the latest news on SA z/OS, visit the SA z/OS home page at:

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

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

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

Summary of Changes for SC34-2718-00
This document contains information previously presented in System Automation for z/OS V3R4.0 User’s Guide, SC34-2647-00.

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.

The following sections list important or major changes to this document.

New Information
Monitoring for IPL Completion
A description of the above parameter is given in “Monitoring for IPL Completion” on page 98. This parameter indicates whether the entry types; Application group (APG), Application (APL) or Monitoring resources (MTR) must have a status of AVAILABLE before a notification of a successful IPL can be issued.

Inform List Concepts
A description of how to use the Inform List field is added. The concept of notification for SA z/OS resources is outlined in “Inform List concepts” on page 63 and how external systems are defined to monitor automated resources and receive alerts or notifications.

Message Capturing
The ability to capture messages for SA z/OS monitoring is described in “Message Capturing” on page 116 and how to direct and forward messages for automation monitoring.

Shutdown of Primary Automation Manager system
Additional guidance for “Shutdown of Primary Automation Manager system”
(PAM) system on page 71 is added to Starting, Stopping, and Maintaining the Automation Manager on page 69 when multiple system shutdowns or IPLs are required.

Groups
The use of Groups under System Automation is explained conceptually in a new section, outlining the type of groups available and their relationships.

Using UP Status Delay
The setting Up Status Delay is described in Using UP Status Delay and indicates how a delay can be applied while initializing to allow applications to reach an effective UP status.

Looping Address Space Monitor
The Looping Address Space Suppression procedure is described in Working with the Looping Address Space Monitor on page 140.

SNMPv3 Security Settings
SNMPv3 security settings are added in How to Monitor Communications Path Status on page 131.

Application Pacing
A description of how to use Pacing Gates for application management is provided in Application Pacing on page 96.

Config Refresh Monitoring under SDF
Panel and tree definitions for Config Refresh Monitoring on page 138 are described and added in Monitoring with the Status Display Facility on page 131.
Part 1. Introduction and Concepts

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

- Chapter 1, “Introducing SA z/OS,” on page 3
- Chapter 2, “Concepts,” on page 11
- Chapter 3, “Starting and Stopping SA z/OS,” on page 69
- Chapter 4, “Refreshing Automation Policy,” on page 75
Chapter 1. Introducing SA z/OS

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

Overview

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

Monitor
Monitor your resources to respond before they affect end users:
- Monitor hardware components
- Monitor software products and applications
- Monitor automated processes
- Monitor messages and alerts

Control
Take action to control conditions:
- Start and stop your entire enterprise system; initiate hardware and software startup and shutdown sequences
- Manage both remote and local operations and support any System z® or 390-CMOS zSeries processor within a Parallel Sysplex
- Manage several operating systems: z/OS, OS/390®, MVS™, VM, VSE, and Linux on System z
- Control a coupling facility as a target system with coupling links in a Parallel Sysplex environment
- React to errors and unscheduled events

Automate
Automate many repetitive and complex tasks:
- Start and shut down software resources
- Start and shut down hardware resources
- Control channels and channel paths
- Control availability of I/O devices
- Control switching of I/O device ports
- Detect and respond to system messages
- Perform initial program load (IPL)
- Perform system power-on reset (POR)
- Build automation policy for your enterprise
- Extend the built-in automation routines by writing your own automation policies
Overview

You monitor and control hardware and software resources from a NetView Management Console (NMC) workstation, a NetView console or Tivoli Enterprise Portal (TEP), or monitor them from the Tivoli Enterprise Console® (TEC).

Component Description

System Operations

System operations monitors and controls operating system components and middleware such as CICS®, IMS™, and DB2®.

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

Enterprise monitoring is used by SA z/OS to update the NetView Management Console (NMC) resource status information that is stored in the Resource Object Data Manager (RODM), and update the health status information that is displayed on the Tivoli Enterprise Portal (TEP) via the IBM Tivoli Monitoring infrastructure.

Processor Operations

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

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

I/O Operations

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

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

Hardware Overview

Figure 1 on page 5 illustrates a basic hardware configuration.
It shows a two-processor Parallel Sysplex configuration with systems running on it. One is playing the role of a SA z/OS focal point. For example, the role of the SA z/OS NMC focal point with information about all the systems and applications in the sysplex, running under the control of SA z/OS.

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

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

SA z/OS Resources

This section describes the functions and resources used by SA z/OS and its components.

Focal point processor
In a multisystem environment, the primary (or issuing) host processor has SA z/OS installed. You might think of it as SA z/OS’s command center. This is where you enter commands you want to process. It includes a central processor complex, which is a collection of hardware that consists of central storage, one or more central processors, timers, and channels. NetView and SA z/OS are installed on this processor.

Target processor
The processor that is controlled by a focal point processor. It can be a local or remote processor, or a logical partition of a processor. When you send a command from a focal point system, the target is the processor you want the command to affect.

The operating system can be z/OS, OS/390, VM, VSE, Linux, or a coupling facility. If the operating system is z/OS or Linux, SA z/OS should be installed on the target processor for full SA z/OS functionality. If the operating system is VM or VSE, or if the target processor is a coupling facility, only SA z/OS processor operations functions are available for the target.

OCF-based processor
CMOS-S/390, System z operations command facility (OCF) for interacting with human operators or external programs to perform operations management functions on the central processor. Processor operations uses this facility to manage and control the processors.

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 simplex functions are all parts of a sysplex.

An z/OS sysplex can be viewed and managed as a single system image on the NMC workstation.

Logically Partitioned (LPAR) mode
A processor with the PR/SM™ feature that can be divided into partitions with separate logical system consoles that allocates hardware resources among several logical partitions. (It is called logical because the processor is not physically divided, but divided only by definition.) The partitions are defined, monitored, and activated separately by processor operations.

A processor that does not use logical partitions is in “basic mode”.
Hardware Overview

Images
Processors can operate as single images or partitioned images. A single image allows a multiprocessor system to function as one central processor complex.

A multiprocessor system that functions as two or more independent processors (with separate power, water, and maintenance boundaries) is a physically partitioned configuration image.

Communications links
Paths that connect the focal point system processor to target processors so that commands, messages, and alerts can flow. Two types of communication links that can be selected for communication between the focal point and the target processor hardware: INTERNAL communication via operating system (BCPII) and IP based SNMP connections.

OSA I/O operations can display the open system adapter (OSA) channel logical definition, physical attachment, and status. You can configure an OSA channel on or off.

ESCON/FICON Director
The enterprise systems connection (ESCON) and fiber channel connection (FICON) Directors (also known as switches or dynamic switches) are hardware electronic switching units with ports that dynamically switch to route data to input/output devices. Each port on the switch is an addressable connection, controlled by I/O operations commands. Two switches (one dynamic and one static) can be linked or chained together to increase the path length.

Ports On an ESCON/FICON switch, a port is an addressable connection. The switch routes data through the ports to the channel or control unit. Each port can be given a name that can be entered into a switch matrix, and you can use commands to change the switch configuration.

Control units
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.

Coupling Facilities
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 stand-alone 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.

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.

Sysplex Timer
An IBM unit that synchronizes the time-of-day (TOD) clocks in a
Hardware Overview

multiprocessor or in processor sides. External Time Reference (ETR) is the generic name for the IBM Sysplex Timer (9037).

**NMC Workstation**
A NetView feature running on a personal computer that provides a graphic series of windows controlled by NetView allowing you to monitor the SA z/OS enterprise graphically.

**Automation policy**
Operations and procedures (or policy) information is kept in files that describe routine repetitive procedures, as well as the resources in your enterprise and their relationships. Automation automatically starts, restarts, monitors, or stops resources. You define the policy using the customization dialog. The resulting file is the SA z/OS configuration file.

**Tivoli Enterprise Portal**
Tivoli Enterprise Portal (TEP) is the Java-based interface to the data monitoring and management resources of the IBM Tivoli Monitoring infrastructure. It displays information about the status of automation on z/OS systems and z/OS sysplexes that has been collected by the SA z/OS monitoring agent.

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

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

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

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

CFR/CFS and ISC/ISR
The I/O operations component of SA z/OS can display connectivity data and control connectivity for integrated system channels (ISC) connected to
Hardware Overview

a coupling facility. This includes coupling facility receiver (CFR) channels and coupling facility sender (CFS) channels.

Processors and Systems

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

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

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

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

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

I/O Devices

On a dynamically updated resource display, the status of an I/O device is updated when a message to the operator console or to the z/OS system log is trapped at the target system.
Chapter 2. Concepts

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

- Automation Concepts
- Groups on page 27
- Controlling Application Move and Server Management on page 28
- SA z/OS Automation Flags on page 39
- Statuses Supplied by the Automation Manager on page 43
- Application Statuses Supplied by Automation Agents on page 48
- Using SA z/OS for Monitoring on page 62
- Inform List concepts on page 63

Automation Concepts

This section provides details of the following automation concepts of SA z/OS:

- Overview of Automation Logic
- How Requests Become Goals on page 17
- How Request Propagation Works on page 17
- Relationship Support on page 18
- How the Automation Manager and Automation Agents Cooperate in a Sysplex on page 21
- Locked Votes and Requests on page 22
- Goal Driven Automation on page 22
- Automatic Calculation of Schedules (Service Periods) on page 24
- Event and Trigger Support on page 24
- Automation Operators on page 65

Overview of Automation Logic

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

- The observing, reacting and doing parts are located in the NetView address space and are known as the automation agents. The automation agents are responsible for:
  - Recovery processing
  - Message processing
  - Active monitoring: they propagate status changes to the automation manager

- Within each sysplex, the coordination, decision making and controlling functions are gathered into a single address space outside of NetView. This address space is called the automation manager.

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

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

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

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

The automation manager knows several different statuses for each resource:

- The observed status
- The desired status
- The automation status
- The startability status
- The compound status
- The health status

More detail about these statuses is provided in “Statuses Supplied by the Automation Manager” on page 43.

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

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

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

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

1. **OPERATOR**
   - By the **operator** command INGREQ (see “How Requests Become Goals” on page 17). These requests are labeled with the originator OPERATOR(USERID) in the INGVOTE command dialog.

2. **AUTOOPS**
   - By the INGREQ command issued from automation procedures (REXX exec, CLIST). These requests are labeled with the originator AUTOOPS in the INGVOTE command dialog.
3 E2EMGR
By requests from the end-to-end automation manager. These requests are handled by the SA z/OS end-to-end automation adapter, which maps the requests to SA z/OS commands.

4 EXTERNAL
By TWS Automation. If TWS Automation interfaces with SA z/OS, TWS takes care of the startup of supporting resources required for applications that are scheduled with TWS. These requests are labeled with the originator EXTERNAL in the INGVOTE command dialog.

5 SCHEDULE
There are three different ways that schedules can be used:

Resource override
The INGSCHED operator command with a resource parameter interactively overrides a service period specifically for the specified resource (see “Resource Override” on page 106).

Schedule override
With the INGSCHED operator command, which can be used to pseudo-interactively override service periods for all resources that use this service period (see “Schedule Override” on page 106).

Automation policy
With the automation policy, as defined in the customization dialog. The automation programmer uses service periods for applications or application groups as an easy method to let SA z/OS automatically schedule up and down times for these resources. Thus these service periods are also referred to as schedules.

These requests are labeled with the originator SCHEDULE in the INGVOTE command dialog.

6 A/A
By GDPS Active/Active Controller.

7 Default
By any other source/method, for instance runmode requests originating from INGRUN.

8 Group
When a group propagates a request to one of its members, it increments the Priority by 1000. For nested groups, this can result in multiples of 1000s.

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

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

Table 2 on page 14 shows the hierarchy of all possible requests. The strongest goal that you can specify for a resource is interactively to issue an INGREQ command to make a resource unavailable, such as:

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

The weakest goal you can define for a resource is to make it available with low priority via a service period.
A request to make a resource available is in most cases preferred to a request to make the same resource unavailable.

Table 2. Hierarchy of Requests

<table>
<thead>
<tr>
<th>Rank/Priority</th>
<th>Source</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>03760000 FORCE</td>
<td>1 Operator</td>
<td>Unavailable</td>
</tr>
<tr>
<td>03740000 FORCE</td>
<td>1 Operator</td>
<td>Available</td>
</tr>
<tr>
<td>03680000 FORCE</td>
<td>6 A/A</td>
<td>Available</td>
</tr>
<tr>
<td>03670000 FORCE</td>
<td>6 A/A</td>
<td>Unavailable</td>
</tr>
<tr>
<td>03660000 FORCE</td>
<td>2 AUTOOPS</td>
<td>Unavailable</td>
</tr>
<tr>
<td>03640000 FORCE</td>
<td>2 AUTOOPS</td>
<td>Available</td>
</tr>
<tr>
<td>03560000 FORCE</td>
<td>3 E2EMGR</td>
<td>Unavailable</td>
</tr>
<tr>
<td>03540000 FORCE</td>
<td>3 E2EMGR</td>
<td>Available</td>
</tr>
<tr>
<td>03460000 FORCE</td>
<td>4 EXTERNAL</td>
<td>Unavailable</td>
</tr>
<tr>
<td>03440000 FORCE</td>
<td>4 EXTERNAL</td>
<td>Available</td>
</tr>
<tr>
<td>03260000 FORCE</td>
<td>7 DEFAULT</td>
<td>Unavailable</td>
</tr>
<tr>
<td>03240000 FORCE</td>
<td>7 DEFAULT</td>
<td>Available</td>
</tr>
<tr>
<td>02740000 HIGH</td>
<td>1 OPERATOR</td>
<td>Available</td>
</tr>
<tr>
<td>02720000 HIGH</td>
<td>1 OPERATOR</td>
<td>Unavailable</td>
</tr>
<tr>
<td>02680000 HIGH</td>
<td>6 A/A</td>
<td>Available</td>
</tr>
<tr>
<td>02670000 HIGH</td>
<td>6 A/A</td>
<td>Unavailable</td>
</tr>
<tr>
<td>02640000 HIGH</td>
<td>2 AUTOOPS</td>
<td>Available</td>
</tr>
<tr>
<td>02620000 HIGH</td>
<td>2 AUTOOPS</td>
<td>Unavailable</td>
</tr>
<tr>
<td>02540000 HIGH</td>
<td>3 E2EMGR</td>
<td>Available</td>
</tr>
<tr>
<td>02520000 HIGH</td>
<td>3 E2EMGR</td>
<td>Unavailable</td>
</tr>
<tr>
<td>02440000 HIGH</td>
<td>4 EXTERNAL</td>
<td>Available</td>
</tr>
<tr>
<td>02420000 HIGH</td>
<td>4 EXTERNAL</td>
<td>Unavailable</td>
</tr>
<tr>
<td>02240000 HIGH</td>
<td>7 DEFAULT</td>
<td>Available</td>
</tr>
<tr>
<td>02220000 HIGH</td>
<td>7 DEFAULT</td>
<td>Unavailable</td>
</tr>
<tr>
<td>02140000 HIGH</td>
<td>5 SCHEDULE</td>
<td>Available</td>
</tr>
<tr>
<td>02120000 HIGH</td>
<td>5 SCHEDULE</td>
<td>Unavailable</td>
</tr>
<tr>
<td>01740000 LOW</td>
<td>1 OPERATOR</td>
<td>Available</td>
</tr>
<tr>
<td>01720000 LOW</td>
<td>1 OPERATOR</td>
<td>Unavailable</td>
</tr>
<tr>
<td>01680000 LOW</td>
<td>6 A/A</td>
<td>Available</td>
</tr>
<tr>
<td>01670000 LOW</td>
<td>6 A/A</td>
<td>Unavailable</td>
</tr>
<tr>
<td>01640000 LOW</td>
<td>2 AUTOOPS</td>
<td>Available</td>
</tr>
<tr>
<td>01620000 LOW</td>
<td>2 AUTOOPS</td>
<td>Unavailable</td>
</tr>
<tr>
<td>01540000 LOW</td>
<td>3 E2EMGR</td>
<td>Available</td>
</tr>
<tr>
<td>01520000 LOW</td>
<td>3 E2EMGR</td>
<td>Unavailable</td>
</tr>
<tr>
<td>01440000 LOW</td>
<td>4 EXTERNAL</td>
<td>Available</td>
</tr>
<tr>
<td>01420000 LOW</td>
<td>4 EXTERNAL</td>
<td>Unavailable</td>
</tr>
<tr>
<td>01240000 LOW</td>
<td>7 DEFAULT</td>
<td>Available</td>
</tr>
<tr>
<td>01220000 LOW</td>
<td>7 DEFAULT</td>
<td>Unavailable</td>
</tr>
</tbody>
</table>
Table 2. Hierarchy of Requests (continued)

<table>
<thead>
<tr>
<th>Rank/Priority</th>
<th>Source</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>01140000 LOW</td>
<td>5 SCHEDULE</td>
<td>Available</td>
</tr>
<tr>
<td>01120000 LOW</td>
<td>5 SCHEDULE</td>
<td>Unavailable</td>
</tr>
<tr>
<td>00001000</td>
<td>8 GROUP</td>
<td>Available/Unavailable</td>
</tr>
</tbody>
</table>

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

2. Within the same method, for all priorities except FORCE, the request or goal to make a resource available has a higher priority than to make it unavailable. However, a FORCE UnAvailable request that is issued by an operator has the highest priority.
   This is really the only method to beat SA z/OS's attempts to make a resource available if for any reason you want to have this resource down.
   You can easily override your own FORCE Unavailable request for a resource with even a LOW priority request against the same resource. Thus this implementation of priorities ensures high flexibility and a quick reaction in emergency cases.

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

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

For enterprise monitoring, the automation manager has the task of gathering and controlling information about what resources are available, what the status of the resources is and what status updates occur during automation. The SA z/OS topology manager extracts and transfers this information to the enterprise-wide RODM data base from where this information is picked up by the NMC workstation to display user-defined graphical views.

The main commands you can use to retrieve information from the automation manager are:
- INGLIST displays detail information about one or more resources (subsystem, application group, and so on.)
- INGINFO displays lots of details for an individual resource or application group
- INGVOTE displays the requests that have been issued and are currently pending for a specified resource
- INGSCHED displays information about the current UP and DOWN service periods for resources
• INGGROUP displays the members of a group and their settings.
• INGRELS displays the relationships that are defined for a resource.

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

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

Summary of terminology:

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

**Note:** If no goal is explicitly specified, the default goal is the default desired status.

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

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

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

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

It may happen that votes conflict or compete with each other. Such a scenario is illustrated in “Conflicting Relationships and Requests” on page 20. In such a case, the vote with the highest priority will win. Even if votes or requests with the same priority come into conflict, the requested action decides which request will win: the request to make a resource available is higher than the request to make that resource unavailable. To make a resource unavailable when it has the request to make it available, you must send it a higher priority request or revoke the MakeAvailable request.
Using the INGVOTE command, you will see all currently active requests with their priorities, may they have been generated explicitly by an operator, implicitly because of the automation policy or internally as a consequence of relationships.

One automation manager must exist on each sysplex and one on each standalone system. In a sysplex, you can have one or more secondary automation managers (SAM) for backup purposes. That is, if the system that the primary automation manager (PAM) runs on, abends or if the automation manager itself abends, the first secondary automation manager recognizing this, will become the new PAM and will take over all current requests from the former PAM. The automation agents in the sysplex will not notice the switch from one automation manager to the other. It is important though that all automation managers (PAM and SAMs) run on systems which are in the same time zone.

How Requests Become Goals

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

Consider the following example:

```
INGREQ abc REQ=START
```

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

SA z/OS knows the dependencies for each of the actions that can be taken against the resource through the resource's relationships with its supporting resources. Using these, it propagates the request onto those resources. These internally created requests are called votes. Let us have a look at a CICS scenario: It makes sense to start a Terminal Owning Region (TOR_1) only after an Application Owning Region (AOR_1) is up. So you could define TOR_1 (running on a certain system) in our example to become a dependent resource and AOR_1 to become the supporting resource. So if you request to start TOR_1, SA z/OS would generate a vote to start AOR_1.

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

How Request Propagation Works

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

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

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

Relationship Support

One of the most effective automation capabilities of SA z/OS is the concept of dependency relationships. You define relationships between dependent resources and supporting resources.

In the customization dialog you always define relationships from the point of view of the dependent resource, and specify the name of the supporting resource.

Relationships can be specified together with a condition that must be satisfied for the supporting resource, before the specified action can be processed for the dependent resource.

Examples for relationships are MAKEAVAILABLE or MAKEUNAVAILABLE. Examples for conditions are WhenAvailable or WhenDown.

A typical dependency relationship that you might define for a dependent resource, A, would be MAKEAVAILABLE/WhenAvailable, where the condition WhenAvailable applies to a supporting resource B (see Figure 2).

So the relationship defined in Figure 2 would read: Only if the desired state for the dependent resource A is available, make it available as soon as the supporting resource B is available.

Figure 2. Relationships and Conditions

In the example shown in Figure 2, a request to become available is issued for resource A. Resource A cannot be made available unless B is available. SA z/OS implements the concept of request propagation. A request will be propagated along the dependency graph so that ultimately the original request can be fulfilled. In this example, the request to A will be propagated to B, which can make itself available because it is not dependent on anything else. Then, after B is available, A will become available. Thus the original request is persistent until it is withdrawn.

Consider another example: assume that A has its relationship with the supporting resource B defined as HASPASSIVEPARENT, and a request to become available is issued for A. This request will not be propagated along the dependency graph. A separate request must be issued for B to make the resource A available.

There is a special relationship, HASPARENT, that simplifies relationship specifications. It is equivalent to a MAKEAVAILABLE/WhenAvailable dependency between the dependent and supporting resource and a MAKEUNAVAILABLE/WhenDown dependency from the supporting resource to the dependent resource, as shown in Figure 3 on page 19.
Relationship Support

Figure 3. The HASPARENT Relationship

Relationships may be defined between any two resources within the same sysplex. The resources may be on different systems or may be Sysplex Application Groups. Figure 4 presents an example of relationships across system boundaries.

Note: An application group can also be either a supporting resource or a dependent resource. It can even be simultaneously a supporting resource and a dependent resource, as shown in Figure 4.

Be careful not to define loops.

Figure 4. Relationship in a Sysplex

Figure 4 shows an example of grouping and relationships defined across a sysplex consisting of the systems SYS1, SYS2 and SYS3.

In Figure 4 the arrows denote HASPARENT relationships. For example, CICS_TOR will be started as soon as one member of application group CICS_AOR/APG is available and VTAM® is available.

If a group is a dependent resource, all MAKEAVAILABLE relationships apply to the members of the group, as well as to the group as a whole. For example, in Figure 5 on page 20, resource APL will not start before the resources CICS, VTAM, and JES2 are available.
Relationship Support

Figure 5. Relationship between Applications and Application Groups

For information on how to define relationships see “RELATIONSHIPS Policy Item” in IBM Tivoli System Automation for z/OS Defining Automation Policy.

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

In complex environments, it can happen that relationships exist that are opposed in both direction and meaning, as illustrated in Figure 6 on page 21. The following abbreviations are used in this diagram:

- For dependency relationships:
  - **MaWa** denotes MAKEAVAILABLE/WhenAvailable
  - **MuWaD** denotes MAKEUNAVAILABLE/WhenAssumedDown

- For statuses:
  - **Unav** denotes Unavailable
  - **Avail** denotes Available

- For requests or votes:
  - **Av** denotes a request or vote to become available
  - **Un** denotes a request or vote to become unavailable
In Figure 6, resource A is in the status Unavailable (Unav) and receives a request to become available (which internally creates the corresponding vote).

The relationship MAKEAVAILABLE/WhenAvailable propagates requests to make resources available and the relationship MAKEUNAVAILABLE/WhenAssumedDown propagates requests to make resources unavailable.

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

Because requests are stored and propagated internally as votes, in Figure 6 you see two conflicting votes that would be propagated to resource E:

- From A we have the vote to be available
- From resource C we have the vote to be unavailable

E receives both of these votes but cannot satisfy them at the same time. You can view the conflicting votes for E by issuing the INGVOTE command against this resource.

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

A request to make a resource available wins over a request with the same priority to make it unavailable. This is only valid because of the propagation of requests. If, however, an operator sends a request to start a resource, and the same operator (or another one) later sends another request to stop that resource, the later request will win. The difference in this case is that this is not a propagation of requests.

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

As described in "Overview of Automation Logic" on page 11, SA z/OS's automation logic in a sysplex is divided between the automation agents and the automation manager.

For reasons of high availability, SA z/OS also offers the concept of the primary and secondary automation manager (PAM and SAM), where the SAM can take over the work of the PAM. For more details, refer to the section "Manager-Agent Communication and Status Backup" in the chapter "Planning to Install SA z/OS on Host Systems" in IBM Tivoli System Automation for z/OS Planning and Installation.
Relationship Support

In a sysplex, the automation agents communicate with each other using XCF. Automation managers also communicate with the automation agents using XCF and a VSAM data set for backup. Figure 7 shows a typical configuration in a sysplex, showing the communication paths and methods between its members.

Locked Votes and Requests

When the automation manager tells an automation agent to start or stop a resource, it internally locks the current winning vote on the resource. The automation agent is not generally expected to be able to stop an application halfway through its start sequence, or to be able to start an application if it is halfway through its stop sequence. The winning vote is locked to prevent it from changing and thus causing the automation manager to send the automation agent an order for a contrary action.

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

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

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

Goal Driven Automation

SA z/OS implements goal driven automation in two flavors:

- One flavor of goal driven automation in SA z/OS combines events and triggers on the one hand (see "Event and Trigger Support" on page 24), and service
periods on the other (see "Automatic Calculation of Schedules (Service Periods)" on page 24). This allows you to define goals (that is, to be available or unavailable) for applications, application groups, or monitor resources, and to specify external events that, in addition to the defined goals, need to be satisfied before an application, application group, or monitor resource can be actually started or stopped.

In SA z/OS you can define the applications, application groups, and monitor resources to be resources (application resources, application group resources and monitor resources) that you want to automate.

• Another flavor of goal driven automation is that automation programmers define the default behavior or state of resources using the customization dialog so that SA z/OS tries to keep the resource in the specified state during specified schedules under specified prerequisites.

If operators want to change the goal of a resource, they may issue or remove a request to start or stop it using the INGREQ or INGSET command with appropriate startup or shutdown parameters. A request is executed by SA z/OS only if it does not conflict with requests of higher priority. Otherwise, because requests are persistent, they only take effect when conflicting requests of higher priority are resolved. Operators must remove obsolete requests from SA z/OS.

Note: INGREQ requests are persistent. They are remembered across session boundaries until they are explicitly revoked.

Grouping Support

Modern applications are often composed of many components, such as data servers, networking and security components. Examples are client/server applications where the application logic is distributed between a client and one or more servers and where the data can also be distributed between two and more servers. These components are often spread among the various systems in the sysplex.

In SA z/OS you can automate a complete application group. If you want to have all members (for example, all applications) of an application group available, you just tell SA z/OS to have the application group available.

Figure 8 presents a grouping example. Complex applications that are made of individual application components are denoted with G1–G3 (G stands for group). The application components are denoted with A1 and B1. For example, the complex application group G2 comprises application group G1 and the single applications A4 and A5. Application group G1 in turn consists of three applications A1, A2, and A3. Also, application groups may comprise applications that may or may not be contained in other application groups. G3 is such an example.
See the chapter “Scenarios on How to Use the Customization Dialog” in *IBM Tivoli System Automation for z/OS Defining Automation Policy* for a scenario that shows how grouping is performed on the conceptual level.

**Automatic Calculation of Schedules (Service Periods)**

*Service periods* allow you to schedule the availability of applications, application groups, or monitor resources. A service period is a set of service windows during which a resource (application, application group, or monitor resource) should be active or inactive. So with service periods, you can specify both the up and down times for a resource.

Service periods set the desired status of the applications, application groups and monitor resources that they are linked to.

During the process of evaluating operator requests, SA z/OS calculates schedules for the availability of resources. This means that if you have a resource, say a database, that is supporting two applications, you can configure things so that the database will automatically be made available when either one of the two applications should be available, and to be unavailable otherwise.

If, for example, application A should be up from 7a.m. to 7p.m., and application B should be up from 9a.m. to 3p.m. and 5p.m. to 9p.m., simply setting these schedules for the applications and giving them a start dependency upon the database is sufficient to have SA z/OS calculate a schedule for the database of 7a.m. until 9p.m.. Note that this inference is done dynamically at runtime and will also take into account operator requests and schedule overrides.

For more details about service periods see [“Using Schedules” on page 105](#) and *IBM Tivoli System Automation for z/OS Defining Automation Policy*.

**Adaptive Scheduling**

When defining the sysplex policy, the automation programmer can specify that certain resources are backups for other resources. In order to meet or maintain application availability, SA z/OS will activate these resources (if no other requests or policy definitions are opposed). This means that if an application’s normal configuration is unavailable, SA z/OS will start an alternate configuration that will provide the same service. This can be configured so that no operator intervention is required.

This feature works best when the automation programmer sets schedules on a group level, rather than on an individual resource level. Applications need to be defined in groups, because it is selective request propagation by the group resources that achieves the effect.

**Event and Trigger Support**

This section describes the concepts of events and triggers and how you can use them. This is useful for understanding how to define automation policies for events and triggers using the customization dialog, see “Events Entry Type” and “Service Periods Entry Type” in *IBM Tivoli System Automation for z/OS Defining Automation Policy* for more details.

These entry types allow you to specify external conditions when certain applications on certain systems should either be automatically started or stopped.
Events are used to represent processes outside of automation. You can use the INGEVENT command to let SA z/OS know that an event has occurred. One or more events are part of a trigger condition. You can define a condition to be either a startup condition or a shutdown condition.

Events are used to decide whether the goal for a resource should be realized. Even if all dependency relationships and conditions are fulfilled, but a resource's startup condition is not fulfilled, because not all events of the startup condition are set, the resource will not be started.

An event can be in one of two states: set or unset. The state of an event is changed with an INGEVENT command. For more information on the INGEVENT command see IBM Tivoli System Automation for z/OS Operator’s Commands. The state of an event is set by the user (or a procedure) and memorized throughout the sysplex. Each time, an event is set, SA z/OS checks all triggers defined in your environment or in your sysplex to find out whether applications need to be started or shut down according to the new conditions.

Triggers can be connected to applications or application groups. They act as inhibitors for the requested action. For an application to be started, its desired status must be AVAILABLE and its startup trigger must be satisfied. For an application to be stopped, its desired status must be UNAVAILABLE and its shutdown trigger must be satisfied.

If a trigger is connected to an application group then, if the trigger startup condition for the application group is not satisfied, none of its members can be started. Also, if the trigger shutdown condition is not satisfied, none of the group members can be stopped. So the trigger is kind of inherited to the members of the application group. This is also valid if the group contains nested groups. Then the trigger is also inherited to the members of the nested group.

**Figure 9. How a Trigger Uses Events**

In Figure 9, the application connected to Trigger T1 would be started if either EVENT1 and EVENT2 are set or if EVENT3 is set, assuming the desired goal of the application is: available. The application would be stopped, if EVENT4 is set and the desired goal of the application is: unavailable.

Triggers are used to control the starting and stopping of applications in a single system or a sysplex. So the trigger must be linked to one or more applications and has a list of conditions attached to it.

A trigger's condition list consists of up to 10 startup conditions and up to 10 shutdown conditions, each having up to 20 events. Each condition is either a startup or a shutdown condition. All events within a condition must have the
Event and Trigger Support

status SET to meet the condition. If more than one condition of the same type, STARTUP or SHUTDOWN, is specified, only one of that type has to be met to satisfy the trigger.

Triggers and Traffic Lights

Like a traffic light, a trigger controls whether a running application should stop, or an application with the goal MAKEAVAILABLE should really drive away from a traffic light that switched to green, just like a ready-to start resource may only run if the trigger's startup conditions are satisfied. You see that the goal, or the request to start a resource must be there in order that a trigger with fulfilled startup conditions has an effect, just as a green traffic light is of no importance if there are no cars around the street.

However, a trigger with fulfilled shutdown conditions does not work like a red traffic light. Instead, a shutdown trigger (the red traffic light) stops the application only if the application's goal is MAKEUNAVAILABLE (while a red traffic light stops a running car anyhow). And an application with goal MAKEUNAVAILABLE is only stopped if a connected shutdown trigger is set.

![Figure 10. Triggers, Events, and Conditions](image)

Figure 10 shows another scenario of how triggers, events and conditions work together. For example, if EVENT1 and EVENT2 are set, Startup Condition 1 is true and Trigger T1 starts the application, provided that the application's goal is being available. If EVENT4 is set, Shutdown Condition 1 is true and Trigger T1 stops the application, provided that the application's goal is being unavailable.

A startup condition is satisfied when all events of the condition are set. A shutdown condition is satisfied when all its events are set.

Where no real dependency exists between the applications it is more appropriate to use events and triggers to control the STARTUP and SHUTDOWN.

Error Thresholds

Error thresholds influence whether or not SA z/OS recovers from an error situation. Thresholds are a number of error conditions within a certain time
Error Thresholds

interval, such as five error conditions requiring restart within one hour. During a condition that requires a restart, SA z/OS checks whether the number of occurrences of the condition reaches a defined critical threshold. If it is reached, SA z/OS does not attempt to restart the resource. You can define a critical threshold for applications to restart them.

For z/OS components, such as dump data sets or log data sets, you can define thresholds to limit how often they may be deleted after they have become full without an action being taken or a notification being sent to the operator.

Error thresholds also determine when you should be alerted to problems. The primary use of error thresholds is to track subsystem abends and ensure that the abend and restart cycle does not become an infinite loop, but you can also customize them for other uses.

The following sections show you how to define error thresholds. Information on how to display threshold settings and occurrences, and to interactively set thresholds (using the command INGTHRES) is provided in “How SA z/OS Uses Error Thresholds” on page 174.

Groups

Groups are active automation elements that manage the availability of a small set of resources. For example, a group might be all the resources needed to run a CICS or an IMS or, at a higher level, a complex application, relying on subgroups to manage the complex resources that are needed to make it work. The advantages of groups are that they automatically manage the availability of the resources they are looking after and provide you with a single point of control to request that resources are stopped or started. The down side is that if an operator does not tell them what they want them to do, they can end up working at cross purposes.

Group Types

There are four types of groups available within System Automation - Basic, Move, Server and Simple.

Basic Groups
These groups manage the resources defined to them as if they were a single resource. Ask the group to start and it starts them all, ask it to stop and it stops them all. If one of them fails and cannot be restarted, it concludes that the group as a whole is broken, which can lead to it being shutdown. This can also happen if the manager automation flag for one or more group member is turned off.

Move Groups
These groups manage the availability of a single resource, choosing between alternative instances of it on different systems. The resource may be a single APL or may be a group of resources. The group is responsible for picking the system to start it on and for ‘moving’ it to other systems as required to maintain its availability. A Move group will not pick a member that cannot be started. If its members are basic groups, this means that if a single member of the basic group has a problem, the Move group will not pick the entire Basic group.

Server Groups
Like Move Groups, Server Groups manage the availability of resources
Groups

over multiple systems. A Server Group gets a set of resource instances and a target number of them to keep available. It is responsible for picking the instances that originally get started and for ‘moving’ them around to maintain availability. Like Move Groups, Server Groups will not pick members that cannot be started.

Simple Groups
These groups are different, in that they are a construct only used in the Customization Dialogs. They do not appear as a resource. They are differentiated from other group types by the absence of an automation name. Simple groups are used to simply place resources on systems and provide no ongoing management of them once automation is running. Resources with Simple Groups are managed mostly by their Default Desired Status, propagated votes and schedules.

Relationships

Relationships upon a group work on the group, and are usually passed onto the group members as well. This means that if a group is a child of a resource, then every member of the group is also considered to be a child of the resource, at least from the point of view of working out what can be stopped and started. Note that there is an exception to this for Move and Server groups which allow their members to be stopped (provided the group as a whole is not being stopped) while their Stop dependency is unsatisfied. This is necessary to provide their ability to change their set of active members.

Runmodes

Runmodes are not groups, but they can provide a similar function at some levels. Runmodes let you ‘tag’ a resource with a runtoken, and then define runmodes as sets of runtokens. The INGRUN command can be used to activate or deactivate particular runmodes. Resources whose runtokens are not selected get shutdown.

It is strongly suggested that runmodes rather than large groups containing unrelated resources be used for ‘bulk’ application control on a system.

Operating Groups

You should start and stop groups with the same commands, INGREQ and INGVOTE, as you use to operate normal APL resources. These simply request that the group be started or stopped according to its predefined priorities.

Moving MOVE Groups

You can use the INGMOVE command to issue Move directives. These completely override the group’s default policy and force it to move to a specific system. Their effect persists until a specific command is issued to release the resource.

Controlling Application Move and Server Management

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

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

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

**Move Functions**

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

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

You can initiate move functions using either the INGMOVE or INGGROUP command.

**Using the INGMOVE Command**

The INGMOVE command allows you to specify where a sysplex application group should be moved to. INGMOVE provides details of the observed status of sysplex application groups and those systems that at least one member of the sysplex application group would be able to run on.

You can use the INGMOVE command dialog to:

- Initiate a move to a specified system.
- Prepare the move of a sysplex application group.
- Reset the preference values to those defined in the policy database.
- **Box** the group. The currently active member receives a preference of 3200 so that if the system where it runs is terminated, no other group member is activated instead and the group becomes unavailable.

This can be useful if, when IPLing the system that hosts the primary group member, moving that member to another system and back to its home system (that is, the system with the highest base preference value) later on takes longer than the entire IPL process.
Using the INGGROUP Command

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

EXCLUDE/AVOID/INCLUDE

This method allows you to:
- Remove movable members from a system (exclude)
- Prohibit SA z/OS from activating movable members on a system (avoid)
- Undo either of the other two actions (include)

Availability Target (AVTGT)

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

Preference (PREF)

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

Satisfactory Target (SATTGT)

This parameter specifies the number of members in the group that must be active before the automation manager treats the group as being in a satisfactory state. The value must be a positive, decimal number. The parameter will be ignored for groups other than server groups. You can dynamically overwrite a group’s satisfactory target that is statically defined in the automation policy.

RECYCLE/CANCEL

This causes a rolling recycle to be initiated for each selected server or move group. This consists of sequentially stopping and then restarting each member of the group, so that at the end of the recycle all members that were active when the operation was requested have been stopped and recycled or stopped and left down. The CHUNK parameter specifies the number of server group members to recycle in parallel. Specifying a chunk size so that only a few passes are necessary to recycle the entire server group will significantly improve the performance.

Controlling Application Move with Preference Values

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

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

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

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

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

<table>
<thead>
<tr>
<th>Bonus</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>Member is currently selected and has an adjusted preference ≤1000.</td>
</tr>
<tr>
<td>175</td>
<td>Member is currently selected and has an adjusted preference &gt;1000.</td>
</tr>
<tr>
<td>220</td>
<td>Member has an observed status of Starting, Stopping or Degraded and the system is not being stopped.</td>
</tr>
<tr>
<td>225</td>
<td>Member has an observed status of Available and the system is not being stopped.</td>
</tr>
<tr>
<td>-400</td>
<td>Member has an adjusted preference &gt;1500 and either the system is being stopped or the system has already been stopped and the member thus has an observed status of SysGone.</td>
</tr>
</tbody>
</table>

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

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

### Available Preference Values

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

**Table 3. Preference Values and Behavior of Members**

<table>
<thead>
<tr>
<th>Adjusted Preference</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>3100</td>
<td>The resource will always be selected, even if it has an observed status of HardDown, or SysGone, its system is stopping or its system has been excluded.</td>
</tr>
<tr>
<td>2900</td>
<td>The resource will always be selected unless it has an observed status of SysGone or its system is stopping and there is a viable alternative resource. It will still be selected if its system has been excluded or it has an observed status of HardDown.</td>
</tr>
</tbody>
</table>
Table 3. Preference Values and Behavior of Members (continued)

<table>
<thead>
<tr>
<th>Adjusted Preference</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>2400</td>
<td>The resource will always be selected unless it has an observed status of SysGone or HardDown, or its system is stopping and there is a viable alternative resource. It will still be selected if its system has been excluded.</td>
</tr>
<tr>
<td>1000</td>
<td>The member is always selected when the group has been started and the system associated to this member comes up. Other resources from the same group with preference 700 are stopped in favor of this member.</td>
</tr>
<tr>
<td>900</td>
<td>The member is always selected when the group has been started and the system associated to this member comes up. Members with preference 500 will be stopped in favor of this group.</td>
</tr>
<tr>
<td>700</td>
<td>The member is always selected when the group has been started and the system associated to this member comes up. Members with preference 300 will be stopped in favor of this group. It is not stopped unless a very high-preference alternative (1000) becomes available. This value is the default. Use it in a MOVE group for the resource that normally runs.</td>
</tr>
<tr>
<td>&lt;600</td>
<td>This value is used as a threshold. A resource with preference 600 and lower will not be started at group startup. Only picked if the group is in recovery mode.</td>
</tr>
<tr>
<td>500</td>
<td>This resource is not selected but will be selected when the group enters recovery mode. Once the problem is over, it will be stopped as soon as a primary member with a calculated preference &gt;750 is available.</td>
</tr>
<tr>
<td>300</td>
<td>This resource is not selected but will be selected when the group enters recovery mode. Once the problem is over, it will be stopped as soon as a primary member with a calculated preference &gt;550 is available.</td>
</tr>
<tr>
<td>100</td>
<td>This resource is not selected but will be selected when the group enters recovery mode. As soon as a preference 500 backup member is available, it will be stopped in favor of the backup member.</td>
</tr>
<tr>
<td>1</td>
<td>A resource with this preference will never be selected by SA z/OS automation; it is always sent an Offline vote. Operators may make such resources available interactively by using the INGGROUP command and giving it a higher preference. Use this value to provide completely manually controlled application availability.</td>
</tr>
<tr>
<td>0</td>
<td>This value indicates that the member is passive and will never have a vote propagated to it. If a member is made passive, any votes that have been propagated to it will be withdrawn.</td>
</tr>
</tbody>
</table>

Preference Scenarios

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

Backup with Non-Disruptive Restoration:

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

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

Backup with Disruptive Restoration:
This scenario is like the above, but you want the application to return to SYS1 as soon as SYS1 is available, even if this causes an additional service interruption. You can use preferences of 700 for SYS1 and 300 for SYS2, or 900 for SYS1 and 500 for SYS2. As these are the only two applications involved, either combination will work (the important thing is that they are more then 250 apart). The example uses the 700 and 300 values.
1. If SYS2 is IPLd first, the application will not be started (preference 300, not in recovery mode).
   When SYS1 is IPLd, its instance of the application will be started (preference 700, highest startable instance).
2. If SYS1 should fail, the instance on SYS2 will be started (preference 300, in recovery mode, highest startable instance).
3. When SYS1 is IPLd again, its instance will be started (preference 700, highest startable instance) and the SYS2 instance will be stopped (preference 550 (300+250 for being available), lower than 700). This causes a short service interruption as the application moves back to SYS1.
4. At the end of the service period the instance of SYS1 will be stopped.
5. At the start of the next service period, the instance of SYS1 will be started (preference 700, highest startable instance).

Figure 11 on page 34 illustrates the difference of non-disruptive and disruptive restoration between the steps 2 and 3 in each scenario: in the non-disruptive scenario, the application ABC continues to run on SYS2 until the end of the service period for the group, because this resource now has a higher priority of 750 than the resource on SYS1, which would only have priority 700. In the disruptive restoration on the other hand, SYS1 takes over for the rest of the service period after the problem has been fixed.
Real Hierarchy of Preferences:
In this scenario, you have three systems where the application can be run. SYS1 is where you normally run it, SYS2 is a back up and SYS3 is where it can run in extreme emergency only. You would specify preferences as 700, 500 and 100.

1. If either SYS2 or SYS3 are IPLd first, their instances will not be started (preferences 500 and 100, not in recovery mode). The operator can, of course, explicitly request one of the instances to be started (which should be done if they want the application to be available and know that SYS1 will not be IPLd for a while).

2. When SYS1 comes up its instance will be started (preference 700, highest startable instance).

3. If SYS1 fails, the SYS2 instance will be started (preference 500, in recovery mode, highest startable instance).

4. If SYS1 were to return at this point, we would be back in the first scenario, a non-disruptive backup.

5. Instead, let us suppose that SYS2 now fails. This causes the instance on SYS3 to be started (preference 100, in recovery mode, highest startable instance).

6. If either of the other two systems came back (let us assume it is SYS2), its instance will be started (preference 500, in recovery mode, highest preference instance) and the instance on SYS3 will be stopped (preference 350 [100 plus 250 for being available], lower than 500). This would cause an application outage, but would get the application off SYS3.

7. If SYS1 then comes back, its instance will not be started: Preference 700, lower than 750 (500+250 for being available).

8. After the next service period stop, we would be back to running the instance on SYS1.

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

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

1. When all systems are IPLd, the resources on SYS1, SYS2 and SYS3 will be started (preference 700, 3 highest startable resources). The resource on SYS4 will not be started (preference 300, not in recovery mode).

2. If either SYS1, SYS2 or SYS3 fails, the resource on SYS4 will be started (preference 300, now in recovery mode).

3. When the failed system is restored, its resources will be restarted (preference 700, three highest startable resources) and the resource on SYS4 will be stopped (preference 550 [300+250 for being available]).

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

Move Groups and Recovery Mode:

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

1. When all systems are IPLd, the resource on SYS1 will be started (preference 700, highest startable resource). The resources on SYS2, SYS3, and SYS4 will not be started (availability target of 1 and APG not in recovery mode with preferences ≤600).

2. If SYS1 should fail then we enter recovery mode, which means that SYS2 and SYS3 become eligible to be started. In this case SYS2 will be chosen (highest eligible preference 450). If however, SYS1 was a planned shutdown (INGREQ ALL) then we would not be in recovery mode and neither SYS2 nor SYS3 would be eligible, so the application would not be moved automatically. To move the resource to another system use INGGROUP to raise that system’s preference to >600.

3. If SYS2 should fail, as well as SYS1, the resource will be started on SYS3 without further adjustment (minimum startable preference 250).

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

Absolute Manual Control:

Member to be activated gets preference 3200, all others get preference 1.

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

This does not change even if the system with the selected member leaves the sysplex.
Controlling Application Move and Server Management

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

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

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

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

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

The backups are always viable.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

**Overview:** Resource aware application move is achieved by the automation manager querying WLM for the capacity data of all systems in a sysplex. Capacity information is returned in SUs. These are hardware-independent numbers that describe the amount of capacity a system can provide or an application uses.
WLM provides the number of SUs used per system. This is divided by the number of resources on the system to give an average. The average number of SUs that a resource uses is known as the resource weight.

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

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

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

If members of a MOVE or SERVER group that are to be moved are also groups then the decision-making is slightly different:
1. If all the group’s members are on the same system, the automation manager uses the expected free capacity of the system divided by the number of members of the group as the basis to decide which group to move first. This tends to favor groups with fewer members.
2. If the group’s members are spread over multiple systems, it is not possible to calculate a reasonable capacity value and resource aware application move is not possible.

Example Scenario:
Consider the sysplex in Figure 12. (Note that the SU values are for illustration only, normally they are 8- or 9-digit numbers.)

![Figure 12](image)

**Figure 12. Resource Aware Application Move Example Sysplex**

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

The initial free SU value of SYSB is 380 and of SYSC it is 300. SA z/OS first starts 5 resources, one after the other, on SYSB (see Figure 13 on page 39). Now there is more free capacity on SYSC, so SA z/OS then starts alternately one of the
remaining 5 resources on SYSC and SYSB. Each resource that is started reduces the free expected SUs by the resource weight (17 on SYSB and 15 on SYSC). This results in free SUs of 261 on SYSB and 255 on SYSC. Thus neither system is overloaded.

Figure 13. Resource Aware Application Move

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

```
WLMQUERYINTERVAL=5
```

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

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

SA z/OS Automation Flags

Automation flags are automation policy settings defined in the customization dialog that specify whether certain types of automation are on or off for a resource.

The Automation Flag Specification panels of the customization dialog define the cumulative effects of your automation flag settings, including:
- When automation is active
- Resources that automation applies to
- The types of automation mode you want SA z/OS to handle, such as starting and shutting down applications
- Any exceptions to the defined automation
- Whether any automated operator function will operate in log mode, an SA z/OS option allowing a human operator to log the flow of an automated operation

SA z/OS automation flags control whether automation occurs for a resource and, if so, when. Using the customization dialog, you can set default automation flags for all systems and applications, or you can override automation flags for specific systems or applications.

For more information about how to use automation flags and when SA z/OS checks automation flags, see IBM Tivoli System Automation for z/OS Customizing and Programming. See also “Determining Automation Agent Flag Values” on page 108.
Automation Agent Flags

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

Figure 14 shows how specific automation flags cover the phases in the lifetime of an application.

PRESTART and STARTUP commands are only issued if the INITSTART respective RESTART flag is turned on. POSTSTART commands are only issued if the START flag is turned on. SHUTINIT, SHUTNORM, SHUTIMMED, SHUTFORCE and SHUTFINAL commands are only issued if the TERMINATE flag is turned on.

REFRESHSTART and ANYSTART commands are issued regardless of any automation flag specification in case of NetView recycle or INGAMS REFRESH processing. During the start-up process of an application the ANYSTART commands are only issued if the START flag is turned on.

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

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

The other flags that apply to each resource are:

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

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

Recovery
If this is on, automation takes the specified action for any MVS message that is not associated with the initialization, startup, shutdown, or restart of a subsystem. For example, if this flag is on, automation takes corrective actions in response to messages that indicate shortages of WTO buffers and
SA z/OS Automation Flags

JES spool space. Automation also invokes offload routines when the following MVS data sets are about to fill up: SYSLOG, LOGREC, SMF, MVS dump.

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

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

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

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

  You can display these scheduled overrides using the DISPSCHD command dialog, as explained in "Displaying Automation Agent Flag Override Settings" on page 109.

- Turns the automation off using the INGAUTO command dialog. These command dialogs are discussed later in the chapter.

- Causes a higher level flag to be turned off, for example the subsystem’s Automation flag or the SUBSYSTEM Restart flag.

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

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

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

The Automation Manager Global Automation Flag

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

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

You can set automation flags for:

- System Defaults (using their AUTOMATION FLAGS policy item)
- Application Defaults (using their AUTOMATION FLAGS policy item)
- MVSCOMP Defaults (using their AUTOMATION FLAGS policy item)
- Applications (using their AUTOMATION FLAGS or MINOR RESOURCES policy item)
- MVS Components (using their MINOR RESOURCES policy item)

**Note:** For MVS components, only the Automation and Recovery flags apply.

Because flags have inherited values, it is not necessary to define them all explicitly. Specific resources need only have flags defined if they differ from the values inherited from the defaults.

A common step during automated handling of events is to check the automation flags to determine whether automation is on or off and whether automated handling of an event can continue.

The AOCQRY command performs this automation flag check. See *IBM Tivoli System Automation for z/OS Programmer’s Reference* for more information.

The benefit of this automation hierarchy is that you can customize SA z/OS to automate as many activities and resources as necessary in the z/OS environment. You can choose to have SA z/OS handle all phases of startup, monitoring, recovery, and shutdown for all resources or have a human operator control some phases for some resources. You can control all automation by setting just one flag (the Automation flag) on one panel and then setting exceptions. The simplest way to automate your resources is to use the Automation flag.

Periods that automation is not to apply for can be specified.

There are several commands to display automation flag settings, for example, DISPFLGS. The INGAUTO command dialog can be used to set or override automation flags.

SA z/OS provides an exit capability for automation flag processing. See *IBM Tivoli System Automation for z/OS Customizing and Programming* for more information.

**Relationship Between Automation Flags and Resource Status**

From initialization through normal operation to shutdown, a resource can have several automation modes, such as being started, in an up state, abending, and shutting down. SA z/OS uses resource states to track these automation modes for monitored resources. A resource must have a particular status for certain automated actions to occur. The effect of these actions may, in turn, change the resource status from one value to another.

Understanding how SA z/OS typically uses resource states and automation flags is important when planning and writing your own automation procedures. For more details on automation flags, resource states, and their use in automation procedures, see *IBM Tivoli System Automation for z/OS Customizing and Programming*.

SA z/OS provides an exit for processing state changes. See *IBM Tivoli System Automation for z/OS Customizing and Programming* for more information.
Resource States and the Status Display Facility
SA z/OS also uses resource states in status information forwarded to the status display facility (SDF). This status information results in status changes being reflected on SDF panels.

Automation Flags and Minor Resource Flags
In addition to the automation flag settings for the entire application, you can set flags for minor resources (MINOR RESOURCES policy item) that control responses to specific situations. Typical minor resources include message IDs and states.

For example, TSO is defined as application TSO. You have defined a reply to message IKT010D on the MESSAGES/USER DATA policy item of the Application policy object for TSO. You can change the shutdown process by:
- Turning the Terminate flag for TSO off. This means you have to shut down TSO manually.
- Turning the Terminate flag for minor resource IKT010D off. This allows SA z/OS to shut down TSO normally. When the IKT010D message is issued, shutdown processing waits for a manual response.

Statuses Supplied by the Automation Manager
The automation manager keeps track of various statuses for each resource:

- **Observed Status**
  This indicates the current status of the resource, as reported by the automation agent. See “Observed Status” on page 44.

- **Automation Status**
  The automation status of a resource tells the automation manager what automation status a resource is currently in.
  
  If a resource's automation status is currently *idle*, the automation agent does nothing for automation. Conversely, if the automation status is *denied*, automation is currently not allowed, possibly because the automation flag has been set off. See “Automation Status” on page 46.

- **Desired Status**
  This status reflects the goal of a resource. The automation manager will be continuously trying to place the resource into this status unless the goal is revoked or changed. This status can be either *Available* or *Unavailable*.
  
  If it is *Available*, the automation manager will try to start the resource. If it is *Unavailable*, the automation manager will try to stop the resource.

  You can define a *default* desired status for a resource:
  
  **ALWAYS**
  The desired status is set to Available and the resource is started once its dependencies are satisfied.

  **ONDEMAND**
  The desired status is set to Unavailable and the resource is not be started. To start it a MakeAvailable vote must be propagated to it. This can come from an operator request, a service period, or an application group request.

  **ASIS**
  The desired status is always set to the observed status, that is, the group remains in the status it currently has and no action is taken by SA z/OS at any time, as long as there is no request placed or propagated to the group.
Statuses Supplied by the Automation Manager

See “Desired Status” on page 46.

Startability Status
This is a calculation as to whether it is possible to start the resource if this is requested. It includes the startability of all inactive start dependent resources. This status is mainly used as input for Compound Status calculations and by groups to prevent them from selecting members that cannot be started. See “Startability Status” on page 47.

Compound Status
This status is the composition of the other statuses to give the operator a chance to check from the automation manager if a resource is currently OK concerning the automation. See “Compound Status” on page 47.

Health Status
The health status can be used by the automation manager to make decisions and, if necessary, trigger automation for the application if there is a change in the health state. See “Health Status” on page 48.

An operator can retrieve the status using two commands:

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

Three of these statuses are supplied by the automation agent that is responsible for running the resource:

- Observed status
- Automation status
- Health status

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

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

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

Possible values are:

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

**SysGone (unavailable)**
The system that the resource is defined on is no longer a member of the sysplex.
Statues Supplied by the Automation Manager

SoftDown (unavailable)
The resource is unavailable and automation may restart it.

HardDown (unavailable)
The resource is unavailable and automation will not restart it.

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

Available
The resource is ready for use.

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

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

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

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

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

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

Observed Status Transitions

![Observed Status Transitions Diagram]

Figure 15. Observed Status Transitions
Automation Status

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

Possible values are:

- **Unknown**
  No connection with the automation agent.

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

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

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

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

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

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

Automation Status Transition

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

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

Desired Status

You can set the desired status of a resource either:

- With the INGREQ command


Statutes Supplied by the Automation Manager

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

Possible values are:

**Available**
The resource should be started (made available).

**Unavailable**
The resource should be stopped (made unavailable).

**Startability Status**
The startability status is calculated from the resource's current status, its automation status, its automation flag and the startability of its supporting resources.

Possible values are:

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

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

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

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

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

Possible values are:

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

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

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

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

**InAuto[mation]**
SA z/OS is in the process of starting or stopping the resource.

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

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

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

Possible values are:

- **Normal**
The observed resource is behaving normally.
- **Warning**
The resource's health state has crossed the warning threshold.
- **Minor**
The resource's health state has crossed the minor threshold.
- **Critical**
The resource's health state has crossed the critical threshold.
- **Fatal**
The resource's health state has crossed the fatal threshold.
- **Unknown**
The health state of the resource is not known.

### Application Statuses Supplied by Automation Agents
There are different types of statuses that may be applied to an application. They are all retrieved on the side of the automation agent. These are:

- “Automation Agent Statuses”
- “Application Monitor Status” on page 56
- “Automatic Restart Manager Statuses” on page 56

This chapter discusses the following related topics:

- “Mapping the Automation Agent Statuses to the Automation Manager Observed Status” on page 57
- “SA z/OS Processes Involved in Status Determination” on page 61

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

These statuses are used on the automation agents to control the automation. They are translated to observed statuses (see “Mapping the Automation Agent Statuses to the Automation Manager Observed Status” on page 57) and then sent to the automation manager.

On the automation agent you can use the DISPSTAT and DISPINFO commands to see the automation statuses.
### Table 4. Automation Agent Statuses

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ABENDING</strong></td>
<td>The application is undergoing a recoverable abnormal end. This status is entered when a TERMINATION message with the ABEND=YES attribute is received. It remains in ABENDING until its final termination message is received. If the application is defined to Automatic Restart Manager it may be posted to EXTSTART when Automatic Restart Manager attempts to recover it. If the application has not exceeded its critical threshold, SA z/OS posts the application to RESTART and attempts to restart it. If it has exceeded its critical threshold it is put into BROKEN status when it has been cleaned up, and SA z/OS does not attempt to start it. If it is slow to leave the system after its FINAL message is received it may go to ZOMBIE status. If the application is undergoing a normal shutdown by SA z/OS no further shutdown commands are issued. The shutdown may be resumed by using the INGREQ command with a stronger type.</td>
</tr>
</tbody>
</table>
| **ACTIVE** | The application is running, but is not yet ready for work. An application can be put into ACTIVE status in response to the following conditions:  
  1. SA z/OS has received an ACTIVE message for the application.  
  2. The SA z/OS startup checker has run for the application when its automation status was STARTED. The startup checker found the application monitor status for the application to be ACTIVE.  
  3. During routine status checking the application monitor status for the application was found to be ACTIVE when its automation status indicated it should be otherwise.  
  4. SA z/OS attempted to start the application, but found that its application monitor status was ACTIVE.  
  5. SA z/OS found that the Automatic Restart Manager status for the application is STARTING, but has not received any messages concerning the application status.  
  6. SA z/OS checked an attempt by Automatic Restart Manager to restart the application and found that its Automatic Restart Manager status is UNREGISTERED, but the application monitor status for the application is ACTIVE.  
An application remains in ACTIVE status until its UP message is received. If the application is starting or restarting, an SA z/OS monitoring check for the application is done after its start delay time and number of start cycles. If the application is not found and if the critical threshold is not reached, SA z/OS will attempt to restart the application. If it is found, but is not yet UP, it is put into STARTED2 status. |
| **AUTODOWN** | The application is shut down. SA z/OS may restart it to comply with an operator request. If the shutdown specified that the application was to be restarted, it is put into RESTART status when the shutdown is complete.  
  1. SA z/OS has shut the application down.  
  2. When SA z/OS initialized, the operator replied NOSTART to the AOF603D WTOR. Any applications that would have been put into the DOWN status during initial status determination have instead been put into the AUTODOWN automation status. You can use the SETSTATE command to change the application status to either RESTART or CTLDOWN. SA z/OS may attempt to restart an application in AUTODOWN status when SA z/OS is reloaded, or when an operator requests SA z/OS to start one of the application descendents. In both cases, the application goes to RESTART status. |
### Statuses Supplied by Automation Agents

Table 4. Automation Agent Statuses (continued)

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
</table>
| AUTOTERM   | SA z/OS is in the process of shutting the application down. The shutdown is in response to a INGREQ REQ=STOP command. This status persists until SA z/OS is sure that the application has been cleaned up.  
Many things may happen to an application that is being shut down. If the shutdown is successful, the application is placed in either AUTODOWN or CTLDOWN status. If the shutdown specified that the application should be restarted it goes through AUTODOWN to RESTART status.  
If the application abnormally ends while it is being shut down it may go into either ABENDING or BREAKING. A normal shutdown will stop processing an application that abends, but other shutdowns will continue. If the shutdown runs out of commands to issue, the application is placed into STUCK status. If it has problems shutting down, the application may be placed into ZOMBIE status. |
| BREAKING   | The application is undergoing a nonrecoverable abend; that is, it has received a termination message specifying BREAK=YES. If the application is undergoing a normal shutdown by SA z/OS no further shutdown commands are issued. The shutdown may be resumed by using the INGREQ REQ=STOP command. This status persists until SA z/OS receives its final termination message and is sure that the application has been cleaned up. If the termination experiences difficulties, the application may be posted to ZOMBIE status. |
| BROKEN     | The application has suffered a nonrecoverable abend. SA z/OS will not restart it. An application can be put into BROKEN status in response to the following conditions:  
1. The application has suffered a nonrecoverable abend, indicated by the reception of a TERMINATION message with the BREAK=YES attribute.  
2. The application has suffered sufficient recoverable abends to exceed its critical threshold.  
3. Issuing the application’s prestart or startup command resulted in a non-zero return code.  
This status is preserved across a recycle of SA z/OS or a re-IPL of the processor, unless the application does not have its Restart after IPL option set to NOSTART. |
| CTLDOWN    | The application is shut down and SA z/OS is not allowed to restart it.  
1. An operator asked SA z/OS to shut the application down and not to restart it until authorized to do so by an operator.  
2. An operator used a SETSTATE command to tell SA z/OS that an application should not be restarted until an operator authorizes SA z/OS to do so.  
You can use the SETSTATE command to change an application status from CTLDOWN to RESTART or AUTODOWN, in which case SA z/OS will attempt to restart it. |
| DOWN       | The application has not been started during the lifetime of this SA z/OS.  
The DOWN status is set only during initial status determination and is possible only if the application monitor status is INACTIVE. The automation status of the application when SA z/OS was last shut down on this system is used in the following manner to determine if it is to be placed into the DOWN status.  
1. The previous automation status was not one of STOPPED, CTLDOWN or BROKEN.  
2. The application does not have its Restart after IPL option set to NOSTART, the previous state was BROKEN, CTLDOWN or STOPPED, and this is the first time the agent has been started since the system was IPLed. |
| ENDED      | This status is used for transient applications only, and indicates that the job for the application has finished and left the system without any errors. Any start-dependent resources for the application will be started as though it were a normal z/OS subsystem that was UP.  
If the transient application can be rerun, you can use the SETSTATE command to restart it. If the transient application cannot be rerun, it will remain in ENDED status. |
## Statuses Supplied by Automation Agents

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
</table>
| ENDING   | A transient application is in the process of terminating. A transient application goes to ENDING status when a termination message is received for it, and it is not being shut down by SA z/OS. This status shows that the application is terminating, but that this is expected. A transient application may also go to ENDING if an operator is shutting it down outside SA z/OS control, or if it has abnormally ended, but the abend messages are being treated as normal termination messages. The application remains in ENDING status until either:  
  1. An abend message is received that will put it into either ABENDING or BREAKING status. If the application abends then either SA z/OS or Automatic Restart Manager can restart it.  
  2. The application final termination message is received, at which point the RESTARTOPT for the application is checked. If it is ALWAYS then the application is put into RESTART status and SA z/OS will attempt to restart it. If it is anything else, the application goes to ENDED status. It is assumed that if a transient application ends normally then it will deregister from Automatic Restart Manager. If it is slow to clear the system after its FINAL message is received it may go to ZOMBIE status.  
   If an ACTIVE or UP message is received for the application, its automation status is changed to either ACTIVE or UP, as appropriate. |
| EXTSTART | SA z/OS has determined that the application is being started or restarted by an agent external to SA z/OS. In situations where SA z/OS is able to identify the external agent (such as Automatic Restart Manager), it takes appropriate steps to monitor that agent's actions and, if necessary, step in to assist it. An application can be put into EXTSTART status in response to the following conditions:  
  1. SA z/OS is unable to identify the external agent.  
  2. Automatic Restart Manager is in the process of restarting the application. |
| FALLBACK | The application is not running on the primary system where it should run, but this status has been encountered for this application on one of its secondary systems. It should be active on another system. If the other system fails, the application can fall back to this system, where it could possibly be restarted. However, SA z/OS, will not perform the restart on the fallback system, but this may be done by an operator request. This is implemented to leave the decision of restarting the application on the fallback system to the operator.  
   An application can be put into FALLBACK status in response to the following conditions:  
   1. It is defined with a secondary association on this system.  
   2. An operator has used the SETSTATE command to put the application into the MOVED status. If this is one of the secondary systems for the application it will go to FALLBACK instead. |
| HALFDOWN | SA z/OS was in the process of shutting the application down, but the stop request was canceled while it was in progress. The application shutdown did not complete (for example, ASCBs may still be active). You may sometimes find that some, but not all, of the shutdown commands have been issued. To recover an application from HALFDOWN status you must determine where it is in its shutdown process and complete the shutdown manually. Applications go into HALFDOWN only when you cancel a stop request. Alternatively, you can use SETSTATE to put the application back into UP or RUNNING status. |
| HALTED   | The application is still running, but something has happened that may have severely impacted its capabilities.  
  1. SA z/OS has received a HALT message for the application.  
  2. SA z/OS has detected that the application represents a JES2 application that is running short of spool space.  
  3. The Automatic Restart Manager status for the application is ELSEWHERE, but SA z/OS found its application monitor status to be either STARTING or ACTIVE.  
   An application is taken out of HALTED status if its UP message is received. Also, operators may use the SETSTATE command to put the application into UP or RUNNING status. |
### Statuses Supplied by Automation Agents

#### Table 4. Automation Agent Statuses (continued)

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
</table>
| MOVED   | The application is not running: this is one of its primary systems: it should be active on another system. An application can be put into MOVED status in response to the following conditions:  
- An operator has used the SETSTATE command to put the application into the MOVED status. This is possible only on a primary system.  
- The startup detected that Automatic Restart Manager has started the subsystem on another system.  
A subsystem will remain in the MOVED status until it is restarted on the primary system by an external agent, such as an operator. |
| RESTART | The application is ready to be started. It has been previously active in the system. An application can be put into RESTART status in response to the following conditions:  
- The application abended and, after checking thresholds, SA z/OS is allowed to restart it.  
- SA z/OS has shut the application down in response to an operator request and is now preparing to restart it.  
- An operator has used the INGREQ REQ=START command to ask SA z/OS to restart the application.  
- SA z/OS checked an attempt by Automatic Restart Manager to restart the application and found that its Automatic Restart Manager status is UNREGISTERED and the application monitor status for the application is INACTIVE. This implies that the attempt by Automatic Restart Manager to restart the application timed out while the application was in RESTARTING status. SA z/OS changes the automation status of the application to RESTART and attempts to start the application itself.  
During restart processing, the application RESTART automation flag is checked. If it is turned on, the application start commands are issued and the application is put into STARTED status. If the RESTART automation flag is off, the application remains in RESTART status and the startup monitor cycle initiates the startup process each time it runs. |
| RUNNING | This status is equivalent to UP, but is used for transient applications. It indicates that the UP message has been received for the transient application, or an operator has used the SETSTATE command to change the status of a transient application to UP. A transient application is one that SA z/OS expects to terminate on its own. When the job finishes the application goes through ENDING status to ENDED, at which point its descendants are started. Unlike the UP status, the descendants of a transient application are not started until it has terminated.  
A transient application should leave RUNNING status on its own. If it gets stuck, you should investigate it. You can use the INGREQ REQ=STOP command to put it into AUTODOWN status. |
| STARTED | The commands to start the application have been issued, but it has yet to start running. An application can be put into STARTED status in response to the following conditions:  
- SA z/OS has issued, or will soon issue, the commands.  
- When SA z/OS attempted to start the application it found that the application monitor status for the application was STARTING.  
- During initial status determination, SA z/OS found that the application monitor status for the application was STARTING.  
- SA z/OS checked an attempt by Automatic Restart Manager to restart the application and found that its Automatic Restart Manager status is UNREGISTERED, but the application monitor status for the application is STARTING.  
Note that the relevant automation flag, Initstart or Restart, must be on. The application startup commands as defined in the automation control file are issued after the application is placed in STARTING status.  
An application remains in STARTING status until either its ACTIVE or its UP message arrives. After the application start delay time, an SA z/OS monitoring check is issued for it. If it is not found, and if the critical threshold is not reached, SA z/OS will attempt to restart the application. If it is found, but is not yet UP, it is put into ACTIVE status. |
### Statuses Supplied by Automation Agents

**Table 4. Automation Agent Statuses (continued)**

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
</table>
| **STARTED2** | The application has become active, but has not indicated that it is ready to accept work within its *start delay* and number of start cycles. An application can be put into STARTED2 status in response to the following conditions:  
  - The startup checker found that the application monitor status was still STARTING.  
  - The startup checker was called for an application whose automation status was ACTIVE.  
  - SA z/OS found the Automatic Restart Manager status for the application to be AVAILABLE-TO.  
  - SA z/OS checked an attempt by Automatic Restart Manager to restart the application and found that its Automatic Restart Manager status is one of AVAILABLE-TO, RESTARTING or RECOVERING.  
  - SA z/OS checked an attempt by Automatic Restart Manager to restart the application and could not find a better status to put it into. Its application monitor status is ACTIVE. This covers situations where the restart checker is unable to determine the Automatic Restart Manager status for the application, or its status is FAILED.  

An application remains in STARTED2 status until either its UP message arrives or an operator uses the SETSTATE command to change the application status to UP or RUNNING. |
| **STOPPED** | The application has been shut down by an external agent, such as an operator cancel. SA z/OS is not permitted to restart it and will not allow the Automatic Restart Manager to restart it. This status is preserved across a recycle of SA z/OS or a re-IPL of the processor unless the application has its Restart after IPL Option set to START or NOSTART.  

An application remains in STOPPED status until an operator uses the SETSTATE command to change its status to either RESTART or CTLDOWN.  
An application may also leave STOPPED status if it is restarted outside the control of SA z/OS. In this case, it goes to either the ACTIVE or UP status, depending on which message is received. |
| **STOPPING** | The application is being shut down by an external agent. This status is entered if a TERMINATION message is received but SA z/OS is not in the process of shutting the application down. It may also indicate that the application is abending, but the abend messages issued are being treated as normal termination messages by automation.  

An application that is STOPPING remains in that status until either an abend message is received that puts it into ABENDING or BREAKING status, or the final termination message for the application is received, at which point the application RESTART option is checked. If RESTART is ALWAYS, the application is put into RESTART status and SA z/OS attempts to restart it. Otherwise, the application goes to STOPPED status. If it is slow to clear the system after the final message is received, it may be placed into ZOMBIE status.  
If an ACTIVE or UP message is received for the application, its automation status is changed to either ACTIVE or UP, as appropriate.  
Automatic Restart Manager interaction with this status depends on a number of factors. If the application is deregistered then there is no interaction. If the application is registered then Automatic Restart Manager will attempt to restart it. If the application goes to STOPPED status before the SA z/OS or Automatic Restart Manager exit is invoked then SA z/OS will tell Automatic Restart Manager not to restart the application. If it does not, SA z/OS will tell Automatic Restart Manager that it can restart the application. |
| **STUCK** | An application can get STUCK when it is being shut down. This is because SA z/OS has run out of NORM, IMMED, or FORCE shutdown commands. |
### Table 4. Automation Agent Statuses (continued)

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
</table>
| **UP** | The application has finished initializing and is ready for work. An application can be put into UP status in response to the following conditions:  
  - SA z/OS has received an UP message for the application.  
  - An operator has used the SETSTATE command to change the application automation status. In this case, SA z/OS assumes that the operator has ensured that the application is actually UP.  
  - During initial status determination the application monitor status was found to be ACTIVE.  
  - SA z/OS found the Automatic Restart Manager status for the application to be AVAILABLE.  
There are a number of ways for an application to leave the UP status, if:  
  - It is shut down with the INGREQ REQ=STOP command, it goes to AUTOTERM status  
  - It is shut down outside SA z/OS, it goes to STOPPING status  
  - It abends, it might go to STOPPING, ABENDING or BREAKING status  
  - It has problems, it may go to HALTED status  
  - The regular monitor cannot find it, it will call the TERMMSG generic routine  
  - The application abends, SA z/OS does not pick up the abend messages, and Automatic Restart Manager detects that the address space has ended, the application may go to EXTSTART. |
| **ZOMBIE** | When an application is leaving the system it can enter a ZOMBIE status. This indicates that the final termination message for the application has been received but that SA z/OS monitoring still finds the application. SA z/OS retries monitoring after a delay and the application is put into ZOMBIE status if this situation persists for more than eight retries.  
There are three ways that an application can enter a ZOMBIE status:  
  - If MVS is slow in clearing the application and the termination delay time is short.  
  - If there are two jobs with the same name in the system, one of which is the application. When either of them terminates and SA z/OS does not know the address space ID of the application or does not get the address space ID with the termination message, SA z/OS assumes that the application has stopped, but SA z/OS monitoring will find the other job. To change the status to UP, either manually shut down the other job, or use the SETSTATE command to change the application status back to UP.  
  - The job may have become stuck in the system after issuing its final message.  
From ZOMBIE status, if the application suffers an unrecoverable abend it will go into BREAKING status.  
**Note:** The Restart after IPL option of the customization dialog may override these resource statuses at SA z/OS IPL or recycle, resulting in SA z/OS starting the subsystem.  

[Figure 17 on page 55](#) and [Figure 18 on page 55](#) indicate the relationships between the automation statuses. You can change the states illustrated here with the SETSTATE command (see "Changing the Automation Agent Status" on page 116).
Figure 17. Transitions for a Normal Subsystem

Figure 18. Transitions for an Abending Subsystem
**Application Monitor Status**

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

<table>
<thead>
<tr>
<th>Application Monitor Status</th>
<th>Automation Status Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVE</td>
<td>The application automation status is changed to ACTIVE.</td>
</tr>
<tr>
<td>STARTING</td>
<td>The application automation status is changed to STARTED and a startup checker is scheduled for it.</td>
</tr>
<tr>
<td>INACTIVE</td>
<td>If the application is defined with a RESTART option of ALWAYS, the automation status is changed to RESTART and the restart process is invoked. If its RESTART Option is not ALWAYS, its automation status is changed to INACTIVE.</td>
</tr>
</tbody>
</table>

**Automatic Restart Manager Statuses**

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

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVAILABLE</td>
<td>The element name is registered with Automatic Restart Manager, the job is running on this system and the Automatic Restart Manager status for it is AVAILABLE.</td>
</tr>
<tr>
<td>AVAILABLE-TO</td>
<td>The element name is registered with Automatic Restart Manager and the job is running on this system. The job has been restarted by Automatic Restart Manager, has issued its IXCARM REGISTER call, but did not issue its IXCARM READY call within the ready timeout defined to z/OS for it.</td>
</tr>
<tr>
<td>STARTING</td>
<td>The element name is registered to Automatic Restart Manager and the job is running on this system. The job was started by an agent external to MVS, has issued its IXCARM REGISTER call, but has not yet issued its IXCARM READY call. Unlike the RECOVERING status, this status does not time out to the AVAILABLE-TO status.</td>
</tr>
<tr>
<td>RECOVERING</td>
<td>The element name is registered to Automatic Restart Manager and the job is running on this system. Automatic Restart Manager has issued the restart command for the application, it has issued its IXCARM REGISTER call, but has not yet issued its IXCARM READY call. If it takes too long to do this, it is placed in the AVAILABLE-TO status.</td>
</tr>
</tbody>
</table>
### Statuses Supplied by Automation Agents

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAILED</td>
<td>The element name is registered with Automatic Restart Manager, the job has failed, but was last active on this system. It is unknown at this point if Automatic Restart Manager will attempt to restart the job. SA z/OS will continue to report an elements status as FAILED until MVS either deregisters it, or is committed to restarting it. MVS query commands issued during this time will probably return a RESTARTING status, however MVS Automatic Restart Manager is still consulting its policies and exits to determine whether it should restart the element.</td>
</tr>
<tr>
<td>RESTARTING</td>
<td>The element name is registered with Automatic Restart Manager, the job has failed and Automatic Restart Manager is attempting to restart it. This status is entered very early in the restart process, before Automatic Restart Manager is committed to restarting the element. SA z/OS continues to report the status as FAILED until Automatic Restart Manager is committed to issuing the restart commands. This means that MVS query commands may report a status of RESTARTING although SA z/OS reports a status of FAILED. The RESTARTING status is exited when the job issues its IXCARM REGISTER call. If it takes too long to do this, its element name is deregistered.</td>
</tr>
<tr>
<td>ELSEWHERE</td>
<td>The element name is registered with Automatic Restart Manager, but is currently active on a different system. Unless that system fails, Automatic Restart Manager will not move the application to this system.</td>
</tr>
<tr>
<td>UNREGISTERED</td>
<td>The element name is not, currently, registered with Automatic Restart Manager. This may mean that the application has never been active within the sysplex, or it may mean that it failed and Automatic Restart Manager could not recover it. If the SA z/OS routine to determine an Automatic Restart Manager status finds that Automatic Restart Manager recovery is inactive, the UNREGISTERED Automatic Restart Manager status is returned for all enquiries. This means that SA z/OS effectively ignores Automatic Restart Manager.</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>Automation Agent</th>
<th>Observed</th>
<th>Automation</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOWN</td>
<td>SoftDown</td>
<td>Idle</td>
<td>The automation agent is ready to perform an initial start of the resource.</td>
</tr>
<tr>
<td>RESTART</td>
<td>SoftDown</td>
<td>Idle</td>
<td>The automation agent is ready to perform a startup of the resource.</td>
</tr>
<tr>
<td>STARTED</td>
<td>Starting</td>
<td>Busy</td>
<td>The automation agent has issued the start commands for the resource.</td>
</tr>
<tr>
<td>EXTSTART</td>
<td>Starting</td>
<td>Busy</td>
<td>This means that the automation agent has detected that an external agency (probably ARM) is in the process of starting the resource.</td>
</tr>
</tbody>
</table>
### Statuses Supplied by Automation Agents

<table>
<thead>
<tr>
<th>Automation Agent</th>
<th>Observed</th>
<th>Automation</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVE</td>
<td>Starting</td>
<td>Busy</td>
<td>The automation agent has received some indication that the start is successful, although it is not yet complete.</td>
</tr>
<tr>
<td>STARTED2</td>
<td>Problem</td>
<td>Problem</td>
<td>Although the resource is still active, it does not appear to have started properly within the time period that the automation agent expected it to start within.</td>
</tr>
<tr>
<td>UP</td>
<td>Available</td>
<td>Idle</td>
<td>The resource has entered a stable state and is available for usage.</td>
</tr>
<tr>
<td>HALTED</td>
<td>Degraded</td>
<td>Busy</td>
<td>The resource has encountered a problem, but is still active. The automation agent may be attempting to return it to an UP state.</td>
</tr>
<tr>
<td>RUNNING</td>
<td>Starting</td>
<td>Busy</td>
<td>The resource is a transient subsystem that is currently executing.</td>
</tr>
<tr>
<td>ENDING</td>
<td>Starting</td>
<td>Busy</td>
<td>The resource is a transient subsystems that is terminating.</td>
</tr>
<tr>
<td>AUTOTERM</td>
<td>Stopping</td>
<td>Busy</td>
<td>The automation agent is in the process of stopping the resource.</td>
</tr>
<tr>
<td>STOPPING</td>
<td>Stopping</td>
<td>Busy</td>
<td>The automation agent has detected that the application is terminating normally outside of its control.</td>
</tr>
<tr>
<td>ABENDING</td>
<td>Stopping</td>
<td>Busy</td>
<td>The automation agent has determined that the application is terminating abnormally but can be recovered.</td>
</tr>
<tr>
<td>BREAKING</td>
<td>Stopping</td>
<td>Busy</td>
<td>The automation agent has determined that the resource is terminating abnormally in a nonrecoverable fashion.</td>
</tr>
<tr>
<td>ENDED</td>
<td>Available</td>
<td>Idle</td>
<td>The resource is a transient subsystem that has completed successfully.</td>
</tr>
<tr>
<td>AUTODOWN</td>
<td>SoftDown</td>
<td>Idle</td>
<td>The resource is down and may be restarted as needed by automation.</td>
</tr>
<tr>
<td>CTLDOWN</td>
<td>HardDown</td>
<td>Idle</td>
<td>The resource is down and may not be restarted by automation.</td>
</tr>
<tr>
<td>STOPPED</td>
<td>HardDown</td>
<td>Idle</td>
<td>The automation agent has determined that the resource has been stopped normally outside the control of automation and its policy indicates that an automated restart is not allowed.</td>
</tr>
<tr>
<td>BROKEN</td>
<td>HardDown</td>
<td>Idle</td>
<td>The automation agent has determined that the application has failed in a nonrecoverable fashion.</td>
</tr>
<tr>
<td>HALFDOWN</td>
<td>Degraded</td>
<td>Problem</td>
<td>A shutdown for the resource by the automation agent was cancelled at the users request. Although the resource is still active, its status is indeterminate.</td>
</tr>
<tr>
<td>STUCK</td>
<td>Stopping</td>
<td>Problem</td>
<td>While trying to stop or start the resource, the automation agent ran out of instructions.</td>
</tr>
</tbody>
</table>
### Table 6. Automation Agent to Automation Manager Status Mapping (continued)

<table>
<thead>
<tr>
<th>Automation Agent</th>
<th>Observed</th>
<th>Automation</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZOMBIE</td>
<td>Problem</td>
<td>Problem</td>
<td>The automation agent believes that the resource should be inactive, but it is not.</td>
</tr>
<tr>
<td>MOVED</td>
<td>Standby</td>
<td>Idle</td>
<td>The automation agent has determined that the resource is active elsewhere within the sysplex. The manager may reactivate it on this system.</td>
</tr>
<tr>
<td>FALLBACK</td>
<td>Standby</td>
<td>Idle</td>
<td>The resource is a backup for a resource that normally runs on a different system within the sysplex. The manager may start it if it needs to.</td>
</tr>
</tbody>
</table>

### Monitor Statuses Supplied by Automation Agents

Table 7 describes the statuses available for a monitor and supplied by the automation agents. The statuses for a monitor include both the monitor status itself and the health status of the monitored object.

#### Table 7. Monitor Statuses

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INACTIVE</td>
<td>The monitor is not running.</td>
</tr>
<tr>
<td>FAILED</td>
<td>The monitor has failed. Recovery may be in progress. No acceptable health status was provided.</td>
</tr>
<tr>
<td>BROKEN</td>
<td>Both the monitor and recovery failed. This is a permanent condition. The monitor will not be re-invoked.</td>
</tr>
<tr>
<td>UNKNOWN</td>
<td>The health status is not available yet.</td>
</tr>
<tr>
<td>NORMAL</td>
<td>The monitor has obtained good results from the object, or objects, that is watching.</td>
</tr>
<tr>
<td>WARNING</td>
<td>The monitor detected a certain degree of degradation in the operation of the monitored object.</td>
</tr>
<tr>
<td>MINOR</td>
<td>The same as WARNING, but more severe.</td>
</tr>
<tr>
<td>CRITICAL</td>
<td>The same as MINOR, but more severe.</td>
</tr>
<tr>
<td>FATAL</td>
<td>The same as CRITICAL, but more severe.</td>
</tr>
</tbody>
</table>

The significance of a monitoring status, in combination with other application and agent statuses is described in Table 8.

#### Table 8. Monitor Status Mapping

<table>
<thead>
<tr>
<th>Monitor</th>
<th>Health</th>
<th>Observed</th>
<th>Agent</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>INACTIVE</td>
<td>UNKNOWN</td>
<td>UNAVAILABLE</td>
<td>IDLE</td>
<td>The monitor is not running.</td>
</tr>
<tr>
<td>STARTING</td>
<td>UNKNOWN</td>
<td>STARTING</td>
<td>BUSY</td>
<td>The monitor is about to start. An activate command might be in progress.</td>
</tr>
<tr>
<td>Monitor</td>
<td>Health</td>
<td>Observed</td>
<td>Agent</td>
<td>Meaning</td>
</tr>
<tr>
<td>---------</td>
<td>-----------</td>
<td>----------</td>
<td>-------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ACTIVE</td>
<td>UNKNOWN</td>
<td>AVAILABLE</td>
<td>IDLE</td>
<td>The monitor is running. No health status has been determined yet. This might be a passive monitor.</td>
</tr>
<tr>
<td>ACTIVE</td>
<td>NORMAL</td>
<td>AVAILABLE</td>
<td>IDLE</td>
<td>The monitor is running and good health is reported.</td>
</tr>
<tr>
<td>ACTIVE</td>
<td>WARNING</td>
<td>AVAILABLE</td>
<td>IDLE</td>
<td>The monitor is running and a problem with health has been reported.</td>
</tr>
<tr>
<td>ACTIVE</td>
<td>MINOR</td>
<td>AVAILABLE</td>
<td>IDLE</td>
<td>Same as ACTIVE/WARNING but more severe.</td>
</tr>
<tr>
<td>ACTIVE</td>
<td>CRITICAL</td>
<td>AVAILABLE</td>
<td>IDLE</td>
<td>Same as ACTIVE/MINOR but more severe.</td>
</tr>
<tr>
<td>ACTIVE</td>
<td>FATAL</td>
<td>AVAILABLE</td>
<td>IDLE</td>
<td>Same as ACTIVE/CRITICAL but more severe.</td>
</tr>
<tr>
<td>FAILED</td>
<td>UNKNOWN</td>
<td>AVAILABLE</td>
<td>IDLE</td>
<td>The monitor is running. A health status could not be determined.</td>
</tr>
<tr>
<td>STOPPING</td>
<td>UNKNOWN</td>
<td>STOPPING</td>
<td>BUSY</td>
<td>The monitor is about to stop. A deactivate command might be in progress. The health status is maintained from the last ACTIVE status.</td>
</tr>
<tr>
<td>BROKEN</td>
<td>UNKNOWN</td>
<td>HARDDOWN</td>
<td>IDLE</td>
<td>The monitor suffered a nonrecoverable problem and is therefore stopped. SA z/OS will not restart it.</td>
</tr>
</tbody>
</table>
SA z/OS Processes Involved in Status Determination

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

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

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

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

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

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

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

- UP
- RUNNING
- ACTIVE
- HALTED

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

- DOWN
- RESTART
- INACTIVE
- AUTODOWN
- CTLDOWN
- BROKEN
- STOPPED
- ENDED
- FALLBACK
- MOVED
**Statues Supplied by Automation Agents**

**STARTING STATUSES**
These statuses indicate that the application monitor status may be STARTING or ACTIVE. They include:
- STARTED
- STARTED2
- EXTSTART

**TRANSITIONAL STATUSES**
These statuses are either part of the shutdown process, or indicate that the application may or may not be present in the machine. All application monitor statuses are possible with any transitional status. They include:
- AUTOTERM
- STUCK
- ZOMBIE
- BREAKING
- ABENDING
- HALFDOWN
- STOPPING
- ENDING

If regular monitoring finds a status mismatch, it changes the application automation status according to Table 5 on page 56.

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

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

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

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

**Using SA z/OS for Monitoring**
As an operator, you can monitor and display information about the resources using two types of interfaces:
- The NMC workstation provides monitoring capability for system operations and processor operations.
- The SA z/OS 3270-type interface provides monitoring capability for:
  - Processor operations
Using SA z/OS for Monitoring

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

You can also monitor messages and alerts. More information about this topic is given in "How to Monitor Alerts on a NetView Console" on page 131.

If SA z/OS has been configured properly, status mismatches between the actual and the monitored status should not occur. In the customization dialog, the automation programmer can specify a monitoring interval for every resource individually. If such an individual interval is not defined for a resource, every 59 minutes (which is the default, but the interval is user specifiable) SA z/OS will run a monitor routine to check the resources' status as known to SA z/OS versus its actual status. If the result indicates an error, its automation status will be updated.

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

Resynchronization

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

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

Inform List concepts

SA z/OS presents information about automation resources such as applications (APL), application groups (APG), monitor resources (MTR) or MVS components (MVC) on NetView panels shown by the corresponding operator commands. For example, the commands DISPINFO and INGINFO show details for a particular subsystem.

These panels, however, are not designed for continuous monitoring of the operating environment. Instead, important information such as status changes or abnormal conditions need to be forwarded to consoles or other targets that are better suited for these tasks.

Therefore SA z/OS allows you to register automation resources for one or more targets and in case there are situations where these targets need to be informed, SA z/OS automatically forwards this information to these targets, provided they have been set up and configured accordingly.
Inform List Concepts

The Inform List field can be specified for individual resources (APL, APG, MTR) but also in the MVS Component (MDF), the system (SDF) or the sysplex (XDF) defaults.

In the following sections, the supported targets are described in more detail.

Inform SDF, NMC

The Status Display Facility (SDF) and the NetView Management Console (NMC) are informed about any status change of automation resources. So, for example, if an APL fails, the transitional ABENDING status will be forwarded directly to these targets resulting in a real-time update of the corresponding console.

SDF and NMC also receive any exceptional message that has been captured either by means of the policy or directly (implicitly by SA z/OS or explicitly by user-defined Automation Table entries or scripts). Refer to “Message Capturing” on page 116 for more information.

For details about using SDF, refer to “Monitoring with the Status Display Facility” on page 131. For information about customizing SDF, refer to IBM Tivoli System Automation for z/OS Planning and Installation and IBM Tivoli System Automation for z/OS Programmer’s Reference.

For details about using NMC, refer to Chapter 10, “Using the NetView Management Console for SA z/OS,” on page 203. For more information about installing and customizing the NMC, refer to IBM Tivoli System Automation for z/OS Planning and Installation.

Inform ITM

When you use the Tivoli Enterprise Portal to monitor the operational environment, you can use so-called “pure event situations” to monitor certain messages of interest. When ITM is specified in the Inform List field, SA z/OS uses its Monitoring Agent to create a pure event for each message captured either by means of the policy or directly (implicitly by SA z/OS or explicitly by user-defined Automation Table entries or scripts). Refer to “Message Capturing” on page 116 for more information.

The SA z/OS Monitoring Agent has to be defined in the policy as an APL with Category ITM and Subcategory KAHAGENT.

For details about installing and configuring the SA z/OS Monitoring Agent, refer to IBM Tivoli System Automation for z/OS Monitoring Agent Configuration and User’s Guide.

For details about creating situations, refer to IBM Tivoli Monitoring Tivoli Enterprise Portal User’s Guide.

Inform IOM, EIF, USR, TTT

The targets IOM, EIF, USR and TTT all have in common that they are intended for alert-based notification. Alerts are generated at an alert point, which can be pre-defined by SA z/OS, user-defined by means of using the INGALERT command or when exceptional messages are captured with a severity of CRITICAL (refer to “Message Capturing” on page 116 for more information).
The keywords have the following meaning:

**IOM**
Alerts are sent to System Automation for Integrated Operations Management – SA IOM.

**EIF**
Alerts in form of so-called *Event Integration Facility* (EIF) events are sent to a software product such as Tivoli Netcool OMNIbus.

**USR**
Customized alert information can be sent by a user-defined alert handler to some user-defined target.

**TTT**
Alerts can be transformed to incident records (Tivoli Trouble Tickets) via the Tivoli Directory Integrator (TDI) and then sent to the IBM SmartCloud Control Desk.

For more details about using alert-based notification, refer to *IBM Tivoli System Automation for z/OS Customizing and Programming* and refer to the INGALERT and INGCNTL commands in *IBM Tivoli System Automation for z/OS Programmer’s Reference*.

### Inform SMF

When the **Inform List** field contains **SMF**, status changes for APL, APG, and MTR resources are reflected by an SMF record of type 114. To write SMF records of type 114, SMF recording must be enabled. For reporting purposes, SMF records have to be dumped into a sequential data set first and then processed by the INGPUSMF batch utility, which creates a tabular report suited for a spreadsheet application.

For more information about using SMF reporting, refer to the chapter "Availability and Recovery Time Reporting" in *IBM Tivoli System Automation for z/OS Customizing and Programming*.

### Automation Operators

Automation operators are crucial to the operation of SA z/OS. The SA z/OS sample automation control file supplies definitions that are required in order to start SA z/OS. If you modify the sample definitions, or you do not want to use them, you need to be aware of which automation operators are required, and what messages should be assigned to them.

SA z/OS defines an *automated function name* for each automation operator that it uses. Certain automated function names must be defined. These are listed in [Table 9 on page 66](#). The automated function names are included in the sample policy databases that are supplied with SA z/OS. If you are not using a sample policy database, you must include these automated function names in your policy database. The automation operators that they are associated with must be defined in the NetView DSIOPF member.
### Automated Function Names/OPERATORS Policy Item

[Table 9](#) lists the automated function names that need to be defined.

#### Table 9. SA z/OS Mapping of NetView Automation Operator to SA z/OS Automated Function

<table>
<thead>
<tr>
<th>Automated Function</th>
<th>Automation Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTO1</td>
<td>AUTO1</td>
<td>Used for SA z/OS initialization</td>
</tr>
<tr>
<td>AUTO2</td>
<td>AUTO2</td>
<td>Used for SA z/OS initialization</td>
</tr>
<tr>
<td>BASEOPER</td>
<td>AUTBASE</td>
<td>Base automation: Backs up all other automation operators.</td>
</tr>
<tr>
<td>SYSOPER</td>
<td>AUTSYS</td>
<td>System automation: Acts as the primary system automation operator.</td>
</tr>
<tr>
<td>LOGOPER</td>
<td>AUTLOG</td>
<td>Used for message writing and for unassigned messages.</td>
</tr>
<tr>
<td>MSGOPER</td>
<td>AUTMSG</td>
<td>Message automation: Formats and routes messages.</td>
</tr>
<tr>
<td>AOFWRKxx</td>
<td>AUTWRKxx</td>
<td>Used for subsystem and monitor resource processing, for example startup, shutdown.</td>
</tr>
<tr>
<td>NETOPER</td>
<td>AUTNET1, AUTNET2</td>
<td>VTAM and NetView automation: Acts as the network products automation operator.</td>
</tr>
<tr>
<td>JESOPER</td>
<td>AUTJES</td>
<td>JES automation: Acts as the JES2 or JES3 product automation operator.</td>
</tr>
<tr>
<td>MONOPER</td>
<td>AUTMON</td>
<td>Monitoring and startup automation: Performs active monitoring of resources.</td>
</tr>
<tr>
<td>RECOPER</td>
<td>AUTREC</td>
<td>Recovery automation: Acts as the primary recovery operator. It is not recommended that you use RECOPER for user automation that issues WTO/Rs. This situation can halt SA z/OS’s WTO/R buffer recovery process, which may result in a buffer full situation.</td>
</tr>
<tr>
<td>RPCOPER</td>
<td>AUTRPC</td>
<td>Used for XCF requests</td>
</tr>
<tr>
<td>SHUTOPER</td>
<td>AUTSHUT</td>
<td>Shutdown automation: Acts as the base shutdown operator.</td>
</tr>
<tr>
<td>GSSOPER</td>
<td>AUTGSS</td>
<td>Used for general automation and communication services for the automation manager(AM), including orders sent from the AM to the automation agents. Acts as the operator for applications using SA z/OS generic automation routines, but does not execute these routines. Instead, it distributes them to the responsible AOFWRKxx automated functions. GSSOPER will also be used as backup in case there is no AOFWRKxx automated function defined. Processes start and stop orders coming from the automation manager, before they are distributed to the responsible work operators.</td>
</tr>
<tr>
<td>GATOPER</td>
<td>GAT&amp;DOMID.</td>
<td>Gateway: Acts as the outbound gateway operator for automation notification message forwarding</td>
</tr>
<tr>
<td>MVSCONS/i</td>
<td>AUTCON</td>
<td>Used for the MVS and NetView console interface.</td>
</tr>
<tr>
<td>PPIOPER</td>
<td>AOFARCAT</td>
<td>ARM PPI receiver</td>
</tr>
<tr>
<td>HBOPER</td>
<td>AUTHB</td>
<td>Used by heartbeat function</td>
</tr>
</tbody>
</table>
### Automation Operators

Table 9. SA z/OS Mapping of NetView Automation Operator to SA z/OS Automated Function (continued)

<table>
<thead>
<tr>
<th>Automated Function</th>
<th>Automation Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HBSLV</td>
<td>AUTHBSLV</td>
<td>Used for heartbeat slave processing.</td>
</tr>
<tr>
<td>XCFOPER</td>
<td>AUTXCF</td>
<td>Used for enhanced Parallel Sysplex自动化 functions.</td>
</tr>
<tr>
<td>XCFOPER2</td>
<td>AUTXCF2</td>
<td>Used for enhanced Parallel Sysplex automation functions.</td>
</tr>
<tr>
<td>HWOPER1–33</td>
<td>AUTHW001–033</td>
<td>Used for Hardware Validation messages.</td>
</tr>
<tr>
<td>PLEXOPER</td>
<td>AUTPLEX</td>
<td>Used for enhanced Parallel Sysplex automation functions.</td>
</tr>
<tr>
<td>PLEXOPR2</td>
<td>AUTPLEX2</td>
<td>Used for enhanced Parallel Sysplex automation functions.</td>
</tr>
<tr>
<td>PLEXOPR3</td>
<td>AUTPLEX3</td>
<td>Used for enhanced Parallel Sysplex automation functions.</td>
</tr>
<tr>
<td>POSTOPER</td>
<td>AUTPOST</td>
<td>Used for the serialization of INGPOST commands on the target system.</td>
</tr>
<tr>
<td>POSTSLV</td>
<td>AUTOPOSTS</td>
<td>Used for the serialization of INGPOST commands on the focal point.</td>
</tr>
<tr>
<td>AOFSExx</td>
<td>AUTSExx</td>
<td>Used for OMEGAMON® sessions.</td>
</tr>
<tr>
<td>EVTOPER</td>
<td>AUTEVT1, AUTEVT2</td>
<td>Used for event subscription and the TWS command request interface.</td>
</tr>
<tr>
<td>ALRTOPER</td>
<td>AUTALERT</td>
<td>Used for alert-based notification.</td>
</tr>
<tr>
<td>OPCACMDR</td>
<td>AUTOPCR</td>
<td>Used to execute requests from TWS batch jobs.</td>
</tr>
<tr>
<td>OPCAMSTR</td>
<td>AUTOPCP</td>
<td>Used to receive requests from TWS.</td>
</tr>
<tr>
<td>OPCAOFR3</td>
<td>AUTOPCE</td>
<td>Used to execute requests from TWS.</td>
</tr>
<tr>
<td>AOFTWS01-06</td>
<td>AUTTWS01-06</td>
<td>Used for TWS requests.</td>
</tr>
<tr>
<td>CICSMSTR</td>
<td>AUTCICS</td>
<td>Used to execute requests from TWS batch jobs.</td>
</tr>
<tr>
<td>AOFKAH01-10</td>
<td>AUTKAH01-10</td>
<td>Used to execute requests from the SA Monitoring Agent.</td>
</tr>
<tr>
<td>AOFRDSEV</td>
<td>AUTRDSEV</td>
<td>Used to execute requests from the RDS Archiver.</td>
</tr>
<tr>
<td>AOFRDSAR</td>
<td>AUTRDSAR</td>
<td>Used to execute requests from the RDS Archiver.</td>
</tr>
<tr>
<td>AOFCMD01-05</td>
<td>AUTCMD01-05</td>
<td>Used to execute requests from the Command Receiver.</td>
</tr>
<tr>
<td>CICSMSTR</td>
<td>AUTCICS</td>
<td>Used for CICS Automation.</td>
</tr>
<tr>
<td>IMSMSTR</td>
<td>AUTIMS</td>
<td>Used for IMS Automation.</td>
</tr>
<tr>
<td>E2EOPER</td>
<td>AUTE2E</td>
<td>Used to execute requests from the E2E Adapter.</td>
</tr>
<tr>
<td>E2EOPR01-03</td>
<td>AUTE2E01-03</td>
<td>Used to execute requests from the E2E Adapter.</td>
</tr>
<tr>
<td>TECOPER</td>
<td>AUTOTEC</td>
<td>Used for TEC Notification.</td>
</tr>
</tbody>
</table>

It is highly recommended that you assign the suggested automated operators to these required automated functions as listed in Table 9 on page 66. If you attempt to share automated operators across more than one automated function you will experience a significant performance degradation during SA z/OS initialization or an INGAMS REFRESH.
Automation Operators

Sample DSIOPF definition statements are included in the AOFOPF* samples that are delivered with SA z/OS. You may define additional automation operators, to suit your automation needs.

When defining SA z/OS automated functions, you must associate the SA z/OS automated function names with actual automation operator IDs defined in member DSIOPF in the NetView DSIPARM data set. For example, you can associate the SA z/OS name BASEOPER with the NetView automation operator ID AUTBASE, MSGOPER with AUTMSG, and JESOPER with AUTJES. You must also define synonyms for them in AOFMSGSY. See IBM Tivoli System Automation for z/OS Customizing and Programming for more information on synonyms.

You can find required definitions of automated functions in several add-on policies (for example, *BASE).

Structure of AOFWRKxx Operators

AOFWRKxx operators are used to distribute the work on different NetView tasks, so that they can run in parallel. AOFWRKxx operators are used as dedicated automated function responsible for all the work on a particular subsystem. However, because there might be too many subsystems defined, it may not be possible that each subsystem has a unique automated function for exclusive usage. Instead, several subsystems will share the same automated function. This automated function is responsible for the subsystem's startup, shutdown, monitoring and message handling activities.

You must define at least three automated functions which are responsible for the subsystem automation. You have the possibility to define additional ones. So you can decide how much parallelism you want.

The naming of these automation operator entries is of type AOFWRKxx where xx is any number or character. The xx part of the automated function name does not have to be in sequence.

At SA z/OS initialization or an INGAMS REFRESH, each defined subsystem will be assigned an automated function from the pool of defined AOFWRKxx operators.

Behavior of SHUTDOWN, STARTUP and GSSOPER

There are changes in the behavior of the shutdown, the startup and in the usage of the GSSOPER automated function.

SHUTDOWN and STARTUP:
A shutdown/startup can continue to process another subsystem on another automated function before a shutdown/startup command is completed. The shutdown/startup process can run in parallel.

GSSOPER Usage:
The GSSOPER automated function is used for the generic routines ACTIVMSG, TERMMSG and HALTMSG. These routines are routed from the message automation table to GSSOPER. The GSSOPER automated function will no longer execute these routines, it will only distribute them to the responsible AOFWRKxx automation operators. GSSOPER will also be used as backup when there is no AOFWRKxx automation operator defined.
Starting, Stopping, and Maintaining the Automation Manager

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

Starting the Automation Manager: START (S)

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

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

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

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

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

Also you cannot start more than nine instances of an automation manager per z/OS system within the sysplex or per single system. To allow your installation to specify a preference as to which automation manager instance should assume PAM responsibility at sysplex IPL, each automation manager enters a DELAY state. The duration of this state is set with the DELAY parameter of the automation manager initialization member of the PARMLIB (see also the description of the HASPRM00
Starting, Stopping and Maintaining the Automation Manager

sample member in IBM Tivoli System Automation for z/OS Planning and Installation). This allows other instances with a lower delay value to assume PAM mode if started within the difference (in seconds) between the respective delay values. This makes sense when you would prefer that the primary automation manager runs on a special powerful processor.

Stopping the Automation Manager: STOP (P)

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

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

Making an Automation Manager the Primary

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

F jobname,GO

It is also possible to set the manager as the primary using the MVS MODIFY (F) command.

F jobname, SETMODE,PAM

Stopping the Automation Manager with Different Options

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

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

The available options are:

NORMAL
The instance will be stopped after all in progress work has been completed and committed. No new work will be accepted from the automation agents, however any internally queued work will be processed. This is the default option.

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

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

DEFER
The instance will be conditionally and normally stopped after the
Stopping, Stopping and Maintaining the Automation Manager

STOPDELAY interval has expired. Termination will occur only if another (secondary) instance is available, or if no automation agent is active.

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

**NORESTART**

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

**RESTART**

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

**Shutdown of Primary Automation Manager (PAM) system**

Provided a Secondary automation manager (SAM) is running in the sysplex, shutting down the PAM system is normally straightforward, with the SAM on an alternate system taking over the role of the PAM (see the section "Recovery Concepts for Automation Manager" in *IBM Tivoli System Automation for z/OS Planning and Installation*).

If however multiple system shutdowns or IPLs are initiated in the sysplex, it is occasionally possible to be left without an effective PAM. To avoid this, it is good practice to move the PAM to a stable system before initiating the shutdowns.

If the manager APLs are members of a MOVE or SERVER group, the INGGROUP command may be used with the EXCLUDE or AVOID parameters to facilitate the move (see *IBM Tivoli System Automation for z/OS Operator’s Commands*, INGGROUP).

**Obtaining Automation Manager Diagnostic Information**

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

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

  F jobname,DIAGINFO,SNAP,data_set_name

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

  F jobname,DIAGINFO,RECORD,data_set_name

  With a second command, you terminate the log:

  F jobname,DIAGINFO,ENDREC

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

Takeover File

The takeover file is dynamically allocated during the initialization of the primary automation manager (PAM). The name of the takeover file is defined in the HSAPRMxx parmlib member. The name of the takeover file cannot be changed when an automation manager (PAM or SAM) is running. All automation managers (in the same XCF group) must use the same takeover file name.

The takeover file is allocated using the DD name HSATKOVR. Note that an automation manager started as a SAM does not allocate the takeover file.

If an allocation error or a VSAM open/read error occurs during a hot start or takeover processing, the operator is prompted to decide whether to continue with a Warm start and lose all the information that is stored in the takeover file or to retry reading the takeover file. If the operator continues with a Warm start and this fails too, the automation manager terminates, which causes another takeover.

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

The automation manager allows you to repair the takeover file while the PAM and the SAMs are running. To do this:

1. Disable the takeover file if this has not already been done by the PAM.
2. Repair the file, for example, by enlarging it.
3. Enable the takeover file again.

Starting NetView (System Operations)

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

Your system programmer may have set up your system so that SA z/OS is started automatically when the Automation NetView initializes. (Directions for performing this setup are given in IBM Tivoli System Automation for z/OS Planning and Installation). As SA z/OS initializes, a multiline write-to-operator (MLWTO) message, AOF767I, is issued describing the available options. See Appendix A, “Automation Initialization Options,” on page 223 for the options that can be defined and for details about the MLWTO. To continue automation, enter:

```
R number
```

Where number is the message reply number of the accompanying AOF603D message. This is the default. If you do not issue a reply within two minutes,
automation continues and SA z/OS loads automation policy from an automation
control file. If Runmodes are defined in the policy with this reply, you could also
set a runmode.

**Note:** Use the AOFINITREPLY advanced automation option (AAO) to suppress the
issuing of the AOF603D message and have SA z/OS initialize with the default
settings (see also the table “Global Variables to Enable Advanced Automation
(CGLOBALS)” in the appendix “Global Variables” IBM Tivoli System Automation for
z/OS Customizing and Programming).

### Starting and Stopping I/O Operations

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

1. I/O operations is started as an application running on the Automation NetView
   at the focal point system and at the target systems. To start I/O operations,
   issue the following command from the z/OS operator console, specifying the
   I/O operations procedure name that has been created as a member in your
   SYS1.PROCLIB or USER.PROCLIB (the location of the I/O operations
   procedure depends on your installation). In the following example, INGE10 is
   the procedure name. Optionally, you can append an MVS identifier, which is 10
   in the example.

   `/S INGE10.IO, SUB=MSTR[, PARM='TIMEOUT=n']`

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

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

2. You stop I/O operations by issuing one of the commands /P INGE10 or /P 10.

### Starting and Stopping Processor Operations

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

   `ISQSTART ACF`

   For more details about the ISQSTART command, see *IBM Tivoli System
   Automation for z/OS Operator’s Commands*.

2. You stop processor operations by issuing **ISQSTOP** from the NetView operator
   console.
Starting and Stopping Processor Operations
Chapter 4. Refreshing Automation Policy

The recommended method to refresh the SA z/OS automation policy is to issue the command with the REFRESH option:

INGAMS REFRESH

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

Other Ways to Refresh Automation Policy

Another way to refresh the automation policy is to use:

- The REFRESH subcommand of the MVS MODIFY (F) command, see "Refreshing an Automation Manager Configuration"

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

ACF ATLOAD

How the ATs and MRT are refreshed and reloaded depends on the setting of the AOFSMARTMAT advanced automation option (AAO). For more details see its entry in the table “Global Variables to Enable Advanced Automation (CGLOBALS)” in the appendix “Global Variables” in IBM Tivoli System Automation for z/OS Customizing and Programming).

Refreshing an Automation Manager Configuration

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

F jobname,REFRESH,new_data_set_name

The new_data_set_name may be specified in several forms, as follows:

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

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

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

*(generation)
This specification form specifies that a member of the same GDG used for the previous refresh operation is to be used. The generation number is relative to the most recently created GDG member at the time of command entry.
### Identifying Current Configuration Files

The automation configuration file is the file that controls system-level automation. It contains the SA z/OS automation policy, which specifies what, when, and how to automate. The NetView automation table (AT) specifies which automation procedure to run when various messages are received. The message revision table (MRT) enables user-defined modification of the attributes and text of original z/OS messages.

An automation configuration file is made available during SA z/OS initialization. The AT and MRT are loaded into NetView and the automation manager configuration file is loaded into SA z/OS.

The following sections describe how to identify the configuration file that is being used by an automation agent or the primary automation manager.

#### For an Automation Agent

You can use the DISPSYS command to identify the automation configuration file that is currently active. Figure 19 shows an example of the information that is displayed.

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

Message automation table(s)
DSITBL01 INGMSGSA INGMSG01
```

Figure 19. DISPSYS Command Dialog

You can also use the details option of the INGAMS command. If you enter the command code B (Show Details) for an automation agent it displays the DISPSYS output shown in Figure 19.

#### For the Primary Automation Manager

To identify the current automation manager configuration file, you use the details option of the INGAMS command. On the INGAMS command dialog, you enter the command code B (Show Details) for a primary automation manager. A panel similar to Figure 20 on page 77 is displayed, which shows relevant information about the configuration files.

Example:
REFRESH,*(0) - use most recent generation of the specified GDG
Identifying Current Configuration Files

<table>
<thead>
<tr>
<th>Config dataset name</th>
<th>OPER1.USER.KEYPLEX.ACF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Config member</td>
<td>HSAZ099</td>
</tr>
<tr>
<td>Z999CLGC STRUCTURE 20140405105506</td>
<td></td>
</tr>
<tr>
<td>Z999CRES STRUCTURE 20140405105506</td>
<td></td>
</tr>
<tr>
<td>Z999CSCH SCHEDULE 20140405105506</td>
<td></td>
</tr>
<tr>
<td>Config token</td>
<td>201404051055068A0345169672</td>
</tr>
<tr>
<td>Config version</td>
<td>01</td>
</tr>
</tbody>
</table>

Figure 20. INGAMS Command Dialog

You can also use the INGAMS command in line mode:

INGAMS DETAILS name OUTMODE=LINE

Where `name` is the name of the primary automation manager.
Identifying Current Configuration Files
Part 2. Using SA z/OS on the Host

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

- Chapter 5, “Getting Started,” on page 81
- Chapter 6, “How to Get Resource Information Sysplexwide,” on page 85
- Chapter 7, “How to Monitor and Control Resources,” on page 93
- Chapter 8, “Solving Problems with Resources,” on page 183
- Chapter 9, “Using I/O Operations,” on page 195
Chapter 5. Getting Started

Different sets of commands are available from host sessions:

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

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

- Fullscreen mode: If it is a command to display information and you just enter the command name, a panel is displayed showing all available resources. On this panel, you can specify further actions/commands for a special resource. If you enter a command to maintain a resource without further parameters, a fullscreen panel prompts you for more information.

- Line mode: From a command line, you can enter the complete syntax of a command to receive either the desired output directly or to manipulate the resource you wanted in the way you wanted.

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

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

Issuing Commands

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

- You can use command dialogs to enter command parameters and issue commands using 3270 panels. See “Using Command Dialogs” on page 82.

- You can issue a Take Action command on the Tivoli Enterprise Portal (TEP). You can do this either by defining a situation that includes a Take Action command that runs when the situation is triggered, or from the Take action view, which lets you enter a command or select a predefined command.

- You can use an object’s popup menu on the NMC workstation.

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

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

For information about how to use this graphical command interface, refer to Chapter 10, “Using the NetView Management Console for SA z/OS,” on page 203.
Issuing Commands

- You can enter I/O operations commands at the z/OS operator console. See “Using the z/OS Operator Console for I/O Operations Commands” on page 198.
- You can issue processor operations, system operations, and automation manager commands from the same interface via NetView terminals as described in “Using Command Dialogs.”
- I/O operations commands can be issued from ISPF dialog panels as described in “Using the I/O Operations ISPF Dialog” on page 198.

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

Using Command Dialogs

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

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

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

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

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

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

Command Logging

SA z/OS allows you to log the commands and their parameters that are issued by an operator. You can then use this log for audit purposes.

Several operator commands write an AOF705I message for each command execution. The AOF705I message lists all the parameters that have been specified together with the user ID of the operator or autotask that issued the command, for example:

AOF705I USER1 issued command INGREQ with MYAPL/APL/SAT1 REQ=START SCOPE=ONLY VERIFY=YES SOURCE=OPERATOR PRI=LOW PRECHECK=YES

AOF705I USER1 issued command INGIMS with IMS3CTL/APL/KEYC REQ=CMD CMD=/DIS
You can enable and disable command logging using the INGCNTL command, which sets the AOF_AAO_LOG_COMMAND advanced automation option (AAO), for example:

INGCNTL SET COMMAND_LOGGING=YES
Using Command Dialogs
Chapter 6. How to Get Resource Information Sysplexwide

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

SA z/OS provides commands that let you examine resources from the point of view of the automation manager or automation agent.

The automation manager decides how automation is performed for known resources in the sysplex. The automation manager decides how to handle the requests that it receives through the various channels (for example, the defined automation policy or interactive operator requests) using an internal hierarchy of priorities that is given in Table 2 on page 14.

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

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

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

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

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

Using INGLIST to View Resources

The INGLIST command provides details about resources in the sysplex from the point of view of the automation manager. It displays information about a resource, such as statuses, flags, and schedules. For more detailed information, see IBM Tivoli System Automation for z/OS Operator’s Commands.

On the INGLIST panel, you can:

- See more information about the resources by scrolling horizontally using PF11.
- Use PF9 (Refresh) to obtain a new set of data for the displayed subsystems.
- Use PF5 to invoke a filter dialog (as described in “Specifying Filters with INGFILT” on page 86), which is equivalent to using the INGFILT command that is available from any command line. With this filter, you can reduce the amount of information shown in the display. It enables you to setup a specific view for an operator. For example, you can set up a filter that shows only certain resources in specified observed and desired states.
• Launch other actions to maintain and control resources or groups.
• Use the CMD column on the left of the panel to issue various commands against any of the resources that are displayed.

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

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Update</td>
<td>B Start</td>
<td>C Stop</td>
<td>D INGRELS</td>
<td>E INGVOTE</td>
<td>F INGINFO</td>
<td>G Members</td>
</tr>
<tr>
<td>H</td>
<td>DISPTRG</td>
<td>I</td>
<td>INGSCHED</td>
<td>J INGROUP</td>
<td>K INGCICS</td>
<td>L INGIMS</td>
</tr>
<tr>
<td>M</td>
<td>DISPMTR</td>
<td>P</td>
<td>INGPAC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>INGTWS</td>
<td>U</td>
<td>User</td>
<td>X INGLKUP</td>
<td>/ scroll</td>
<td></td>
</tr>
</tbody>
</table>

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

Enter the letter corresponding to the command in the entry field next to the resource you want it issued against and press Enter. When you return from the command there may be a slight pause as the display refreshes itself.

• Use PF4 to toggle to the DISPSTAT command dialog, which gives you the automation agent view of the resources in the sysplex (see "Displaying the Automation Agent View with DISPSTAT" on page 88). To return to the INGLIST panel press the PF3 key on the DISPSTAT panel.

Pressing PF4 on the DISPSTAT panel displays the INGLIST panel for all resources that are shown on the DISPSTAT panel. However, this might not be the same as the original INGLIST display because the DISPSTAT display might have been modified due to filtering or suppressing of resources that the automation agent does not handle.

**Specifying Filters with INGFILT**

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

You can specify one or more of the following filters:

- Resources
- Observed status
- Desired status
- Automation status
- Compound status
- Health status
- Automation flag
- Category
- Subcategory
- Group type
- Jobname(s)
- Description
- RunToken(s)
- Pacing Gate(s)

You can abbreviate the status in the status lists, for example, av for Available. In order to be eligible for the display, a resource must match all filter criteria. An asterisk means that the filter is not set. A ‘^’ or ‘\’ in front of the status displays all resources whose status is not what you specified.
Use the PF4 key to clear the currently established filter settings. The filter will then contain an asterisk, meaning that the filter is not set, or a blank. Use the PF5 key to revert to the currently established filter settings.

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

Using INGINFO to View a Selected Resource

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

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

On the INGINFO panel, you can use PF4 to toggle to the DISPINFO command dialog, which gives you the automation agent point of view of the current resource (see "Using DISPINFO to Display Detailed Information" on page 90).

If the resource has pacing gates defined, you can use PF5 to toggle to the INGPAC command dialog, which gives you more details about the pacing gates.

Displaying Application Group Information

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

Displaying Monitor Information

The DISPMTR command displays information about monitors that you have defined using the customization dialog for your system and allows you to manage them (that is, resetting the health status of the resource that is being monitored to NORMAL, and starting and stopping monitors).

Displaying Relationship Information with INGRELS

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

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

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

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

You can change the level of detail that is displayed using the following keys:
+ Shows the condition details that exist for the selected relationship. This is only applicable if the relationship conditions were previously collapsed.
– Collapse the condition details. Only the relationship will be shown.

### Displaying the Automation Agent View with DISPSTAT

The DISPSTAT command displays fullscreen panels that show information for resources defined to the current automation agent, such as automation status, automation flag setting overrides, and automation status changes. For more detailed information, see *IBM Tivoli System Automation for z/OS Operator’s Commands*.

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

On the DISPSTAT panel, you can:

- Cycle through further information using the PF10 and PF11 keys. If you use a wide (132 column) display, all of the data can be displayed on two screens. On a smaller 80 column display there is a third screen that displays Description data.
- Use PF9 (Refresh) to obtain a new set of data for the displayed subsystems. Some of this information, especially the automation status, changes in real time.
- Use PF5 to invoke a filter dialog (as described in “Specifying Filters with DISPSFLT” on page 89), which is equivalent to using the DISPSFLT command that is available from any command line. With this filter, you can reduce the amount of information shown in the display. It enables you to setup a specific view for an operator. For example, you can set up a filter that shows only resources of a certain name or that are in a specific state.
- Use the CMD column on the left of the panel to issue various fullscreen commands against any of the resources that are displayed.

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

```
A dispflgs B setstate C ingreq-stop D thresholds E explain F info G tree H trigger I service J all children K children L all parents M parents
```

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

Enter the letter corresponding to the command in the entry field next to the resource you want it issued against and press Enter. When you return from the command there may be a slight pause as the display refreshes itself.

- Use PF4 to toggle to the INGLIST panel to see the same resources as shown in the DISPSTAT panel but from the automation manager point of view. To return to the DISPSTAT panel press the PF3 key on the INGLIST panel.
Pressing PF4 on the INGLIST panel displays the DISPSTAT panel for all resources shown in the INGLIST panel. However, this might not be the same as the original DISPSTAT display because the INGLIST display might have been modified due to filtering.

**Specifying Filters with DISPSFLT**

The DISPSFLT panel displays the filters that are used for the DISPSTAT display. The filters are stored in task global variables, and you can save and update them by pressing PF9.

When you invoke the panel from DISPSTAT, pressing Enter takes you to the DISPSTAT display using the filters that are currently displayed. The filters are not saved unless you have previously pressed PF9 in the DISPSFLT command dialog.

The filters that you can set are:

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

- **Statuses:**
  You can specify a list of automation statuses to be included or excluded from the display. This filter is used whenever the display is refreshed.
  SA z/OS provides the following sets of pre-grouped statuses, which can be abbreviated by their numbers:
  1. All resources that are currently UP or ENDED
  2. All resources that are in a normal status: DOWN, STARTED, UP, AUTOTERM, AUTODOWN, ACTIVE, ENDED, ENDING, RESTART, EXSTART, RUNNING
  3. All resources that are in a down status: DOWN, INACTIVE, RESTART, AUTODOWN, CTLDOWN, STOPPED, BROKEN, MOVED, FALLBACK
  4. All resources that are in a start transition state: RESTART, STARTED, STARTED2, ACTIVE, RUNNING, EXSTART
  5. All resources that are in a stop transition state: AUTOTERM, ENDING, STOPPING, ABENDING, BREAKING

- **Target Systems:**
  This lets you specify the systems within the sysplex that you want data gathered from. By default it shows just your local system.
  You can specify a list of one or more system, domain, or sysplex names (in any combination), and the query is sent to those systems. Alternatively you can specify ‘*ALL’, and the query is sent to all active systems within the sysplex.
  In general the more specific your query and the fewer systems you send it to, the faster it will execute.
  When you have updated the filters (the defaults will do for just exploring), press Enter to go into the DISPSTAT display.
Using DISPINFO to Display Detailed Information

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

The DISPINFO command displays information about the selected resource including many of the definitions that have been made to SA z/OS about the subsystem. You can use PF4 to toggle to the INGINFO command display for the selected resource, which shows the automation manager view of it.

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

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

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

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

AOF169I THE LAST LINE IS ALREADY DISPLAYED ON THE SCREEN

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

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

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

Using DISPTREE to Display Dependency Information

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

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

The panel uses reverse video to improve the highlighting. Non-subsystem resources are indicated with |.
PF4 will refocus the display onto the subsystem under the cursor. PF5 will invoke DISPINFO against the subsystem under the cursor. However, you cannot use DISPINFO on non-subsystem resources.

Displaying Statistical Information

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

The summary report lists information about, for example:
- The number of resources defined to NetView and managed by the automation agent
- The total number of messages automated
- The total number of resulting commands
- The total number of START and STOP commands issued
- The number of systems in the sysplex
- The total number of resources managed by the automation manager

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

For each resource the following information is shown:
- The number of messages automated
- The number of commands resulting from the message traps
- The number of START and STOP commands that were issued
- The number of critical threshold conditions that have occurred

You can select a resource to display details by entering S in the Cmd column.
Chapter 7. How to Monitor and Control Resources

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

In addition to the principal methods for controlling resources as introduced in Chapter 2, “Concepts,” on page 11, SA z/OS offers more sophisticated techniques to the operator.

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

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

Table 10 shows the various tasks that are discussed in this chapter along with the main commands that you can use to perform them. For more detailed information about the commands refer to IBM Tivoli System Automation for z/OS Operator’s Commands.

Table 10. Operator Tasks and Related Commands

<table>
<thead>
<tr>
<th>Task</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Starting and Stopping Resources” on page 94</td>
<td>INGREQ, INGVOTE</td>
</tr>
<tr>
<td>“Application Pacing” on page 96</td>
<td>INGPAC</td>
</tr>
<tr>
<td>“Using Runmodes” on page 99</td>
<td>INGRUN</td>
</tr>
<tr>
<td>Displaying history information, “Using INGHIST” on page 96</td>
<td>INGHIST</td>
</tr>
<tr>
<td>“Using Schedules” on page 105</td>
<td>INGSCHED</td>
</tr>
<tr>
<td>“Displaying and Setting Trigger Information” on page 107</td>
<td>DISPTRG, INGTRIG</td>
</tr>
<tr>
<td>“Displaying and Setting Events” on page 107</td>
<td>DISPEVT, DISPEVTS, INGEVENT</td>
</tr>
<tr>
<td>“Determining Automation Agent Flag Values” on page 108</td>
<td>DISPFGLS, DISPSCHD, DISPSTAT, INGAUTO</td>
</tr>
<tr>
<td>“Working with Application Groups” on page 110</td>
<td>INGGROUP, DISPAPG</td>
</tr>
<tr>
<td>“Moving Sysplex Application Groups” on page 114</td>
<td>INGMOVE</td>
</tr>
<tr>
<td>“Setting Timers” on page 115</td>
<td>SETTIMER</td>
</tr>
<tr>
<td>“Interacting with the Automation Manager” on page 115</td>
<td>INGAMS</td>
</tr>
<tr>
<td>“Changing the Automation Agent Status” on page 116</td>
<td>SETSTATE</td>
</tr>
<tr>
<td>“Monitoring Messages” on page 119</td>
<td>INGMSGS, INGNTFY, ISQCMON, ISQXMON</td>
</tr>
</tbody>
</table>
Table 10. Operator Tasks and Related Commands (continued)

<table>
<thead>
<tr>
<th>Task</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Monitoring Hardware” on page 122</td>
<td>ISQXDST</td>
</tr>
<tr>
<td>“Monitoring Health Performance” on page 123</td>
<td>DISPMTR, INGLIST</td>
</tr>
<tr>
<td>“Monitoring OMEGAMON Sessions” on page 125</td>
<td>INGSESS</td>
</tr>
<tr>
<td>“Monitoring Processor Status” on page 128</td>
<td>ISQXDST</td>
</tr>
<tr>
<td>“Monitoring with the Status Display Facility” on page 131</td>
<td>SDF</td>
</tr>
<tr>
<td>“Monitoring and Controlling a Sysplex” on page 142</td>
<td>INGPLEX, INGST</td>
</tr>
<tr>
<td>“Monitoring and Controlling Guest Machines and Logical Partitions” on page 160</td>
<td>ISQESUM, ISQCCMD</td>
</tr>
<tr>
<td>Monitoring and controlling the resources of a Blade Extension (zBX) in a zEnterprise ensemble environment on page 171</td>
<td>ISQSTART, ISQXIII, ISQECMD, ISQXDST</td>
</tr>
<tr>
<td>“Using the SA z/OS Discovery Library Adapter” on page 174</td>
<td>INGDLA</td>
</tr>
<tr>
<td>“Displaying Threshold Settings and Occurrences” on page 174</td>
<td>DISPASF, DISPERRS, INGTHRES</td>
</tr>
<tr>
<td>“Handling Jobs That Are Not Controlled by SA z/OS” on page 177</td>
<td>INGLKUP</td>
</tr>
<tr>
<td>“Communicating with Other Systems” on page 178</td>
<td>INGSEND, ISQSEND</td>
</tr>
</tbody>
</table>

**Starting and Stopping Resources**

In addition to goal driven automation that you exploit by defining automation policy, SA z/OS offers a series of commands that let an operator react to special circumstances beyond regular automation, including:

- INGREQ, see “Using INGREQ” on page 95
- INGVOTE, see “Using INGVOTE” on page 95
- INGHIST, see “Using INGHIST” on page 96

Each of these commands is documented with complete syntax and further sample output screens in _IBM Tivoli System Automation for z/OS Operator’s Commands_.

**Using INGREQ**

As described in “Automation Concepts” on page 11, commands issued with INGREQ and a high priority are the most powerful automation commands to make applications available or unavailable. They supersede all other requests that exist against a resource at that time.

You can use this command to:

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

A request is persistent until:

- You cancel it using the INGVOTE, INGREQ, or INGSET command
- It expires according to the specified EXPIRE parameter
Using INGREQ

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

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

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

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

Use PF11 to see additional parameters.

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

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

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

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

Using INGVOTE
Whenever you (or a program) want the automation manager to do something, you must send it a request. Requests are remembered within the automation manager. They are made against individual resources and then propagated across the relationship graph to other resources within the automation manager. If, for example, your request is to start resource A, and A has a relationship to B such that B needs to be started before A can run, the automation manager generates a vote to resource B to start it. Thus a request can be viewed as a black box of related votes that is generated by the automation manager.

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

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

- The name of resource the request was made against
- The request action
- The source of request
- The request priority
- The request vote
- Default desired status
- Runmode qualification

From the INGVOTE command dialog, you can enter various commands against a selected vote.

Using INGHIST

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

- limit the output if many work items exist within the specified time period
- specify the maximum number of work item records to be shown after work item expansion in fullscreen mode.

Time-interval search criteria are entered in local time. However, because the history data is stored using GMT time stamps, it is necessary for INGHIST to convert the local-time search criteria to GMT before retrieving the data. This may make it necessary for the operator to make a manual adjustment to the time-interval search criteria after a daylight-saving-time change has been set. The local-time-to-GMT conversion calculation will be one hour different after a daylight-saving-time change. Data written after a daylight-saving-time change will be retrieved as expected but data written before will be offset by one hour.

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

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

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

Application Pacing

The resource consumption of applications (that is, of type APL) during the start and stop phases can vary a lot. While many applications are started or stopped very quickly, other applications may consume a lot of CPU resources and may even tend to dominate how the system assigns CPU resources among the started tasks in need.
Application Pacing

To facilitate the management of those applications aiming for efficient resource utilization while avoiding customization efforts, the application pacing capability can be used to control how many applications of a kind can be started or stopped at the same time. For this, any application that is eligible to receive a start or stop order from the automation manager has to 'transit' through a defined Pacing Gate. If the number of applications that are currently in transition reaches the maximum concurrency level, additional applications are held back (waiting) until another application finished this transition and reached the final desired status or terminated during that transition.

In order to use application pacing, you need to define one or more Pacing Gate entries using the Customization Dialog, option 13 (PAC) and specify the concurrency level of starting and stopping applications. See Figure 21.

Next, you assign those applications that you want to more tightly control to a Pacing Gate. This is done by selecting a pacing gate using the PACING GATE policy. Note, each application can only be associated with one single Pacing Gate at a time.

Typically, application start-up and shutdown times can be minimized if you allow for maximum parallelism that the system can handle. Therefore, it is not recommended to use Pacing Gates as a general vehicle to control all applications. Using them for certain CPU intensive workloads, however, can help to decrease system IPL, shutdown times or both, as Pacing Gates can prevent too many workloads dominating the CPU resources during these times.

To see how Pacing Gates affect the applications associated with them, invoke the INGPAC command from the NetView console. INGPAC shows a list of Pacing Gates based on the selection criteria specified by the operator. It also allows you to display all applications that are currently either in transition or waiting for transition through the Pacing Gate. An example display is shown in Figure 22 on page 98.
Application Pacing

Refer to IBM Tivoli System Automation for z/OS Operator’s Commands for further details about using INGPAC to manage Pacing Gates.

Usage notes:
1. Performing a configuration refresh while resources are being started or stopped under the control of a Pacing Gate may cause the resources to be released immediately:
   - if you remove a resource from a Pacing Gate
   - if you remove a resource from a Pacing Gate and link it to another Pacing Gate

   For these reasons it is not recommended to perform a configuration refresh while resources are being started or stopped under the control of pacing gates.

2. While you may use Pacing Gates with externally started and stopped resources, be aware that SA z/OS can neither prevent nor delay an externally triggered action against a resource and releases resources as soon as it detects an upcoming external start or stop for them so that the agent does not consider the event to be unexpected.

   Recommended usage is just to put the resources that cause the external actions under the control of a Pacing Gate and to be aware that each one brings along a number of external address spaces when working out your pacing limits.

Monitoring for IPL Completion

A new attribute can be defined in Customization Dialog for entry types Application Group (APG), Application (APL), and Monitor Resource (MTR) to indicate that an IPL is only considered complete if the APGs, APLs and MTRs, defined with this attribute have reached the status of AVAILABLE.

If the attribute 'Monitor for IPL Complete' is set to YES for any APG, APL, MTR resources, additional processing is triggered from the System Automation initialization process. This process checks whether the selected Application...
Monitoring for IPL Completion

Group(s), Application(s) or Monitor Resource(s) have reached the status of AVAILABLE within the specified time limit, and the IPL is indicated as 'successfully completed'.

The attribute may be set to NO or blank, indicating no assessment is made for a resource's availability at IPL completion.

Within policy AUTOMATION OPTIONS of entry type System Defaults (SDF), the time limit is expressed as 'IPL Complete Time Limit' (blank or hh:mm:ss). This attribute allows the specification for the maximum time the resources are allowed to take before reaching the status of AVAILABLE, before the IPL completion as 'over'.

The following messages become available at the console depending on the YES setting(s) you have made for your chosen resource(s):

- ING313I indicates the relevant APG/APL/MTR resources are available and the IPL is complete
- ING314I indicates the relevant APG/APL/MTR resources are not available although specified and the IPL is therefore not complete.
- ING315I indicates that an IPL is complete after the specified time limit, as an optional message.

Refer to the advanced automation variable AOF_AAO_IPL_COMPLETE_MESSAGE in IBM Tivoli System Automation for z/OS Customizing and Programming for additional information on how to customize this function's behavior.

Using Runmodes

Runmodes is a flexible way to control the availability of resources without the need to place explicit START or STOP requests against them or manipulating their automation flag. You can use runmodes to:

- start up your system in two or more stages. This brings up a subset of your resources, do some checks, bring up another set of resources, do some more checks and finally bring up the complete system. Note that you can also shut down from an intermediate stage.
- shut down your system in two or more stages. That is shut down a subset of your resources, apply maintenance, shut down another set of resources, apply some more maintenance and finally shut down the complete system. Note that you can also bring up your system from an intermediate stage.
- switch between two or more scenarios with the same configuration. For instance, you can switch between "Weekday" and "Weekend" or "Normal" and "disaster recovery".

In order to use runmodes, prepare the appropriate definitions in the Customization Dialogs (refer to IBM Tivoli System Automation for z/OS Defining Automation Policy). Valid runmodes must be defined for each system. Each runmode has a name and lists so-called "runtokens". Resources like APLs and MTRs also list runtokens. For APGs and system resources, the runtokens are automatically determined by the runtokens and the runtokens of the current runmode. If at least one runtoken is common to both, the resource qualifies for that runmode. A resource also qualifies if it has no runtokens at all and the current runmode contains the special runtoken *NULL. If no runmode is set, all resources qualify as in earlier releases.
A runmode can be set at initialization time as a reply to message AOF603D or by means of the INGRUN command (refer to IBM Tivoli System Automation for z/OS Operator’s Commands). INGRUN additionally allows the forcing of runmode qualification for resources regardless of their runtokens. If such a resource is a group, all members inherit the qualification.

A resource may be qualified for one of the three following reasons:

- qualification is forced for this resource (see Setting Runmodes or Forcing a Runmode Qualification using INGRUN on page 104)
- resource is a member of group forced to qualify
- at least one of its runtokens matches a runtoken designated for the current runmode.

Internally, setting a runmode means injecting a special STOP request against the sysname/SY6/sysname resource. This STOP request is generated with SOURCE=INGRUN and will therefore be of request method DEFAULT. STOP votes are propagated to all resources of that system. The automation manager then decides according to the above rules whether the STOP vote can be winning or not. All rules for STOP requests apply also to the runmode request. Especially, there can be only one runmode request per system.

**Note:** When a resource is (forced) qualified for a runmode, this does not necessarily mean that it is started. There might be other factors that keep the resource down like the default desired status, unsatisfied relationships or other existing stop votes. So if you want to start a resource outside of its runmode scope you may also consider using an appropriate START request.

Runmode requests are persistent as long as you do a HOT start of the automation manager. This means that the request stays in the system until it is explicitly cancelled which is done by setting the runmode to *ALL. The request does not change when the configuration is changed unless you place a new runmode request. It also means that the decision of the automation manager can be overruled by injecting START or STOP requests with the appropriate priority.

### Defining Runmodes and RunTokens

Runmodes are defined in the customization dialogs using the Entry Type UET (User E-T pairs). The runmode definition includes the runtoken(s) assigned to each runmode. The entry-type pair INGRUN MODE must be included.

<table>
<thead>
<tr>
<th>Cmd</th>
<th>Keyword</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>___</td>
<td>BASIC</td>
<td>(TBASE1 TBASE2)</td>
</tr>
<tr>
<td>___</td>
<td>DISASTER_RECOVERY</td>
<td>(TDISASTER1 TDISASTER2 TDISASTER3)</td>
</tr>
</tbody>
</table>

The column **Keyword** specifies the name of the runmode and can be up to 20 alphanumeric characters. Additionally _ and ? is allowed.

The column **Data** specifies the runtokens included in this runmode and must be a list of runtokens, up to 20 characters long consisting of alphanumeric characters ? and _. Additionally a special runtoken *NULL is allowed to qualify all resources which have no runtokens specified.

You can link a User Entry-Type to one or more systems to make runmodes available there.
Note: If you link more than one User Enty-Type containing INGRUN-MODE to a single system, the data is not merged. Instead the last one read takes effect.

Assigning Runtokens
Define runtokens in the customization dialog for entry types APL and MTR within the APPLICATION and MONITOR INFO policy respectively. The Runtokens field is available on these policy items.

| Runtokens........ TBASIC1 TDISASTER1 |

The Runtokens field allows you to specify a list of tokens, up to 20 characters long, consisting of alphanumeric characters, ? and _.

Displaying Valid Runmodes
Display runmodes by using the DISPSYS command for the current system. This section is only displayed if either the current runmode is not *ALL or if there are runmodes defined for the system.

| CFG Token : 20111011163620FF066AAD2094 |
| Runmode(s)          |
| Current : TESTPRODW0 |
| Defined             |
| TESTBASIC : TB      |
| TESTPRODW0 : TB WD  |
| TESTPRODWE : TB WE  |
| TESTPRODREC : TB WD WE RE |
| Message automation table(s) |
| DSITBL01 SWITCHTAB INGMSGSA INGMSG01 INGMSGHW |
| Flags               |
| Automation : Yes    |
| Init Start : Yes    |

Figure 23. DISPSYS output for Runmodes

Current shows the runmode currently set. If no runmode is set, it shows *ALL.

Defined shows the list of runmodes defined for the system together with their runtokens.

Note: The complete section "Runmode(s)" is displayed only when runmodes are used.

Displaying Runtokens
Display runtokens for a resource by using the INGINFO command:
Using Runmodes

The Qualification field shows how the resource qualifies for the runmode:

- If a Token is listed at least one runtoken matches a runtoken of the current runmode.
- If a Resource is listed, runmode qualification has been forced for this resource.
- If this resource is qualified because it is a member of groups that are forced to be qualified, the number and names of the groups are displayed.

Runtokens show the runtokens defined for the resource. If the resource is a group, it lists the set union of runtokens of all members.

Note: The complete section Runmode is displayed only when runmodes are used.

Displaying Runmode Qualifications

Display runtokens for a resource by using the INGLIST command:

```
Figure 24. INGINFO output for Runtokens
```

```
<table>
<thead>
<tr>
<th>CMD</th>
<th>Name</th>
<th>Type</th>
<th>System</th>
<th>Category</th>
<th>Subcategory</th>
<th>Jobname</th>
<th>Qual</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM</td>
<td>APL</td>
<td>AOC4</td>
<td>AMA4</td>
<td>--T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AM</td>
<td>APL</td>
<td>AOC5</td>
<td>AMA4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AM_X</td>
<td>APG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AM2</td>
<td>APL</td>
<td>AOC4</td>
<td>AMS4</td>
<td>--G</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AM2</td>
<td>APL</td>
<td>AOC5</td>
<td>AMS5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AOC4</td>
<td>SYG</td>
<td>AOC4</td>
<td>--T</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AOC4</td>
<td>SYS</td>
<td>AOC4</td>
<td>--T</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AOC5</td>
<td>SYG</td>
<td>AOC5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AOC5</td>
<td>SYS</td>
<td>AOC5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>APPC</td>
<td>APL</td>
<td>AOC4</td>
<td>APPC</td>
<td>--R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>APPC</td>
<td>APL</td>
<td>AOC5</td>
<td>APPC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASCH</td>
<td>APL</td>
<td>AOC4</td>
<td>ASCH</td>
<td>--T</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

```
Figure 25. INGLIST output for Runmode qualifications
```

The Qual column shows runmode qualification for each resource listed as the three characters:

- **G** Forced by Group
- **R** Forced by Resource
Using Runmodes

T Runtoken match
- attribute does not apply

Note: If runmodes are not used, blanks are displayed.

Runmode qualification shows runmode qualification for each resource listed as the three characters G, R, T or -. (See "Displaying Runmode Qualifications" on page 102.)

Note: This field is only displayed when runmodes are used.

Displaying Runmode Requests or Votes

Runmode requests or votes for a resource can be displayed via the INGVOTE command:

```
Cmd: C Cancel request K Kill request S Show details V Show votes
Cmd Name Type System Request Data
--- ----------- ---- -------- ------ ------------------------------------------
AOCA SYG AOCA Req : MakeUnAvailable
Run : Mode=NIGHT_SHIFT Tokens=NIGHTSHIFT1 NIGHTSHIFT2 ...
Org : INGRUN(AUTO1)
Pri : 01220000 Should Be Down - Default
Stat: Pending
PSVPEVNT SVP Req : MakeUnAvailable
At : 2014-10-19 10:00:00
Org : SCHEDULE
Pri : 01120000 Should Be Down - Schedule
Stat: Winning
PSVPH073 SVP Req : MakeUnAvailable
```

Figure 26. INGVOTE output for Runmode requests or votes

Select S to show more details.

```
Resource. . : AOCA/SYG/AOCA
Originator. : INGRUN(AUTO1)
---------------------------------------------------------------
Request priority : 01220000 Should Be Down - Default
Request status : Pending
STOP type : NORM
Runmode : NIGHT_SHIFT
Runtokens : NIGHTSHIFT1 NIGHTSHIFT2_TOKENALL
AgentParms : STATE=AUTODOWN APPLAPRMS***
Restart : No
Override : No
```

Figure 27. Detail display for INGVOTE command results

The Runmode and Runtokens fields show the runmode and its runtokens if this is a runmode request.
Using Runmodes

Note: There can only be one runmode request at a time.

Setting Runmodes at SA z/OS Initialization Time

At SA z/OS initialization time, you can set a runmode as a reply to message AOF603D. Message AOF767I offers you the following choices.

AOF767I AUTOMATION OPTIONS:
- STOP - CANCEL AUTOMATION
- PAUSE - SUSPEND AUTOMATION
- NOSTART - DO NOT AUTOMATE SUBSYSTEM STARTUP
- RUNMODE=x - SET RUNMODE (CURRENTLY *ALL)
- ENTER - CONTINUE

If you are using runmodes in your configuration you see the runmode currently set. You can reply RUNMODE=x where x is a valid runmode defined on the current system. You can enter RUNMODE=? to see a list of valid runmodes:

AOF779I VALID RUNMODES ARE:
- *ALL
- BASIC = TBASIC1 TBASIC2
- DISASTER_RECOVERY = TDISASTER1 TDISASTER2

Setting Runmodes or Forcing a Runmode Qualification using INGRUN

Runmodes are also set using the INGRUN command:

![INGRUN command dialog](image)

In order to set a runmode, use the SET request option, enter a system name and a valid runmode for the system and press ENTER. The INGREQ verification panel should appear. Pressing PF10 will then inject the runmode request.

Forcing Runmode Qualifications

In order to force a runmode (for a system/sysplex), use the ADD request option and a resource name. The resource can be an incomplete specification and it can also contain extended wildcards. You will be presented with a selection panel where you must mark the resource(s) to process and then press PF10 to execute finally the command. This is the case even if only one resource matches your specification. You can see this as a verification panel.
Using Runmodes

On the selection panel you can enter the line command SEL pattern to select easily any multiple resources. Specify SEL * to select all.

Forced runmode qualification is persistently stored in the automation manager. However, this should be an exception and you should remove unnecessary forced runmode qualifications using INGRUN.

```
Request => ADD Request type (SET, ADD or DEL)
Target => System name, domain ID or sysplex name
--- Parameters for SET request --------------------------------------------
System => System name
Runmode => Runmode name (mode or ?)
Persistent => Keep request across IPL (YES/NO)
Type => Type of processing (NORM/IMMED/FORCE)
Priority => Priority of request (FORCE/HIGH/LOW)
Override => (ALL/NO/TRG/FLG/DPY/STS/UOW/INIT)
Verify => Check affected resources (YES/NO/WTOR)
Comment =>

--- Parameters for ADD or DEL requests ------------------------------------
Resource => MYAPL1/APL/SYS1 format: name/type/system
```

*Figure 29. INGRUN command dialog (for ADD request type)*

In order to delete the forced runmode qualification specify request DEL and a resource name. The resource name can be an incomplete specification and it can also contain extended wildcards. You will be presented with a selection panel as for the ADD request. You also specify a question mark (?) which lists all resources previously processed by an ADD request.

Using UP Status Delay

SA z/OS recognizes that applications reach an UP status by trapping a specific message that indicates the application is running.

There are some applications that do not issue an appropriately-timed message. In these cases, SA z/OS is instructed to trigger the resource’s transition to UP from a message that is issued earlier in the start sequence. However, in some situations there can be problems when commands for the dependent applications are sent while the parent application is still initializing.

The UP Status Delay setting in the application’s APPLICATION INFO policy adds a pause between the processing of the message that indicates that the resource is up and the transition of the resource into an UP status. The effect is to delay the processing of the dependent resources, giving the application time to finish initializing.

Using Schedules

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

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

A Using this action code to view the From-To time windows of the selected service period for the week. No updates are possible on this panel.

B Displays the actual From-To time windows for the selected schedule including the overrides, starting from the requested day (initially it is the current day). Use this option if you want to override a schedule for all resources that use this schedule (that is, for all resources that this schedule is linked to in the customization dialog). Such an override is called a schedule override and is described in "Schedule Override."

C Displays the resources that are using the selected schedule. You can select a resource to view the schedule that is used for this resource. You can override the schedule just for the selected resource. Such an override is called a resource override and is described in "Resource Override."

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

Schedule Override

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

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

Resource Override

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

There are three ways to reach the resource override panel:

- Use action code C against a schedule on the INGSCHED panel and then, from the resource list that is displayed, select the resource that you want to override the schedule for.
- Issue the INGSCHED command with the resource name as a parameter, for example:
  INGSCHED AGFG2/APL/AOCA
- From the INGLIST command dialog (see "Using INGLIST to View Resources" on page 85), issue action code I against a resource.

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

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

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

Displaying and Setting Trigger Information

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

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

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

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

```
DISPTRG BATB000P100
```

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

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

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

Displaying and Setting Events

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

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

DISPEVT
You can reach the DISPEVT command dialog by issuing the DISPEVT event_name command or using the command code D from the DISPTRG trigger condition list.

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

• The name of the resource
• The resource type
Displaying and Setting Events

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

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

From this command dialog, you can issue the following action codes against an event:
- **D** to show related resources (that is, it invokes DISPEVT for the event)
- **S** to set or unset the event (that is, it invokes INGEVENT for the event)

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

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

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

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

Successful completion produces the following message:

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

---

**Determining Automation Agent Flag Values**

Automation flags can be predefined in the customization dialogs or can be set during run time. Their meanings are explained in detail in "SA z/OS Automation Flags" on page 39.

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

Each flag has one of the following values:

- **Y**: The flag is turned on.
- **N**: The flag is turned off.
- **E**: The value of the automation flag depends upon the values returned by one or more user exit and will be determined when the value is required.
- **L**: Resulting commands or replies are logged only.
Displaying Actual and Effective Automation Agent Flags

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

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

The following command codes are supported:

A Allows you to define automation flags for a resource using the same flag settings as the selected resource. Another panel is displayed where you can specify the resource name and optionally overtype the flag settings.

C Allows you to modify the automation flags of the selected resource. Another panel is displayed showing the current flag settings where you can overtype the flag values.

R Causes to reset the automation flags to the values specified in the automation configuration file (ACF).

S Shows the scheduled override settings of the automation flags for the selected resource. You can specify scheduled overrides using the customization dialog. These are specific times when the automation is turned off for a particular flag and resource.

Displaying Automation Agent Flag Override Settings

The customization dialog allows you to turn automation flags off for a scheduled time period. You can view these time period settings by issuing the DISPSCHD command, which leads to the DISPSCHD command dialog panel.
- The System field shows the name of the system where the resource is defined.
- The Resource field shows the name of the resource.
- The Fl (flag) field shows an abbreviation of the automation flag name.
- The Day field shows the day of the week that automation is turned off.
- The Start and End fields show the time periods that automation is turned off.

Setting Automation On or Off: INGAUTO

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

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

Enter INGAUTO on any command line to reach the INGAUTO command dialog panel.
Determining Automation Agent Flag Values

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

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

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

**Working with Application Groups**

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

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

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

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

There are two types of application groups:

- **System Application Groups**: Application groups that are associated with
  particular systems.

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

Groups can also have one of three different **natures** (see Figure 30 on page 111):

**BASIC**

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

**MOVE**

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

**SERVER**

In a SERVER application group any number of components can be specified
Working with Application Groups

that must be available before the group is considered to be available. With SERVER application groups you can specify what should happen if their components fail to start.

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

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

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

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

Using INGGROUP

This section introduces the functions that are provided by the INGGROUP, gives details of application move, and describes the options you can use with the INGGROUP command dialog.

Functions

The functions provided by the INGGROUP command are:

- Automatic group management to maintain application availability. This includes the activation of backup or alternate resources in response to resource or system failure.
- A superset of the behavior provided by Automatic Restart Manager. Although INGGROUP does not provide a true cross-system restart, it is able to take recovery actions in a variety of situations where Automatic Restart Manager is not. These include application failure, supporting application failure, and 'total active instance' based management (for SERVER groups).
- The ability to move resources away from a system, on either a group or subgroup level or a system level.
Working with Application Groups

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

Application Move

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

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

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

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

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

For more information about application move see “Controlling Application Move and Server Management” on page 28.

INGGROUP Options

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

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

- The target sysplex for the command via the **System** field. If omitted, this defaults to the local system/sysplex. Specify the system name only when viewing or changing the policy of an application group that is not part of the local sysplex.
- The policy change that you want to make through the **Action** field.
- Next you must specify either the name of the target group or a set of templates that will match one or more groups. All policy items work with a list of group
name templates and apply to all Move and Server groups matching those names. While applying a policy to a Basic group is not an error, it will not achieve anything either.

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

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

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

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

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

- With the ADJUST option you can indicate that you want to temporarily adjust a group’s policy.

- With the RESET option you can set the preference value assigned to each member of the group to its initial value.

- With the DEFAULT option you can set the availability target, satisfactory target, group passive attribute and the member’s preference value to the value defined in the policy.

- With the OVERRIDES option you can display any overrides that have been made with the INGGROUP command for the specified resource groups.

- With the POLICY option you can display the policies for the specified resource groups.

- With the RECYCLE-CANCEL option you can initiate or halt a rolling recycle of the specified groups. This affects Server and Move groups that are within the scope of the command. Use the CHUNK parameter to recycle group members in parallel, improving performance.

Note that if the systems in a Server group all have the same preference value, the system that is active after a rolling recycle has completed may not be the same one as before the rolling recycle. Thus, triggering a rolling recycle always results in a move occurring. If you set the preference of the original system more than 250 points higher than that of any other system, the rolling recycle is followed by a second move to return the application to that system. This leads to an increase in the application down time.
INGMOVE makes moving sysplex application groups easier. Rather than manipulating the preference value of each member in the sysplex application group, you simply specify where the group should be moved to. In a sysplex application group of type MOVE (referred to as a move group) only one member is active at a time. By specifying the new location of the move group, the active member is terminated and the member associated with the new location is activated.

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

You can also specify a WAIT period, which is the number of seconds to wait before reporting that a timeout occurred if the automation manager does not provide the requested data. The maximum time interval is 999 seconds and, if it is not specified, it defaults to 30 seconds.

On the INGMOVE panel, the **Group name** column shows the name of the move group. Because the move groups are unique within a sysplex, only the first part of the resource group name is shown.

The **Obs Status** column shows the observed status of the move group as seen by the automation manager.

The **Systems** columns show a list of the systems where a member of the MOVE group exists and could run. Different colors are used to indicate the state of the system or the member of the group that is running on that system, as follows:
- Red if the system is down
- Yellow if the member of the group that is running on that system is not startable
- Green if a member of the group is currently active on that system
- Pink if the system is excluded
- White if the system is avoided
- Turquoise in all other cases

Systems that are underlined have the highest base preference value and are called the home system (or systems).

The system that has the highest actual preference is shown in reverse video, unless it is a home system. Note that this can be more than one system.

The **Move to** column is an input field where you can specify the system that the group should be moved to. Moving a sysplex application move group means terminating the member that is currently active and starting the member on the designated system. You can designate the system by specifying:
- The name of the system in the input field.
- An asterisk (*). This means that the group is moved back to its home system. If more than one home system exists, it is moved to the 1st one in the list.
- An equals sign (=). This is a shorthand that means to take the value from the nearest field above with a value specified in it.
Setting Timers

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

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

### Table 11. Setting Timers

<table>
<thead>
<tr>
<th>Task</th>
<th>Action on the SETTIMER command dialog panel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a timer</td>
<td>Type add at the command line and press Enter. On the Add/Change Timer panel that appears fill in the fields as required. <strong>Note:</strong> If you want to add a new timer but use most of the same values as a timer that has already been defined, type A in the CMD column and modify the SETTIMER panel.</td>
</tr>
<tr>
<td>Modify a timer</td>
<td>Enter C in the CMD column for the timer that you want to change. The Add/Change Timer panel is then displayed where you can modify the settings of your selected timer.</td>
</tr>
<tr>
<td>Delete a timer</td>
<td>Type D in the CMD column for the timer you want to delete and press Enter.</td>
</tr>
<tr>
<td>Suspend a timer</td>
<td>Enter U in the CMD column for the timer that you want to suspend. <strong>Note:</strong> The timer is deleted from NetView but all its relevant data is kept in common global variables. Only CHRON timers can be suspended.</td>
</tr>
<tr>
<td>Resume a timer</td>
<td>Enter R in the CMD column for the suspended timer that you want to resume (or reactivate).</td>
</tr>
</tbody>
</table>

Interacting with the Automation Manager

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

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

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

- **AGENT**
  - An automation agent
- **PAM**
  - A primary automation manager
- **SAM**
  - A secondary automation manager
Changing the Automation Agent Status

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

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

System operations defines 24 different statuses for automation, described in "Automation Agent Statuses" on page 48. For details about these statuses, see the help panels or IBM Tivoli System Automation for z/OS Customizing and Programming. You can change the automation statuses with the SETSTATE command (described in the IBM Tivoli System Automation for z/OS Operator’s Commands).

The SETSTATE command is used to change the automation status of a subsystem or a group of subsystems (supporting and dependent resources). The status change you are able to make depends upon the subsystem’s application monitor status:

- If the subsystem's application monitor status is ACTIVE, you can change its automation status to UP.
- If its application monitor status is INACTIVE, you can change it to RESTART or CTLDOWN. You may also be able to change it to MOVED or FALLBACK, depending upon its system associations.

With the SETSTATE command dialog, you can change the automation agent status of a subsystem or a group of subsystems with the Scope field.

Monitoring of Resources

With SA z/OS you can monitor various aspects of your resources, including:

- Messages, see "Message Capturing" and "Monitoring Messages" on page 119
- Hardware, see "Monitoring Hardware" on page 122
- Health performance, see "Monitoring Health Performance" on page 123
- OMEGAMON sessions, see "Monitoring OMEGAMON Sessions" on page 125
- Processor status, see "Monitoring Processor Status" on page 128

Message Capturing

SA z/OS automation is largely message driven. Messages that pass all the filters (MPFLSxx, NetView Message Revision Table) end up in the NetView automation table containing action rules. In certain situations, however, this is not sufficient. Instead, it is necessary that the messages that represent such situations are retained and presented to the operations team. The messages have to be retained until the situation is resolved, in which case they can be deleted automatically, using SA z/OS, or manually.

For this, SA z/OS allows you to capture messages and view the messages in the context of the automation resource that issued the message. For applications (APL), the captured messages are shown by the DISINFO command. Captured messages
associated with monitor resources (MTR) are shown by the DISPMTR command, and messages captured for an MVS component (MVC) are shown by the DISPSYS command. Refer to the IBM Tivoli System Automation for z/OS Operator’s Guide for further details about using those commands.

Furthermore, SA z/OS allows you to assign a severity to the captured messages. Based on the severity it can relay such messages to other targets. See “Determine Where Captured Messages Are Sent” on page 118 for more information.

SA z/OS distinguishes the following severities:
- IGNORE
- NORMAL
- UNUSUAL
- IMPORTANT
- CRITICAL

Messages with a severity of UNUSUAL, IMPORTANT, or CRITICAL are referred to as exceptional messages. Such messages are also (in addition to the commands mentioned above) displayed by the INGMSGS command. Exceptional messages are retained until they are deleted automatically or explicitly by the operator. Refer to IBM Tivoli System Automation for z/OS Operator’s Guide for further details about using INGMSGS to display and delete exceptional messages.

SA z/OS allows you to limit the number of messages that should be captured on an individual automation resource level (APL, MTR), or by using MVS component defaults (MDF), system defaults (SDF) or application defaults (ADF). Refer to IBM Tivoli System Automation for z/OS Defining Automation Policy for setting the Captured Messages Limit.

Exceptional messages have their own limit that can be specified by setting the Exceptional Messages Limit field in the system defaults (SDF). Refer to IBM Tivoli System Automation for z/OS Defining Automation Policy for further details.

The following subsections describe how to capture messages and how to work with these messages in more detail.

**Capture Messages Using Customization Dialog**

Messages of interest can be captured by making definitions in the policy using the Customization Dialog. For applications (APL), monitor resources (MTR) and MVS components (MVC) you can list a message that you want to capture in the MESSAGES/USER DATA policy and select Capture on the AT Status option panel (AS) to capture it. Messages captured this way will always have a severity of UNUSUAL.

To set the severity of a captured message to a value other than UNUSUAL, you add pseudo message CAPMSGS to the list of messages as well and use the code processing. In the code match table for pseudo message CAPMSGS you list all the messages for that APL, MTR or MVC for which you want to explicitly set the severity. The table below lists the meaning of the Code 1, Code 2 and Code 3 entries. Code processing works from top to bottom and ends at the first row that matches. If there is a match, the severity is used specified under the Value Returned field.
Message Capturing

Table 12. Code Entry Meanings

<table>
<thead>
<tr>
<th>Code 1</th>
<th>Code 2</th>
<th>Code 3</th>
<th>Code 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message ID</td>
<td>Jobname</td>
<td>optional, user-defined (see AOFCPMSG)</td>
<td>Severity</td>
</tr>
</tbody>
</table>

Refer to the MESSAGES/USER DATA Policy Item in *IBM Tivoli System Automation for z/OS Defining Automation Policy* for more details about defining message entries, capturing them with the AT Status option and how to use code processing in the Customization Dialog.

Capture Messages Explicitly

If you want to capture messages yourself, for example in your own Automation Table entry or within a user-defined script, you can use the AOFCPMSG command. Refer to AOFCPMSG in *IBM Tivoli System Automation for z/OS Programmer’s Reference* for details about the usage of that command.

Messages Captured by Default

SA z/OS also captures certain messages by default. This happens when:

- a threshold (infrequent, frequent or critical) has been reached for a command, a subsystem or a minor resource,
- commands fail with a return code > 0 and return code checking is turned on,
- subsystem automation status changes,
- WTORs are retained for reply processing,
- CICS transactions are recovered and for short-on-storage conditions,
- IMS transaction or program abnormally end,
- DB2 recovery takes place.

In all of these cases, SA z/OS determines the severity based on the actual condition detected. If you do not want SA z/OS to capture messages by default, you can create a CAPMSGS pseudo message entry in the policy as described above and assign the severity IGNORE to those messages you are not interested in for capture. The easiest way to do this would be a last line in the code match table that contains asterisks for Code 1, Code 2 and Code 3 and the Value Returned being set to IGNORE.

Determine Where Captured Messages Are Sent

If you want to present captured messages not only on the NetView panels displayed by the commands described above, you can use the Inform List field in the policy to specify one or more targets that the message can be relayed to. This field can be specified by individual resource level (APL, MTR) or in MVS component (MDF) or system (SDF) defaults.

When the Inform List contains:

- **SDF** or **NMC**, exceptional messages are forwarded to the Status Display Facility or the NetView Management Console
- **ITM**, all captured messages are forwarded to the SA z/OS Monitoring Agent and can be displayed on the Tivoli Enterprise Portal, if a situation is defined
• EIF, IOM, USR or TTT, all exceptional messages with a severity of CRITICAL can be transformed into particular alerts and forwarded to the corresponding target based on INGALERT settings.

Refer to [Inform List concepts](#) on page 63 for further details about forwarding automation resource information to any of the supported targets.

### Monitoring Messages

Message suppression facilities at the target system determine the amount of message traffic sent from a target system sent to a console. Messages that do not normally require operator intervention are usually suppressed at the target system. Messages that do normally require operator intervention may also be suppressed at the target system as automation programming is developed to respond for you.

You can monitor messages at the SA z/OS workstation via several mechanisms and at several levels. You can view:

- No messages
- Exceptional messages
- Messages that have been sent to a specific system console or operator console
- Messages that are related to the control of processor operations-managed resources
- SA z/OS messages (specified by message class)

### How to Display Outstanding WTOR Messages

WTOR (Write to Operator with Reply) messages are z/OS messages that require operator action. These messages can be viewed on the z/OS operator console.

### How to Display Exceptional Messages

An exceptional message is a message whose severity is either Unusual, Important or Critical. You use the INGMSGS command to display exceptional messages. From the INGMSGS panel you can either delete the messages or view details about the resource that is associated with the message (that is, by invoking INGINFO for it).

By default the messages are displayed on the INGMSGS panel in chronological order (the oldest message is shown at the top of the panel) but can be rearranged in any other order, for example, by severity with the SORT subcommand. The individual messages are colored depending on their severity. The color attribute is defined via SDF status definitions.

You can also display exceptional messages for one or more resources by specifying them with the INGMSGS command.

For more details about the INGMSGS command see [IBM Tivoli System Automation for z/OS Operator’s Commands](#)

### How to Display Messages for a Specific System or Operator Console

You can choose to receive all messages for a specific target system operator console using the Processor Operations Interested Operator List. These messages are displayed at your NetView console session.

Messages from the target console that are displayed include new message lines displayed on the Support Element console integration (CI) window, sent by z/OS,
Monitoring Messages

OS/390, VM, Linux on System z, VSE, or the Coupling Facility Control Code (CFCC). Multiline messages issued by z/OS look like several single-line messages from the console. Similarly, if an z/OS message extends over two lines, it looks like multiple separate messages from the processor operations console.

When you are monitoring a CI, make sure it is enabled to receive regular messages. For z/OS, refer to the VARY CONSOLE command parameters ACTIVE, DEACTIVATE.

All processor operations messages from a target system begin with the processor operations identifier ISQ900I or ISQ901I plus the target system name and the console type. Messages prefixed by ISQ900I are not displayed but are used by the automation. User automation should key on the ISQ900I messages. Messages prefixed by ISQ901I are sent to interested operators and should not be used for user automation. Multiline messages appear as multiple messages.

Joining or Leaving an Interested Operator List:
For target system operator console messages from CI use the ISQXMON command. For general processor operations resource control information messages use the ISQCMON command on a NetView console.

Viewing the List of Interested Operators:
The processor operations status facility contains status panels showing the list of interested operators for a specific target system.

You cannot directly access the Processor Operations Interested Operator List panel with the ISQXDST command. You can access the Processor Operations Interested Operator List panel only from the following status panels:
- Target System Summary panel.
- Target Hardware Summary panel. This is accessible only from other status panels.

The best way to access the Processor Operations Interested Operator List panel is to issue the following command to access the Target System Summary panel for a specific target system:

```
ISQXDST target_system_name
```

The resulting status panel shows the communication path status for the connection between the processor operations focal point and the target hardware (CPC) where the target system image is running (LPAR or Basic Mode). Place the cursor on the communication status line and press the PF7 key to see the associated Processor Operations Interested Operator List panel. If your operator identifier (the name you use to log on to NetView) is in this list, you should be receiving messages from that path. All other operators whose names are in the list also receive messages from that path.

You check the interested operator list for processor operations resource control messages (the ISQCMON list) by issuing the following NetView command:

```
LIST ASSIGN=GROUP
```

The interested operator list for processor operations is in group +TSCFMON.

How to See SA z/OS Messages Specified by Message Class
Notification operators are defined during customization of SA z/OS using the SA z/OS customization dialog. A notification operator is a NetView console
Monitoring Messages

operator who is authorized to receive messages from SA z/OS. The classes of messages that a notification operator is authorized to receive are specified at the time a notification operator is defined. See the tasks below to understand how to display information about a specific notification operator, how to add and delete notification operators, and how to change the messages that a notification operator receives.

Automation manager messages can be received automatically.

Also, you can receive messages from all systems within the sysplex.

How to Get Notify Messages: INGNTFY
This section provides information about notification operators and those messages that notification operators should be notified of.

Display Settings for a Notification Operator:
If you specify the INGNTFY command without a parameter, the main INGNTFY panel is displayed, showing a list of all the notification operators that have been defined. From this panel you can:

- Add (A) a notify operator using the settings of the selected operator
- Show or change settings (C) for the selected operator
- Delete (D) the notify operator
- Turn off (O) the sending of notify messages to the selected operator

For further information on the INGNTFY command refer to the IBM Tivoli System Automation for z/OS Operator’s Commands.

Query a Specific Notification Operator:
To display information about a specific notification operator, type INGNTFY on any command line and type query into the Status/Action field. Also, type the ID of the operator you want information about.

Turn On or Off all Notification Messages:
Use INGNTFY OFF on a NetView or operator console to temporarily turn off messages when you do not want to see them. Use INGNTFY ON on a NetView or operator console to turn the messages back on when you want to see them again.

Note: Each Notify Operator is issued a copy of the originating message. Consequently, if this message is being automated (it has an entry in the NetView automation table) this automation is replicated for each Notify Operator.

Add a Notification Operator:
You can add more notification operators through the INGNTFY command dialog. Remember that this operator will only be able to receive messages until the next time the automation control file is loaded, when all changes made using the operator interface are lost.

To add a notification operator, enter INGNTFY on any command line. Type add in the Status/Action field. Also, type the ID of the operator you want to add.

In the Classes field, type the classes of messages you want this operator to see. The default classes are 40 and 80.

If you want to hold certain types of messages on this operator’s screen, move the cursor beside the desired Message type (Information, Eventual Action, Immediate
Monitoring Messages

Decision, System Wait, or Immediate Action), and type any character. For more information on message types, see IBM Tivoli System Automation for z/OS Messages and Codes.

Change the Messages a Notification Operator Receives:
To change the messages you or another operator receives, you access the INGNTFY command dialog and you fill in the panel exactly as if you were adding a new operator. (You do not have to delete the definition first because the new definition overwrites the old.)

Because the new definition overwrites the old, you need to be careful when you are changing the messages an operator receives. For example, if the operator was originally assigned to receive classes 41 and 43, and you typed 44 in the Classes field, 44 would be the only class that the operator would now receive. If you want to add a class, you must also type all the classes that were previously assigned.

You can perform a query to find out what messages the operator currently has assigned. See "Query a Specific Notification Operator" on page 121 for more information.

Delete a Notification Operator:
You can also use the INGNTFY command dialog to delete notification operators. Remember that if you use this panel to delete an operator, the deletion will be in effect only until the automation control file is reloaded.

To delete a notification operator, type delete in the Status/Action field. Also, type the ID of the operator you want to delete. This field can contain any valid NetView operator or the keyword SYSOP. If SYSOP is defined as a notification operator its function is to receive messages to the master console when no other operator is logged on, or to receive specific messages depending on the message classes assigned.

Monitoring Hardware
You can monitor all hardware, from processors to devices, using processor operations and I/O operations.

You can also monitor hardware resources and you can issue system operations and processor operations commands from the NetView Management Console as described in Chapter 10, “Using the NetView Management Console for SA z/OS,” on page 203.

General Information About Panels: Pressing the Enter key toggles between a dynamically updated panel and a static panel. If any processor operations screen is updated frequently, it appears to flicker. You may need to place the panel into a static condition to select another function. The Updates field in the top right of the panel shows the current mode of the panel (Dynamic or Static).

Processor Operations Status Information
Processor operations allows you to monitor target processors, target systems, and focal point communication path by using the ISQXDST status panels of your NetView operator console. These panels are available only on a NetView operator console that is attached to the SA z/OS processor operations focal point system.
Displaying Processor Resource Information (SNMP Connections)

To view target hardware summary of a processor, select PF9 from the Target System Summary panel or from the Status Summary panel on a NetView console.

This panel has the following PF keys:
- The PF7 key displays the Processor Operations Interested Operator List panel (also accessed with the ISQXMON command).
- The PF11 key displays the Path Detail panel, which provides detailed status information about a specific NetView path. First place the cursor on the line with the name of the path that you want more information about, and then press PF11.

The Target Hardware Summary panel provides detailed status information about the target hardware that the target system is defined on, including:

- **Initialized Target Systems**
  (Updated dynamically) List of target systems currently initialized to this target hardware.
- **Communication Path Error Status**
  (Updated dynamically) Status information in case of a communication problem between the NetView focal point and the target hardware.

Monitoring Health Performance

You can monitor the health performance of your resources using monitor resources with:
- The Network Communications Control Facility (NCCF) panels, see “Using the NCCF Panels” on page 124
- NetView Management Console (NMC), see “Using the NMC” on page 124
- Tivoli Enterprise Portal (TEP), see “Using the Tivoli Enterprise Portal” on page 124
- The Status Display Facility (SDF), see “Monitoring with the Status Display Facility” on page 131
Using the NCCF Panels

Once you have loaded your configuration you can use the NCCF panels to check the status of your resources. You might want to start with INGLIST. If you scroll to the right using the PF11 key, you can see the Health status column.

Suppose that you are monitoring CLIENT/APL with INGLIST and you find a compound status of DEGRADED and a health status of WARNING. In this case you can enter the command code M in the CMD entry field of CLIENT/APL. This displays the DISPMTR panel showing all monitor resources (MTRs) that are connected to CLIENT/APL with HasMonitor relationships. To find out the reason for the health status WARNING you can scroll to the right to find more information or enter the command code D in the CMD entry field of the MTR, which takes you to the DISPMTR Details panel.

Using the NMC

To use the MTRs in your NMC you can define a view that will contain one or more MTRs by entering the appropriate filter criteria. If you want to see your APL together with the connected MTRs you must work out a naming convention that allows you to specify wildcards. Another (less flexible) possibility is to name every resource (APL and MTR) explicitly in the view.

If you have a view with both APLs and connected MTRs you will see the HasMonitor relationships as lines between them. You can right-click on the icons and issue the associated commands such as DISPMTR or INGMON.

Using the Tivoli Enterprise Portal

The Tivoli Enterprise Portal (TEP) is a Java-based interface to the data monitoring and management resources of the IBM Tivoli Monitoring infrastructure. It displays information about the status of automation on z/OS systems and z/OS sysplexes that has been collected by the SA z/OS monitoring agent.

The SA z/OS monitoring agent uses the TEP to provide a view of your enterprise from which you can drill down to examine details about each system being monitored. The TEP application window consists of a Navigator that shows all the systems in your enterprise where monitoring agents are installed, and a workspace that includes table and chart views of system and application conditions. Each workspace is designed to help monitor a specific component of your system. A table of attributes is provided for each workspace.

Each attribute represents a particular kind of data about system resources being monitored and reported. Attributes can also be used to define situations to test for specific conditions. When the conditions for a situation are met, situation event indicators are displayed in the Navigator.

The SA z/OS monitoring agent provides predefined workspaces, which you can access from the Navigator in the Tivoli Enterprise Portal. The product workspaces enable you to monitor the status of automation of your z/OS systems and sysplex resources. You access these workspaces from nodes at two levels of the Navigator tree: the SA z/OS level and the SA z/OS Agent level.

Attributes are characteristics or properties of the objects monitored by the SA z/OS monitoring agent. Related attributes are organized into attribute groups (also called attribute tables). The attributes are used to define the queries that collect the information displayed in tables and charts in the SA z/OS monitoring agent workspaces and to create situations that trigger alerts in response to specified conditions.
Monitoring Health Performance

The SA z/OS monitoring agent provides a set of predefined situations that monitor the status and health of automation resources within the SA z/OS subplex. These situations check for specified conditions and can trigger Critical, Warning, or Informational situation event indicators (also called alerts) in the Navigator. When a situation triggers an alert, you can investigate the situation event by opening its workspace.

A topology view is also available for Tivoli Enterprise Portal, providing a graphic view of resources and their dependencies.

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

Monitoring OMEGAMON Sessions

SA z/OS allows you to use Monitor Resources to connect to classic OMEGAMON monitors to send commands and receive responses.

You can use the INGSSESS command either from the command line or with the fullscreen command dialog to:

- Display OMEGAMON sessions
- Start sessions manually to test connection and authorization
- Stop sessions to perform maintenance
- Show additional session attributes, for example, logon data, timeout, statistics, etc.

Displaying OMEGAMON Sessions

When you enter the INGSSESS command at the NetView command line the INGSSESS command dialog is displayed.

The panel displays the following session information:

- The Session column shows the name of the session that represents an OMEGAMON monitor.
- The System column shows the system that established the connection to OMEGAMON.
- The Type column shows the type of session, that is, the OS or middleware that the OMEGAMON monitor is monitoring.
- The Status column shows the status of the session. It can be one of the following:

  **INACTIVE**
  - The session does not exist

  **ACTIVE**
  - The session exists and is ready to receive requests

  **AUTHFAIL**
  - The session could not be created because the logon to OMEGAMON failed; all requests will be denied

  **SESSFAIL**
  - SA z/OS received unexpected data or some NetView communication error occurred; all requests will be denied

  **MAINT**
  - The session was terminated for maintenance purposes; all requests will be denied
Monitoring OMEGAMON Sessions

- The **Appl-id** column shows the name of the OMEGAMON VTAM application as defined by the installation during customization of the OMEGAMON product.

- The **User id** column shows the user that is defined to log on to the OMEGAMON application. The user ID is needed to control access to the OMEGAMON application. It may be blank if product level security is not implemented for this OMEGAMON application.

- The **SessOper** column shows the automated function name that was assigned to the session by SA z/OS during initialization.

**Starting OMEGAMON Sessions**

You can start sessions manually to test connection and authorization using either:

- The command line, for example:
  ```
  INNGSESS ims742cr req=start outmode=line
  ```

- The command dialog, as follows:

<table>
<thead>
<tr>
<th>CMD:</th>
<th>Session</th>
<th>System</th>
<th>Type</th>
<th>Status</th>
<th>Appl-id</th>
<th>User id</th>
<th>SessOper</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>CICSKY41</td>
<td>OMIIICS</td>
<td>AOC9</td>
<td>ACTIVE</td>
<td>IPSPOC0</td>
<td>SAOM</td>
<td>AOFSES01</td>
</tr>
<tr>
<td>B</td>
<td>DB2SGG4</td>
<td>OMIIIB2</td>
<td>AOC9</td>
<td>INACTIVE</td>
<td>IPSPOC2</td>
<td>SAOM</td>
<td>AOFSES02</td>
</tr>
<tr>
<td>B</td>
<td>DB2SGI4</td>
<td>OMIIIB2</td>
<td>AOC9</td>
<td>MAINT</td>
<td>IPSPOC2</td>
<td>SAOM</td>
<td>AOFSES03</td>
</tr>
<tr>
<td>B</td>
<td>IMS742CR</td>
<td>OMIIIMS</td>
<td>AOC9</td>
<td>INACTIVE</td>
<td>IPSPO10</td>
<td>SAOM</td>
<td>AOFSES01</td>
</tr>
<tr>
<td>B</td>
<td>OMSY4MVS</td>
<td>OMIIIMS</td>
<td>AOC9</td>
<td>AUTHFAIL</td>
<td>IPSPM2RC</td>
<td>SAOM</td>
<td>AOFSES02</td>
</tr>
</tbody>
</table>

This creates a new session with the appropriate OMEGAMON, if it does not already exist (that is, the session status is INACTIVE or MAINT). After successful session creation, the session status is ACTIVE. The session remains active until it is stopped, either explicitly (using command code C), or implicitly by NetView on behalf of OMEGAMON or session task termination.

If the session could not be started, the status will become either AUTHFAIL or SESSFAIL.

**Stopping OMEGAMON Sessions**

You can stop sessions to perform maintenance using either:

- The command line, for example:
  ```
  INNGSESS cicsky41 req=stop outmode=line
  ```

- The command dialog, as follows:

<table>
<thead>
<tr>
<th>CMD:</th>
<th>Session</th>
<th>System</th>
<th>Type</th>
<th>Status</th>
<th>Appl-id</th>
<th>User id</th>
<th>SessOper</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>CICSKY41</td>
<td>OMIIICS</td>
<td>AOC9</td>
<td>ACTIVE</td>
<td>IPSPOC0</td>
<td>SAOM</td>
<td>AOFSES01</td>
</tr>
<tr>
<td>B</td>
<td>DB2SGG4</td>
<td>OMIIIB2</td>
<td>AOC9</td>
<td>INACTIVE</td>
<td>IPSPOC2</td>
<td>SAOM</td>
<td>AOFSES02</td>
</tr>
<tr>
<td>B</td>
<td>DB2SGI4</td>
<td>OMIIIB2</td>
<td>AOC9</td>
<td>MAINT</td>
<td>IPSPOC2</td>
<td>SAOM</td>
<td>AOFSES03</td>
</tr>
<tr>
<td>B</td>
<td>IMS742CR</td>
<td>OMIIIMS</td>
<td>AOC9</td>
<td>INACTIVE</td>
<td>IPSPO10</td>
<td>SAOM</td>
<td>AOFSES01</td>
</tr>
<tr>
<td>B</td>
<td>OMSY4MVS</td>
<td>OMIIIMS</td>
<td>AOC9</td>
<td>AUTHFAIL</td>
<td>IPSPM2RC</td>
<td>SAOM</td>
<td>AOFSES02</td>
</tr>
</tbody>
</table>

This destroys the session with the appropriate OMEGAMON. The session status will be changed to MAINT. The session is disabled to prevent it from being created implicitly again.

After you have fixed the problem you can restart the session.
Recovering from Session Failure

A session may not be created because either the logon to OMEGAMON failed, or SA z/OS received unexpected data or some NetView communication error occurred. In these cases all requests to the session will be denied.

To recover a failed session:
1. Put session in maintenance mode, for example:
   - From the command line:
     
     ```
     INGSESS omsy4mvs req=stop outmode=line
     ```
   - Using the INGSESS command dialog:

     | CMD: | B Start session | C Stop session | D Details |
     |------|-----------------|----------------|-----------|
     | Session | System | Type | Status | Appl-id | User id | SessOper |
     |--------|--------|------|--------|---------|---------|----------|
     | CICSKY41 | OMICICS | AOC9 | ACTIVE | IPSPOC0 | SAOM | AOFSES01 |
     | DB2SG64 | OMIIDB2 | AOC9 | INACTIVE | IPSPOC0 | SAOM | AOFSES02 |
     | IMS742CR | OMIIIMS | AOC9 | INACTIVE | IPSPOC0 | SAOM | AOFSES03 |
     | OMSY4MVS | OMIIIIVS | AOC9 | AUTHFAIL | IPSPOC0 | SAOM | AOFSES02 |

2. Analyze and fix the problem
3. Restart the session, for example:
   - From the command line:
     
     ```
     INGSESS omsy4mvs req=start outmode=line
     ```
   - Using the INGSESS command dialog:

     | CMD: | B Start session | C Stop session | D Details |
     |------|-----------------|----------------|-----------|
     | Session | System | Type | Status | Appl-id | User id | SessOper |
     |--------|--------|------|--------|---------|---------|----------|
     | CICSKY41 | OMICICS | AOC9 | ACTIVE | IPSPOC0 | SAOM | AOFSES01 |
     | DB2SG64 | OMIIDB2 | AOC9 | INACTIVE | IPSPOC0 | SAOM | AOFSES02 |
     | IMS742CR | OMIIIMS | AOC9 | INACTIVE | IPSPOC0 | SAOM | AOFSES03 |
     | OMSY4MVS | OMIIIIVS | AOC9 | MAINT | IPSPOC0 | SAOM | AOFSES02 |

Displaying Additional Information

You can display additional session attributes (for example, logon data, timeout, statistics, etc.) using either:

- The command line, for example:
  
  ```
  INGSESS omsy4mvs req=detail outmode=line
  ```
- The command dialog by entering the action code D for a session, as follows:

    | CMD: | B Start session | C Stop session | D Details |
    |------|-----------------|----------------|-----------|
    | Session | System | Type | Status | Appl-id | User id | SessOper |
    |--------|--------|------|--------|---------|---------|----------|
    | CICSKY41 | OMICICS | AOC9 | ACTIVE | IPSPOC0 | SAOM | AOFSES01 |
    | DB2SG64 | OMIIDB2 | AOC9 | INACTIVE | IPSPOC0 | SAOM | AOFSES02 |
    | IMS742CR | OMIIIMS | AOC9 | INACTIVE | IPSPOC0 | SAOM | AOFSES03 |
    | OMSY4MVS | OMIIIIVS | AOC9 | AUTHFAIL | IPSPOC0 | SAOM | AOFSES02 |

This produces output where the additional information includes:
Monitoring OMEGAMON Sessions

- The **Description** field shows descriptive information for this session as specified in the automation policy.
- The **Logical Unit** field shows the name of the source LU that was assigned by NetView upon establishing a Terminal Access Facility (TAF) fullscreen session.
- The **Password** field shows the password in the form of a string of asterisks or 'SAFPW'. This is the password used to logon to the OMEGAMON application.
- The **Timeout** field shows the maximum time to wait for a response from OMEGAMON before the request is terminated.
- The **Logon data** field shows the data that is sent to the OMEGAMON session during logon.
- The **Users** field shows a list of operators or *AUTO that have interacted with the session since it became ACTIVE.
- The **Statistics** section shows:
  - The **Total # Commands** field shows the number of commands that have been issued on this session since the session became ACTIVE. The counter is reset each time the session becomes ACTIVE.
  - The **Total # exception analysis** field shows the number of exception trap analysis commands that have been issued since the session became ACTIVE. The counter is reset each time the session becomes ACTIVE.
  - The **Total # exceptions tripped** field shows the number of exceptions that actually tripped as reported by the session since it became ACTIVE. The counter is reset each time the session becomes ACTIVE.

Monitoring Processor Status

Processor status can be monitored by viewing the status of a system icon on the NMC workstation or processor status can be seen in more detail using the processor operations 3270-type panels. On the NMC workstation, the status of a system icon represents the aggregation of status of the target’s operating system software and its underlying processor hardware. You can see at a glance whether the systems that you are responsible for have satisfactory status.

Refer to Chapter 10, “Using the NetView Management Console for SA z/OS,” on page 203 for information on how to enter commands against an unsatisfactory processor operations object.

This section contains the following subsections that describe certain aspects of monitoring processor status:

- “View Status Summary of all Target Systems”
- “View One Specific Target System” on page 129

**View Status Summary of all Target Systems**

On a 3270 console, issue *sqxdst* (with no parameters) on a NetView console. The Status Summary panel is displayed (Figure 32 on page 129).
Monitoring Processor Status

This panel includes the following important sections:

1. This indicates whether updates are static or dynamic (it changes automatically whenever the status changes). You can toggle to a static state with the Enter key (for example, when you want to enter a choice).

2. The status that you see for a target system represents the most severe condition that exists for that system. These values are extracted from the status, target hardware, attention, and console summary status fields of the Target System Summary panel.

3. The PF keys allow you to invoke other status panels.

**View One Specific Target System**

On a 3270 console, use ISQXDST followed by the name of the target system on a NetView console. This displays the Target System Summary panel.

This panel has the following PF keys:

- The PF7 key displays the Processor Operations Interested Operator List panel (accessed with the ISQXMON command).
- The PF9 key displays the Target Hardware Summary panel. This panel provides detailed status information about the target hardware that the target system is defined on.
- The PF11 key displays the connection Path Detail panel. This panel provides detailed status information about a specific connection path.

The Target System Summary panel shows information about the target system, including:

**Target System Name**

The name assigned in the customization dialog to this target system.

**Target System Description**

Short textual description of this target system, defined in the customization dialog.
Monitoring Processor Status

**Status**  (Updated dynamically) The current value of the processor operations internal variable `tstat`.

You can change this status from INITIALIZED, LOADFAILED, or IPLFAILED to IPLCOMPLETE by issuing the ISQVARS command to change the internal variable `tstat`. (For more information about specifying `tstat` as a keyword on the ISQVARS command, see the description of the ISQVARS command in *IBM Tivoli System Automation for z/OS Operator’s Commands*). You would want to do this when you perform a cold start of processor operations while a target system is already running.

After the cold start, you issue the ISQXIII command to the target system that is already running so that its status becomes INITIALIZED, then you change the value of `tstat` to IPLCOMPLETE.

You may also want to change the status manually if the cause of the LOADFAILED status was corrected by using the pass-through facility of processor operations, or if it was corrected locally at the site of the target system.

You can also issue the ISQVARS command to change the internal variable `tstat` from IPLCOMPLETE to IPLFAILED. Processor operations sets a status of IPLCOMPLETE when it initiates a load of a target system and receives an operating system specific message that indicates that the operating system received a level where it is ready to work. However, neither of these operating systems provide a sufficient indication that the load process failed, preventing processor operations from changing the target system status appropriately.

You can find details of valid values for this status field in the description of the ISQXDST command in *IBM Tivoli System Automation for z/OS Operator’s Commands*.

**Target Hardware**  (Updated dynamically) The current value of the processor operations internal variable `thwstat`.

You can find details of valid values in the description of the ISQXDST command in *IBM Tivoli System Automation for z/OS Operator’s Commands*.

**Attention**  (Updated dynamically) The current value of the processor operations internal variable `tattn`.

You can find details of valid values in the description of the ISQXDST command in *IBM Tivoli System Automation for z/OS Operator’s Commands*.

**Status**  (Updated dynamically) The current value of the processor operations internal variable `scstat`.

**Lock Holder**  (Updated dynamically) The current NetView operator that owns the lock for the target system and the connection path. Locks can be set using the ISQXLOC and ISQXUNL commands. Locks grant the exclusive use of the connection path of a target hardware and its associated target systems.

**Last Significant Message**  (Updated dynamically) The text of the last command response or report received from the target system. The message is not removed from the panel when the condition that generated the message is resolved.
Monitoring Processor Status

For further details see the description of the ISQXDST command in IBM Tivoli System Automation for z/OS Operator’s Commands.

How to Monitor Communications Path Status
The communications paths from the processor operations focal point system to target systems are based on SNMP. You can display details of the connection for this type of path.

Press PF11 on the Target System Summary panel or Target Hardware Summary panel to display the Path Detail Panel. This displays information such as:

Support Element IP Address or Hostname
The name or IP address assigned in the customization dialog to the SE of this target hardware.

Hardware Management Console IP Address or Hostname
The name or IP address assigned in the customization dialog to the HMC of this target hardware.

Connection Authorization
The community value is shown. The authorization values are defined in the customization dialog for a selected target hardware.

Lock Holder
(Updated dynamically) The current NetView operator owning the lock for the connection path. Locks can be set and unset using the ISQXLOC and ISQXUNL command. Locks grant the exclusive use of a connection path.

SNMPv3 Security
SNMPv3 security status.

Y The connection is SNMPv3 and authenticated by SNMPv3 User and Password.

N The connection is SNMPv2 and authenticated by the community name (Connection Authorization).

SNMPv3 User
SNMPv3 User specified for the connection.

Path Status
(Updated dynamically) The status of the NetView path.

How to Monitor Alerts on a NetView Console
Alerts are high priority events that require operator action.

Issue NPDA from the command line of your NetView operator’s console. The NPDA (Network Problem Determination Application) menu is displayed. You can request a dynamic display of alerts. From that display you can select an individual alert and see event detail and recommended action. For more details on the use of the NPDA panels, refer to Tivoli NetView for z/OS User’s Guide.

Monitoring with the Status Display Facility
The Status Display Facility (SDF) consists of a hierarchy of dynamically updated panels showing color-coded status conditions. It is set up by your system programmer during the customization of SA z/OS.

SDF displays the status of various resources in a z/OS system, and in other z/OS systems that specify this system as an automation focal point.
Monitoring with the Status Display Facility

You use SDF to monitor the status of application software, WTORs, and gateways on your systems from a NetView or operator console.

The status conditions that can be displayed by SDF include those for:

- Applications and subsystems
- Monitor resources
- WTORs
- Gateways: A gateway is a combination of a NetView-NetView task session and two automation operators (one on each of two systems) that allows communication of messages, commands, and responses between the two systems.
- Application groups
- Spool usage problems from z/OS subcomponents
- Exceptional messages
- Processors
- Ensembles

Because SDF can be customized, your panels may not look like those shown in this topic.

Setting Up SDF

All the resources that you need to monitor are set up by a hierarchical arrangement during customization of SA z/OS, when the system programmer:

1. Defines the SDF hierarchy. The SDF hierarchy indicates which automated resources are affected by status changes in other automated resources (see “How SDF Works” on page 133). The SDF hierarchy can be different from that specified in the automation control file for system startup and shutdown. Use whatever makes sense for your system.

   If SDF displays more than one system, programmers create an SDF hierarchy for each system.

2. Creates the panels that show the systems and subsystems. These panels can also be customized to display gateway sessions and WTORs.

3. Determines the status colors and other screen defaults for the enterprise or takes the defaults for these attributes.

For more information on setting up SDF, see IBM Tivoli System Automation for z/OS Programmer’s Reference.

SDF Focal Point Monitoring

Initially you need to construct an SDF screen that contains all the systems within your sysplex. This gives you an at-a-glance panel that is dynamically updated with the overall status of each system in your sysplex. By selecting an individual system, you can view a more detailed display of the state of the resources on that system.

On the focal point, you can issue a command that specifies a target system. The command interacts with the target system (via XCF or RMTCMD) and gives you a full screen display as if you were logged on to the target system.
This means that from SDF you can issue DISPINFO on a target subsystem by simply pressing a key. This allows you to see detailed information about a subsystem that has a problem.

Other commands that support this kind of usage include:
- DISPERRS
- DISPFILGS
- DISPSTAT
- DISPTREE
- INGGROUP
- INGINFO
- INGREQ
- INGTHRES
- INGVOTE
- SETSTATE

All of these commands can be used from the focal point to interrogate the target system without having to log on to it.

Note: If the target system is not in the same SAplex as your focal point system, RMTCMD is used for intersystem communication. This means that there is a dependency upon VTAM being active on both the focal point and the target system for communication to be possible. If the target system is in the same SAplex, communication occurs via XCF and you do not need to have VTAM active on the target system to talk to it.

Interpreting Colors in SDF and DISPSTAT
In SDF and the DISPSTAT command dialog, subsystems appear in different colors that indicate their status. In SDF, the condition of WTORs, application groups (APGs), monitor resources (MTRs), and gateways are also indicated by color. Refer to "Priority and Color Default Assignments" in IBM Tivoli System Automation for z/OS Programmer’s Reference for the list of default priority and color assignments to status values. You can modify these values or define your own values when you define your own SA z/OS policy. For more details refer to "Status Display Entry Type" in IBM Tivoli System Automation for z/OS Defining Automation Policy.

Note: All SDF panels can be modified according to your needs.

How SDF Works
SDF uses the hierarchy established during SDF customization to propagate colors up to the highest level panel. Figure 33 on page 134 shows a graphic representation of an example SDF hierarchy for KEY1.
Imagine that TSO appears in yellow on your screen because it is in AUTOTERM status. This color is reflected up the hierarchy to APPLICATION, SYSTEM, and KEY1, because AUTOTERM has a higher priority than the status UP (indicated by the color green). If various resources in a system have different status conditions, the one with the highest priority status controls the color of that system. This allows you to resolve issues concerning the highest priority status first, then deal with the lower priority status conditions.

**Accessing SDF**

You can access SDF from either a NetView session or the SA z/OS command dialogs (see "Using Command Dialogs" on page 82). Enter one of the following:

- sdf at a NetView command line
- sdf or 3 (SDF) at the command line of the SA z/OS System Operations Main Menu

This displays the highest level (or root) panel of SDF. Figure 34 on page 135 shows an example of the highest level panel for SA z/OS-supported systems. On this panel, KEY1 is highlighted to indicate that it appears in reverse red on your screen, which means that a resource on KEY1 is BROKEN. All other systems may appear in green, which means UP.

**Using SDF**

You use SDF to monitor the status of subsystems, WTORs, APGs, MTRs, and gateways on your systems and other systems. To monitor status, you observe the highest level SDF panel (also known as the root panel or system summary panel). Figure 34 on page 135 shows a panel with the systems of a sysplex.
Monitoring with the Status Display Facility

On this panel, each system is displayed in a color that reflects the highest priority status of the resources in that system. If a resource in a system changes status, the system changes color to reflect the new status. By observing this panel you can see status changes in any of your systems.

You can use the PF8 and PF7 keys to scroll between any further panels. You can tell whether this is possible because the list of PF keys at the bottom of the panel shows 8=ZOOM/NEXT. Do not position the cursor on a system because this invokes the zoom function of the PF8 key.

You select a system to investigate by moving the cursor to that system using either the Tab key or the mouse.

Imagine that KEY1 IPSFM is displayed in reverse red. This indicates that a resource on system KEY1 is in the BROKEN status. To view more information about the broken resource on this system, select KEY1 by moving the cursor to it and pressing PF2 to display a Detail Status Display panel. This gives a detailed description of the broken resource with the highest status priority. This information is also shown for all the resources that are defined in that domain on successive panels that are ordered from the highest status priority to lowest. See “How to Check Resource Status Conditions” on page 136 for more information about the Detail Status Display panel.

Pressing PF8 when you select a system shows a Resource Summary Status panel that gives an overview of the different resources, messages, and components that are being monitored. To view further details, select an item and press either PF2 or PF8:

PF2 This displays the Detail Status Display panel for the component with the highest status priority.

PF8 This displays a status panel that lists all the components in order of their status priority and in the colors that represent their individual statuses.
Monitoring with the Status Display Facility

Select a component in this summary list and press PF2 to display its Detail Status Display.

There may be a list of PF keys that invoke SA z/OS command dialogs for these components at the bottom of the panel, such as:

13=EXPLAIN 17=SETSTATE 18=INGVOTE 19=INGREQ 23=INGLIST 24=INGINFO

Press the Shift and PF key to access these command dialogs, for example, Shift+PF5 for PF17, SETSTATE.

How to Check Resource Status Conditions

Normally when you use SDF to monitor resources, you observe the highest level panel until you see a system change color. This color change indicates a change in status in one of the resources in the system. To determine which resource, and to see more information on the status change:

1. Select the system (or subsystem) that shows a status change by moving the cursor to it using the Tab key or the mouse.
2. Press PF2. This takes you to the Detail Status Display panel for the resource that caused the system to change color.

If more than one resource changes status at the same time, SDF shows the information for the system with the highest priority status first. You can press PF8 to page through the Detail Status Display panels for other resources on the system.

In the example from Figure 34 on page 135 you select KEY1 (the system in reverse red). When you press PF2, you see the Detail Status Display panel for HUGC2APL0 (Figure 35), which is the resource with the highest priority because it is BROKEN.

--- DETAIL STATUS DISPLAY ---

Component: HUGC2APL0 System : KEY1
Color : RED Priority : 55
Date : 03/04/14 Time : 10:20:35
Reporter : GATIPSFM Node : IPSFM
Jobname : HUGC2APL0
Reference value: C3057D7610F15E0E

AOF571I 10:20:05 : HUGC2APL0 SUBSYSTEM STATUS FOR JOB HUGC2APL0 IS BROKEN - AT SYS-OPS RECYCLE , JOB AND ALL DEPENDANTS WILL NOT BE STARTED

03/04/09 12:34

Figure 35. Example SDF Detail Status Display Panel for HUGC2APL0

This panel also shows the automation or human operator that sent the notification and gives the priority number that corresponds to the automation status. Priority
numbers determine which color the resource is displayed in and the order of the Detail Status Display panels (the lower the number the higher the priority).

In this example, the panel is the first of 242 Detail Status Display panels (as indicated at the top right corner of the panel), one for each of the resources on KEY1 that has SDF has detail information for.

You can use the following PF keys:

**PF3** Returns to the previous panel.

**PF4** Deletes the panel you are currently on. (The record is physically deleted.)

*Note:* You should not normally use this PF key.

**PF7** Scrolls back up the Detail Status Display panels in order of descending status priority.

**PF8** Scrolls down the Detail Status Display panels in order of descending status priority.

**PF11** Moves to the Detail Status Display panel for the resource with the lowest priority, that is, the last panel in the Detail Status Display stack.

**PF12** Moves to the Detail Status Display panel for the resource with the highest priority, the first panel in the Detail Status Display stack.

*Note:* You can customize the PF keys. For more details, see “Status Display Facility Definitions” in *IBM Tivoli System Automation for z/OS Programmer’s Reference*.

**Deleting SDF Records**

The SDFCONF command is assigned to the PF4 key on the Detail Status Display panel to delete an SDF record. This is useful because it prompts you for confirmation before performing the actual deletion and leads to a Delete Confirmation panel, as shown in [Figure 36 on page 138](#). If you do not want the prompt panel to appear, then add "VERIFY=NO" to the end of the SDFCONF command.
Monitoring with the Status Display Facility

![Image of Delete Confirmation Panel for an SDF Record](image)

**Figure 36. Delete Confirmation Panel for an SDF Record**

On this panel you can:

- Press Enter to confirm the deletion of the SDF detail entry
- Press PF3 to cancel the deletion request and to return to the previous panel
- Press PF5 to delete the entry from SDF only using the internal SDFDEL command

**Note:** Deleting an exceptional message from SDF removes the message from all other interfaces where it is shown, for example, in TEP or NMC. Likewise, deleting the message in TEP removes it from the NMC, SDF and the system console.

### Config Refresh Monitoring

The tree and panel definitions are defined in the members INGTCFG:

```
1 INGCFG
2 &SDFCsaplex.
3 AGENT
```

and INGPCFG:

```
P(INGCFG,*,*,SYSTEM,SYSTEM,,*,*)
: BODY(INGCFG,05,=-6,6,03,79,,VA)
BH(01,L,S,T,N,68,79)
CELL(02,09,N,V)
CELL(11,26,N,B)
CELL(28,35,N,C)
:
```

Both members are included in the members AOFTREE and AOFPNLS. The subsequent snapshots base on the following CNMSTYLE sheet definitions:

```
COMMON_AOF_AAO_SDFCSAPLEX.0 = 2
COMMON_AOF_AAO_SDFCSAPLEX.1 = KEYAB_PLEX
COMMON_AOF_AAO_SDFCSAPLEX.2 = KEYC_PLEX AOC1_PLEX
```
The first entry shows the number of subsequent definitions. The remaining entries contain the SAplex names. They can be defined as a single entry or as shown as a list.

Since duplicate names are ignored it would be a good practice to define the local SAplex as the first entry and the names of all SAplexes that define the enterprise in the subsequent entries. Non-focal point systems should specify the ",0" a value of 1 to avoid unnecessary storage allocations and processing overhead.

Refer to "AOFTREE" in IBM Tivoli System Automation for z/OS Programmer’s Reference for the general usage of the AOF_AAO_SDFCxxx.n variables.
Working with the Looping Address Space Monitor

The Looping Address Space Suppression procedure is an active monitor that runs inside an APL. The suggested naming for the APL is to give it a subsystem name of LOOPSUPP.

Stopping and Starting the Procedure

The procedure should normally be running 24 hours a day, but if you need to stop it you can use INGREQ and INGVOTE to ask for it to be shutdown or restarted like a normal APL or both.

Changing its Automation Flag

The execution of the Monitor within the APL is controlled by a minor resource called MONITOR. The default for it is set to LOG. You can use INGAUTO to dynamically change its setting.

Specify LOOPSUPP as the subsystem and MONITOR as the minor resource. Setting NO disables the monitor, setting LOG restricts it to only logging its active recovery commands and setting YES lets it run fully.

Changing the Automation Flags for specific address spaces

Before it takes any recovery action for an address space, the code checks an automation flag specific to the address space. This is a minor resource under the LOOPSUPP system and is named category.jobname.

The category is the recovery category derived from the categorization process. It is printed out in the monitoring report in the Netlog if you need to look it up.

Setting NO turns recovery off, setting LOG only permits non-disruptive recovery and setting YES permits full recovery.

You can also set a flag for just the category, which then affects all address spaces that fall into that category that do not have specific automation flag settings. This enables you to, for example, turn all TSO_USER recovery off.

If you need to make a quick change, you need to use INGAUTO to change the automation flag.

Warning Messages

When the loop suppression monitor issues a warning message (ING601E) in response to a WARN directive, the resulting message is issued to the Netlog as classes 40 and 44. Any notify operator with either of these classes active receives it. If the SYSOPER notify id is subscribed to either class, it is issued to system consoles and the SYSLOG as well.

Warning messages are also posted to SDF, under the MVSESA qualifier. They may be viewed through SDF. The Monitor automatically clears the message if it is shut down or it is no longer detected as being looping. The color of the message (Yellow, Pink, Red) is determined by how many consecutive passes it has detected on (<10, <20, 20+).
Working with the Looping Address Space Monitor

**Viewing the Monitoring Report**

The report is visible in the Netlog with message ID ING600I.

**Viewing the Diagnostic Reports**

The reports appear in the Netlog with the message ID ING602I a few seconds after the monitor report that they are associated with.

A typical diagnostic report looks like the following:

```xml
ING602I DIAGNOSTICS FOR REXXLOOP (ASID 00D8)
Messages from the Inspect tool:
KM3IN008I GRANULARITY SET TO 0X00000B40
Requested 1000 samples at 5us intervals
Took 1000 samples of which 1000 were valid.
Program: EXEC
 TCB : 008E3390 (100.0 JOB)
 Lmod : IRXINIT @ 0C467000 (100.0 JOB 100.0 TCB)
 CSECT: *-UNKN-* @ OC467000 (100.0 JOB 100.0 TCB, 100.0 LMOD)
 Offset Percentage
-------- ----------
+00000B40 73.3
+00009240 6.6
+00000000 5.1
+00009D80 3.4
+00006540 3.1
+000021C0 2.4
+00007080 2.2
+00001680 1.7
+0000A8C0 1.1
+0000B400 0.7
+00007BC0 0.1
```

The first line gives us the jobname and the ASID.

Then follow one or more messages that were produced by the OMEGAMON inspect tool. These all have message ids like KM3INnnnn.

The next two lines give details of the number of inspection samples that were requested, their interval, the number that were taken and the number that were valid. The sampling values used are the default for the inspect tool.

The inspect tool works by observing the location of the program counter within the target address space. This tells us which opcode within the program it is about to execute. By repeatedly sampling this, OMEGAMON can build up a profile of the program short term execution flow. If all the samples are within a few bytes of each other then it is in a very tight loop. If they are more spread out, then the body of the loop encompasses a large part of the program's control structure.

Note that each TCB has its own program counter, so for a multi-threaded program you get multiple CPU traces.

If the program has made a cross memory call, the load module details are proceeded by a line indicating the address space (and its ASID) that the load module is within.

For the report above:
The address space is executing a program called EXEC. This is the normal way of calling the REXX interpreter.

There is a single Task Control Block (TCB), which is at address 008E3390. 100% of the samples indicated that the TCB was being dispatched.

The Load Module is IRXINIT, also the REXX Interpreter, and it is loaded at address 0C467000. 100 percent of the samples found active execution within it. If execution is passing between two load modules, the load module, CSECT and OFFSET records are repeated for each load module.

The name of the CSECT cannot be determined but its load address is the same as the load modules, so we are in the mainline of the load module. 100 percent of the samples placed the program counter within the CSECT.

The OFFSET statements need to be read in conjunction with the KM3IN008I message from the Inspection tool. The sampling found the program counter in a number of different offsets. For reporting purposes they have been grouped together into blocks. Each block is of the width stated in the granularity message (B40 bytes in this case) and is identified by its base offset from the start of the CSECT (so +0000000, +0000B40, +00001680 and so on). The list of blocks is sorted by percentage of the samples that were found within that block. This information can be used to work out what the program is doing.

Monitoring and Controlling a Sysplex

SA z/OS provides a command interface to monitor and operate sysplexes. It helps operators to manage critical sysplex resources like coupling facilities and CF structures, couple data sets, and more. You can benefit from reduced sysplex operations complexity leading to higher sysplex availability and greater productivity.

The following section describes the panels of the system operations commands that support you in working with sysplex specific automation tasks. For details about the syntax and parameters of these commands, refer to IBM Tivoli System Automation for z/OS Operator’s Commands.

How to Work with INGPLEX

SA z/OS provides the INGPLEX command to monitor and control the couple data sets and coupling facilities of your sysplex.

If you enter INGPLEX in the command line without any parameters, the INGPLEX Selection Panel is displayed. To launch other command dialogs that show sysplex-related information, specify the number or the function and press Enter.

The subcommands support the following functions:

1 SYStem
   Displays the target sysplex name, its GRS mode and its systems. For each system, details about the system name and status are shown.

2 CONsole
   Displays details about the target sysplex. For example, the master console name, WTO and WTOR buffer utilization, and the number of queued messages.

3 CF
   Displays the coupling facilities in the sysplex. For each coupling facility the structures and paths can be displayed, and the coupling facility can be drained and enabled.
How to Work with INGPLEX

4 CDS  Displays the couple data sets in the sysplex. For each couple data set you can obtain detailed information, and you can switch the couple data sets or allocate new alternate couple data sets.

6 IPL  Collects IPL information.

7 DUMP  Displays the DUMP submenu.

10 INGSTR  Displays all the structures in the INGSTR panel. See “How to Work with INGSTR” on page 159

These functions are described in more detail in the following sections.

INGPLEX SYStem

INGPLEX SYStem displays the target sysplex name, its GRS mode and its member systems.

- The Sysplex field shows the name of the sysplex.
- The GRS Mode field shows the GRS mode of the target system. The mode can be either STAR or RING.
- The CMD column allows you to specify one of the command codes to display more information. Type the appropriate letter next to the resource name, and press Enter.
- The System column shows the name of the system.
- The Status column shows the status of the system.
- The Monitor Timestamp column shows the last time stamp recorded for status monitoring on this system.
- The INTERVAL column shows the system failure detection interval in seconds. This interval is the time XCF lets elapse without a status update before assuming that the system failed.

The last three columns contain configuration data of the SFM policy (if applicable).

- The SSUM Action field shows the SSUM action. It can be one of the following:
  - ISOLATE
  - DEACTIVATE
  - RESET
  - PROMPT
  - N/A
- The SSUM TIME field shows the SSUM interval as specified in the current SFM policy.
- The SSUM WEIGHT field shows the SSUM weight specified in the current SFM policy. This weight is used in sysplex configurations after a signalling connectivity failure.

Use the PF11 key to view WLM-related information to the right:

- The Timestamp field shows the last time stamp when capacity data was queried from WLM.
- The SUs Total field shows the number of free SUs in the last 10 minutes.
- The SUs Used field shows the number of used SUs in the last 10 minutes.
- The Resource field shows the number of SU-consuming resources with a desired state of AVAILABLE.
The **SUs Exp.** field shows the number of free SUs, taking into account the resources that SA z/OS is about to start or stop.

To display more detailed information for one of the systems, enter the following command codes in the **Cmd** field next to it:

- **C** Displays the online or offline status of one or more processors and any vector facilities, or ICRFs attached to those processors
- **E** Displays the timer synchronization mode and ETR ports
- **I** Displays IPL information
- **O** Displays IOS-related configuration information
- **S** Displays the number of megabytes of central and expanded storage assigned and available to the system
- **D** Displays the device number of one or more inbound or outbound signalling paths that XCF can use and information about inbound or outbound XCF signalling paths to this system
- **T** Displays detailed signalling path information for all coupling facility structures

**INGPLEX CONsole**

**INGPLEX CONsole** displays the following information for the target sysplex:

- The name of the master console
- WTO & WTOR buffer utilization
- Number of queued messages (replies) of various types
- Awaiting mounts
- Operator requests and list of consoles (name, status, authority, number of WTOR buffers, UD, device, system, ALTGRP, MSCOPE)

Use one of the following command codes to get more information for the selected console or consoles.

- **D** Displays details for the console
- **R** Displays current requests for the console

The fields on the command dialog panel display the following information:

- The **Sysplex** field shows the name of the sysplex.
- The **Message Buffer Usage** field shows the limit of the number of WTO message buffers allowed outstanding.
- The **Awaiting Replies** field shows a decimal number of messages awaiting replies.
- The **Immediate Action** field shows a decimal number of outstanding immediate action messages (with descriptor codes 1 or 2). If the number is greater than 99999, asterisks appear in this field.
- The **Critical Action** field shows a decimal number of outstanding critical eventual action messages (with descriptor code 11). If the number is greater than 99999, asterisks appear in this field.
- The **Master Console** field shows the name of the master console.
- The **Reply Buffer Usage** field shows the limit of the number of WTOR message buffers allowed outstanding. The maximum value of yyyy is specified by the RMAX parameter in the CONSOLxx parmlib member.
• The **Eventual Action** field shows a decimal number of outstanding eventual action messages (with descriptor code 3). If the number is greater than 99999, asterisks appear in this field.

• The **Awaiting Mounts** field shows a decimal number of outstanding mount requests.

• The **Operator Requests** field shows a decimal number of outstanding requests for operator intervention.

The following details are shown for each MCS console that has been defined.

If the MSCOPE information does not fit on the primary screen, the PF10 function key is available for you to toggle between the primary panel and the panel showing only MSCOPE information.

• The **Console** column shows the name of the console as specified in the CONSOLxx parmlib member.

• The **Status** column shows the status of the console. The following values can occur:

  - **HARDCOPY**
    - Hardcopy log. This condition is indicated when the console is active on the system where the command processes.

  - **ACTIVE**
    - Active console.

  - **ACTIVE-P**
    - In the process of becoming an active console. This condition will be indicated only when the console is active on the system where the command is processing.

  - **MASTER**
    - Master console

  - **INACTIVE**
    - Inactive console

  - **INACT-P**
    - In the process of becoming a non-active console. This condition will be indicated only when the console is active on the system where the command is processing.

  - **PROB-DET**
    - The active system console is in the problem determination mode. PD is indicated only for the system console.

  - **SUBSYS**
    - Subsystem-allocatable console

• The **AUTH** column shows which commands can be entered from this console. The following values can occur:

  - **ALL**
    - Any INFO SYS, IQ or CONS command can be entered from this console.

  - **CONS**
    - INFO commands and any commands from the console command group can be entered from this console.

  - **INFO**
    - Any command from the informational command group can be entered from this console.

  - **IO**
    - INFO commands and any commands from the I/O Control command group can be entered from this console.
MASTER
The specified console is authorized to enter any operator command.

NONE
This console has no command authority.

SYS
INFO commands and any commands from the system control command group can be entered from this console.

- The NBUF column shows the number of WTO message buffers currently queued to this console. If nnnn is greater than 9999, asterisks (****) appear in this field.
- The UD column shows whether this console is receiving messages with the UD attribute.
- The Device column shows the name of the console as specified in the CONSOLxx parmlib member.
- The System column shows the system name of the active console.
- The ALTGRP column shows the alternate group defined for this console.
- The MSCOPE column lists the name of the system or systems that this console is receiving unsolicited messages from. Note that these systems might be different from the system where this console is physically attached.

INGPLEX CF
With INGPLEX CF you can:
- Drain or enable a selected coupling facility
- Display the paths and structures to a selected coupling facility

This panel displays all coupling facilities in a sysplex including particular storage information of each coupling facility. On this panel you can select one of the following actions by entering the command code next to the coupling facility:

D (Drain CF)
Releases a coupling facility from its structures and connections in order to remove it from the sysplex for maintenance.

E (Enable CF)
Integrates a coupling facility into a sysplex.

M (Start or stop maintenance mode)
Puts the coupling facility into or takes it out of maintenance mode.

P (Display sender paths)
Displays the paths from the systems in the sysplex having a connection defined to the specified coupling facility. For each path the physical and the logical status are displayed.

S (Display structures)
Shows all structures allocated in the specified coupling facility.

These actions are described in detail in the following sections.

Drain CF:
With this function, you can remove a CF from the sysplex, for example, for maintenance purposes. You can perform the following sequence of tasks:
1. Display information for all allocated structures of the CF.
2. Put the coupling facility into maintenance mode if the MAINT function is available.
3. Rebuild all rebuildable structures on another CF and delete instances of structures on the target CF that are being duplexed on another CF.
4. Force the deletion of structures that have no active connectors and cannot be rebuilt.

   **Note:** There are structures that you can neither rebuild nor delete with the force action. These include the structures that have at least one active connector and do not support rebuild. To remove such structures first disconnect all active connectors, and then delete the structure manually if it is persistent, or has persistent connections.

5. When the coupling facility is not in maintenance mode, disconnect the coupling facility from the systems that it is connected to.

6. Disconnect the CF from the systems that it is connected with.

The DRAIN option ensures that these actions are performed in the correct order, as specified above.

Depending on the status of the CF and its allocated structures, you can execute one of four actions with the program function keys:

**REBUILD (PF10)**

 Starts the rebuild of structures that can be rebuilt on another CF. Therefore, a rebuild is only initiated for structures whose preference list contains more than one CF.

There are two methods for rebuild, user-managed and system-managed rebuild. User-managed rebuild is supported for all release levels. System-managed rebuild is only available with systems that have z/OS 2.8 or above; it must have been enabled by formatting the CFRM couple data sets with the specification:

```
ITEM NAME(SMREBLD) NUMBER(1)
```

System-managed rebuild is only performed when the requirements for user-managed rebuild are not met. This applies, for example, to structures without active connectors.

The REBUILD action also deletes all structure instances on the target CF that are being duplexed on another CF.

**FORCE (F5)**

Forces the deallocation of structures with one of the following conditions:

- No connection exists.
- No alternate CF for structure with no active connections.
- No alternate CF for structure with no connections.

This action is only available after all structures that can be rebuilt have been rebuilt.

**MAINTON (F5)**

Puts the coupling facility into maintenance mode.

Note that this function is only available before you start the rebuild process. Once you have started the rebuild process and you want to put the coupling facility into maintenance mode you need to issue the command `INGCF MAINT cf_name MODE=ON`.

**DRAIN (F4)**

Disconnects the coupling facility from its connected systems by setting the sender paths OFFLINE.
This action is only enabled after all structures of the target CF have been removed to another CF or deallocated. Note that structures that have active connectors but do not support rebuild cannot be removed with PF10 or F5. They must be deallocated manually before execution of this step is enabled.

Example of Draining a CF:
The status of the CF (NORMAL) and the authorization type of the operator (ALL) are displayed on the right of the INGCF DRAIN panel. The main part of the panel consists of a list of the structures allocated in CF1 and their conditions. The conditions are classified by color and an asterisk. The asterisk signifies that a structure cannot be rebuilt.

The only action that is enabled is REBUILD with PF10. Pressing PF10 leads to a confirmation panel. After PF10 has been pressed, the structures are in the process of being rebuilt.

After the rebuild has been completed, a panel is displayed that shows that one structure could not be rebuilt because no alternate CF is specified in its preference list. The REBUILD status is no longer available. Instead, the FORCE action (F5) is offered because the structure that could not be rebuilt has a condition that allows forcing the deallocation of the structure. Pressing F5 displays a confirmation panel similar to that for REBUILD.

Pressing PF10 on the confirmation panel and refreshing the command dialog after the action has been completed results in a panel that shows that no more structures are allocated in the coupling facility, so the coupling facility can be released from the connections with the systems of the sysplex. Consequently, INGCF DRAIN enables the DRAIN action (F4). After completion of that action, the status of the coupling facility changes to DRAINED. Because the coupling facility is no longer connected to any system, it can be inactivated. After pressing F11 the status of the coupling facility changes to INACTIVE.

Example of Draining a CF Using MAINTMODE:
In the following example, a coupling facility is drained:
1. The coupling facility is put into maintenance mode.
2. All of its structures that can be rebuilt are rebuilt on another coupling facility, and duplexing is stopped.
3. Deletion is forced for all structures that have no active connector and cannot be rebuilt.
4. The coupling facility is made inactivate.

Note: It is no longer necessary to turn the sender paths offline when the coupling facility is in maintenance mode, regardless of whether or not all systems in the sysplex run z/OS 1.9 or later.

When you issue INGCF with the DRAIN option, you can specify the coupling facility to be drained, for example, by entering INGCF DRAIN CF02. If you do not specify a coupling facility name, INGCF displays a selection panel with all coupling facilities that are defined in the sysplex.

The status of the coupling facility (NORMAL) and the authorization type of the operator (ALL) are displayed on the right side of the panel header. The main part of the panel consists of a list containing the structures allocated in CF02 and their conditions classified by color and an asterisk. The asterisk signifies that a structure cannot be rebuilt.
conditions. The conditions are classified by color and an asterisk. The asterisk signifies that a structure cannot be rebuilt.

Two actions are enabled, MAINTON with F5 and REBUILD with F10. Pressing F5 calls the confirmation panel for starting the maintenance mode.

After pressing F10 on the confirmation panel, the main panel shows the new status (NORMAL MAINTMODE) and leaves only F10 for the rebuild process. Pressing F10 calls the confirmation panel for the rebuild process.

After pressing F10 on the confirmation panel and the rebuild is complete, you can refresh the command dialog with F9.

One structure could not be rebuilt because no alternate coupling facility is specified in its preference list. The REBUILD action is no longer available. Instead, the FORCE action (F5) is available because the structure that could not be rebuilt has a condition that allows forcing the deallocation of the structure.

Pressing F5 calls a confirmation panel similar to that for REBUILD. Pressing F10 on the confirmation panel and refreshing the command dialog after the action has been completed results in an empty panel and the status of the coupling facility has changed to MAINTMODE.

Because no more structures are allocated in the coupling facility, it can be inactivated. After pressing F11 the status of the coupling facility changes to INACTIVE.

Enable CF:
The ENABLE function of the INGCF command is intended to support the integration and reintegration of a coupling facility into a sysplex. With this option, you can:

1. Activate the target coupling facility.
2. Connect the systems of the sysplex with the coupling facility.
3. Take the coupling facility out of maintenance mode when it is in this mode and the MAINT function is supported.
4. Switch to another CFRM policy if the target CF is not defined in the active policy and a suitable policy is available.
   A suitable CFRM policy must contain:
   • A definition of the target CF
   • Appropriate definitions for every active CF and every allocated structure
5. Rebuild all structures on the target CF whose preference list starts with this CF, unless this is excluded by other requirements.

The ENABLE CF option ensures that these actions are performed in the correct order, as specified above.

The following function keys support these actions:

*ACTIVATE (F11)

This action activates the CFCC (Coupling Facility Control Code) through the BCP internal interface by an ACTIVATE command.

Note: This function key is unavailable when running on a z/OS image that runs under z/VM®.
*ENABLE (F4)
Sets the sender paths of all systems of the sysplex to ONLINE. This action is enabled when the CF is active.

*MAINTOFF (F5)
Takes the coupling facility out of maintenance mode.

*SWITCH (F5)
Switches to another CFRM policy when the target CF is not defined in the active CFRM policy and a suitable policy is available. When there is more than one suitable policy you can choose one of these from a selection panel.

A CFRM policy is suitable when it contains:
- A definition of the target CF
- Definitions for every active CF and every allocated structure

This action is only available when the target CF is active, but not defined in the current CFRM policy.

*POPULATE (PF10)
Starts a rebuild process where all structures that have the target CF at the beginning of their preference list, but are currently allocated on another CF, are allocated on the target CF.

This action requires that the CF is enabled, connected to all members of the sysplex, and defined in the current CFRM policy. The action is offered whenever INGCF ENABLE detects that a structure is not allocated on the target CF although it is the preferred CF of that structure.

Note: When you have drained a coupling facility with INGCF DRAIN and then reintegrate it with INGCF ENABLE, be aware that the set of structures that are allocated on the target coupling facility after population can be different from the original set before the draining. Typically, this happens when the original set does not contain exactly those structures that have the target coupling facility at the first position in their preference list.

Note that these actions can only be performed when INGCF ENABLE is called in full mode. In line mode, only the display function is available.

In the following example, a coupling facility that has already been activated is reintegrated into the sysplex in two steps:
1. The coupling facility is connected to all systems of the sysplex.
2. All structures that have the target coupling facility as the first coupling facility in their preference list are allocated on the target coupling facility.

If you issue INGCF with the option ENABLE, you can specify the coupling facility to be reintegrated, for example by entering INGCF ENABLE CF02. If you do not specify a coupling facility name, INGCF shows a selection panel with all coupling facilities that are defined in the sysplex.

After selection of CF02, INGCF displays a panel that shows that the selected CF has already been activated manually, therefore its status, as shown on the right of the panel, is DRAINED. The authorization type of the operator (ALL) is also displayed on the right of the panel. The main part of the panel is empty because no structures are allocated in CF02. The only action that is activated is ENABLE with PF4. If you press PF4 a confirmation panel is displayed.
After pressing PF10 on the confirmation panel, the command dialog shows that the status has changed to NORMAL, and you can populate the CF with PF10. This implies that the target CF is defined in the active CFRM policy.

The structure list contains three entries with the condition 'Structure is currently allocated in CF01.' These are the structures that are currently allocated in CF01, but have CF02 at the first position in their preference list.

Pressing PF10 populates the CF, and the refreshed panel shows that the POPULATE action is no longer available because all structures whose preference list starts with CF02 are now allocated in CF02.

**Display Sender Paths:**
With this function, the paths from the connected systems to the specified coupling facility are displayed.

The last sender path of each system can only be set to OFFLINE when no more structures are allocated.

The following command codes are available:
- **F** Sets the sender path OFFLINE.
- **N** Sets the sender path ONLINE.

The fields on the command dialog panel display the following information:
- If you have issued INGCF with the PATH parameter, the **Coupling Facility** field is an input field. To display the path list of another coupling facility specify the name of the coupling facility in this field and press Enter.
- The **Allocated Structures** field shows the number of allocated structures.
- The **Permission** field shows your authorization level.
- The **System** column contains the names of the systems that are connected to the target CF.
- The **CHPID** column shows the IDs of the sender channel paths.
- The **Physical** column shows the status of the sender channel paths.
- The **Logical** column shows the logical status of the paths to that CF.
- The **Type** column shows the type of the sender channel paths.

**Display Structures:**
The STRUCTURE function of the INGCF displays the allocated structures of a coupling facility. You can initiate a rebuild or deallocation of a selected structure if the conditions for these actions are satisfied.
- If you have specified INGCF with the STR parameter, the **Coupling Facility** field is an input field. To display the structure list of another coupling facility, specify the name of the coupling facility in this field and press Enter.
- The **Include Condition** field is an input field. By entering **Yes** or **No** in this field you determine whether the conditions of the structures are displayed in the **Structure** column.
- The **Permission** field shows your authorization level. There are two possible values, **ALL** and **DISPLAY**. **DISPLAY** indicates that you can only use the display functions. **ALL** indicates that you can also rebuild and delete structures.

You can specify an action code before every structure entry. The codes you can enter depend on your authorization level.
- The **Structure** column shows the names of the structures.
INGPLEX CF

- The letter P in the P column indicates that policy changes are pending for the structure.

A structure has policy changes pending when it was allocated at the time of a CFRM policy switch, and XES could not bring the switch into effect for that structure. One reason for a pending policy change is that the old and the new policy define the structure differently, for example, with different preference lists.

- The D column indicates the type of duplexing that is possible. The following values are possible:
  U  User-managed duplexing
  S  System-managed duplexing
  B  User-managed and system-managed duplexing

- The Condition column shows the status of the structures. You can switch the display of the conditions on and off with the Include Condition field.

The following command codes are available:

D  Displays detail information about the structure.

*F  Forces the deallocation of the structure if it has one of the following conditions:
  • No connection exists.
  • No alternate CF for structure with no active connections.
  • No alternate CF for structure with no connections.

When you try to force the deallocation of a structure that can be rebuilt, an error message is issued.

*P  Stops duplexing of the selected structure.

*R  Starts the rebuild of the selected structure. Depending on the PENDING status, the automation starts the rebuild with a different LOCATION parameter (PENDING uses the parameter LOCATION=NORMAL, otherwise LOCATION=OTHER). A rebuild with the parameter LOCATION=OTHER is only initiated for structures whose preference list contains more than one coupling facility.

There are two methods for rebuild, user-managed and system-managed rebuild. User-managed rebuild is supported for all release levels. System-managed rebuild is only available with systems that have z/OS 2.8 or above. It must have been enabled by formatting the CFRM couple data sets with the specification

```plaintext
ITEM NAME(SMREBLD) NUMBER(1)
```

System-managed rebuild is only performed when the requirements for user-managed rebuild are not met. This applies, for example, to structures without active connectors.

INGCF STRUCTURE accepts a rebuild request for structures with the condition 'No active connection exists.', but deallocates them. See the section "Structure Conditions" in the description of the INGCF command in *IBM Tivoli System Automation for z/OS Operator's Commands*.

The rebuild function works differently depending on the status of the structure (PENDING calls LOCATION=NORMAL, otherwise LOCATION=OTHER).

*S  Starts duplexing of the selected structure.
There are two methods for duplexing, user-managed and system-managed
duplexing. User-managed duplexing is supported for all release levels.
System-managed duplexing is only available when all systems in the
Parallel Sysplex have been upgraded to z/OS 1.2 or later with APAR
OW41617, and appropriate APARs listed in the CFDUPLEX PSP bucket (for
more information, see System-Managed CF Structure Duplexing,
GM13-0103-03). System-managed duplexing must have been enabled by
formatting the CFRM couple data sets with the specification
ITEM NAME(SMDUPLEX) NUMBER(1)

System-managed duplexing is only performed when the requirements for
user-managed duplexing are not met. This applies, for example, to
structures without active connectors.

Starting the duplex rebuild of a structure requires at least the policy entry
allowing the duplex rebuild of the structure. If there is no entry the duplex
rebuild is disabled. The other requirements depend on the type of the
duplex rebuild. When all connectors to a structure allow user-managed
duplex rebuild, this type takes precedence over system-managed duplex
rebuild. However, user-managed rebuild also requires at least one active
connector. Thus, when the operator starts the duplex rebuild for a structure
allowing user-managed duplex rebuild as well as system-managed rebuild
but without having active connectors, XCF tries to initiate a
system-managed duplex rebuild. System-managed duplex rebuild has the
following requirements:
• System-managed rebuild must be supported by all connectors.
• The structure must be allocated in a coupling facility supporting
  system-managed duplexing and another coupling facility supporting
  system-managed duplexing must be defined in its preference list.
• The CFRM couple data set must support system-managed duplex
  rebuild and the structure must not have a policy change pending.
• The structure must be defined in the active CFRM policy when any
  connection state is not active.

INGPLEX CDS
The CDS function displays information about all the couple data sets in the
system, including details of the corresponding policies. For every CDS type that is
required by the implementation INGPLEX CDS allows the operator to:
• Switch from the primary to the alternate CDS
• Define a new alternate CDS
• Change the active policy (if applicable)

This panel header contains sysplex-related information about the system that the
INGPLEX command was executed on. The details are as follows:
• The System field shows the name of the system.
• The Interval field shows the system failure detection interval in seconds. This
  interval is the amount of time XCF lets elapse without a status update before
  assuming that the system failed.
• The OPNotify field shows the number of seconds that XCF waits before
  notifying the operator of a potential system problem.
• The Maxmsg field shows the default value for the maximum amount of
  kilobytes of message buffer space. This default value is used when MAXMSG is
  not specified on SETXCF START commands.
The Cleanup field shows the number of seconds that XCF waits for cleanup of members.

The Retry field shows the default value for the retry limit. This value is used when the RETRY keyword is not specified on SETXCF START commands.

The Classlen field shows the default length (in bytes) of messages allowed for a transport class. This value is used when CLASSLEN is not specified on the SETXCF START CLASSDEF command.

The Max CFlevel field shows the maximum CFLEVEL supported by this system. This system can connect to a coupling facility with a higher CFLEVEL than the value of Max CFlevel but would not be enabled to use any functions supported by the higher level coupling facility.

The COUPLExx field shows the COUPLExx parmlib member used for system IPL.

The SMREBLD field shows whether (value 1) or not (value 0) system-managed rebuild has been activated in the CFRM couple data set.

The Max SMlevel field shows the maximum system-managed process level supported by this system.

For each couple data set you can enter one of the following command codes to initiate an action:

*A: Allocate alternate CDS
Replaces the current alternate CDS for a selected CDS type with a new one. There are two options how to do this:
- The alternate CDS is allocated automatically by SA z/OS.
  This automatic allocation requires that spare volumes have been defined, and that one of these spare volumes is available. For details see IBM Tivoli System Automation for z/OS Defining Automation Policy.
- Specify the data set that is to be used as the new alternate CDS.
  If you specify your own data set, observe the following:
  - The data set must exist
  - It must have been formatted with the XCF formatting tool
  - It must be at least as large as the current primary CDS, which means that every value you have passed to the XCF formatting tool (for example, in the case of a sysplex CDS, the maximum number of systems supported) must be equal to or greater than the corresponding value of the primary CDS.

C: Display CHPIIDs
Displays information about the channel paths for the selected CDS type.

D: Display CDS information
Displays detailed information about the selected CDS type. This comprises the formatting parameters and the policies that are contained in the CDS, if applicable. When the CDSs of the selected type contain policies, the detail information panel provides further actions:

  D: Display policy
  Displays details about the selected policy.

  S: Start policy
  Makes the selected policy the active policy. The policy switch must be confirmed before it is executed.

*P: Switch alternate CDS to primary CDS
This action makes the alternate CDS the primary. Since an alternate CDS is
Making an Alternate CDS the Primary CDS:
In this example, the alternate LOGR couple data set is made the new primary CDS. A new alternate CDS is automatically generated.

To switch the LOGR couple data set, enter P before LOGR on the initial INGPLEX CDS command dialog panel and press Enter. INGPLEX CDS displays a confirmation panel.

Use this panel to determine how a new alternate CDS is to be created after the switch. You can either specify your own new alternate CDS or let SA z/OS create it for you. When you specify the new alternate CDS yourself, the data set must exist and must have been formatted with the XCF formatting tool. Automatic creation requires that spare volumes have been defined for LOGR couple data sets.

Pressing PF10 causes SA z/OS to generate the new alternate CDS. After returning to the CDS command dialog, refreshing the panel, and scrolling down with PF8, the panel shows that the previous alternate LOGR CDS has become the primary, and there is a new alternate that was created by SA z/OS.

Switching the CFRM Policy:
In this example, the active CFRM policy is switched.

Enter D before CFRM on the initial INGPLEX CDS command dialog panel and press Enter. The CFRM Couple Data Set Information panel is displayed.

The panel shows information about the names and locations of the CDSs. The panel also shows the parameters that were used by the formatting tool of XCF for the allocation of the CDS. The POLICY column, for example, displays the maximum number of policies the CDS can contain. The panel also shows information about the policies in the CDS, for example, how many coupling facilities and structures are defined in every policy, and which policy is currently active.

To switch to the HIRPOL policy, enter S before this policy and press Enter. INGPLEX CDS displays a confirmation panel.

Displaying the Channel Paths for a CDS Type:
In this example, the channel paths for the CFRM couple data sets are displayed.

Enter C before CFRM on the initial INGPLEX CDS command dialog panel and press Enter. The CFRM Channel Path Information panel is displayed.

- The System column shows the name of the sysplex members.
- The T column (for ‘type’) indicates whether the CDS is the primary (value ‘P’) or alternate (value ‘A’).
- The DEVN displays the number of the device that the CDS resides on.
The CHPIDs column shows the status of the paths to the devices in the format `chpid=status_code`. The codes are those of the operating system. They have the following meaning:

− The path is logically and physically available and I/O on the path was successful.

* The path is physically, but not logically available. The subchannel's logical path indicator is off but I/O to the path is successful. You can use the command `VARY PATH (ddd,nn),ONLINE` to make channel path `nn` logically available to device `ddd`.

– The path is neither logically nor physically available. The subchannel's logical and physical indicators are both off for this channel path. You can use the command `CONFIG CHP(nn),ONLINE` to make the channel path logically available to all devices connected to the channel.

& The device is reserved to another path. This indicator applies to devices with the dynamic pathing selection feature.

< The path is installed but not physically available. The start subchannel request received a condition code of 3.

> The device microcode has detected an error and will not allow I/O to complete on the path.

B The path is unable to communicate. The device indicates that a busy or reserve condition exists on the path.

C A controller error occurred while accessing the device.

D A device error occurred while accessing the device.

I Intervention is required; the device is not ready.

R The path is available and the device is reserved to this path/group. This only applies to devices with the dynamic pathing feature.

T A time out has occurred; there is no response from the device. The cause of the time out is undetermined and this condition is transient.

U A storage control unit or storage director error occurred while accessing the device.

X Unable to determine the failing unit.

The SSID field displays the storage subsystem that the device belongs to.

INGPLEX IPL
With the INGPLEX IPL function you can record, view and compare the IPL information of the operating system. If a system does not behave after IPL as expected, the IPL recording function enables you to identify parameters that were changed, for example, since the last IPL. The recording function enables you to compare different IPL scenarios. INGPLEX IPL is a tool that helps to identify and resolve the cause of startup problems. The following information can be displayed:

• The selected system (or blank)
• The name of the sysplex
• The maximum number of IPLs that are stored for each system
• An indicator showing whether comments in PARMLIB members are ignored when collecting information

Use PF10 and F11 to scroll through all available columns on the INGPLEX IPL panel. SORT by column numbers is supported as well as the FIND and RFIND.
command to locate information on the panel. You can also limit the display to a particular system by specifying the system name in the appropriate entry field.

The following command codes are available:

C  Compares the complete IPL information with another IPL record. A second panel will be displayed where you can select the second record.

D  Displays detailed information about this IPL record.

E  Erases the IPL information records. This action must be confirmed.

- The **Sysplex** field shows the name of the sysplex.
- The **System** column shows the name of the system in the sysplex.
- The **IPL Timestamp** column shows the date and time of the IPL. The format is YYYY-MM-DD HH:MM converted to local time zone.
- The **Dev** column shows the IPL device number.
- The **Volume** column shows the volume serial of the IPL device.
- The **OpSys** column shows the name of the operating system, for example, z/OS or OS/390.
- The **Release** column shows the release level of the operating system.
- The **FMID** column shows the FMID of the operating system.

For further information about the panel fields refer to the online help.

**INGPLEX Dump Submenu**

The dump functions can be invoked directly by specifying the commands, or from the dump panel of the INGPLEX command selecting the appropriate command. In addition, you can invoke the dump submenu from the main panel of the INGPLEX command selecting command 7.

The dump functions that are available are:

1 **INGPLEX SDump**
   This displays the default SDUMP options of all the systems in the sysplex. The dump options can be changed locally, sysplexwide, or for particular systems.

2 **INGPLEX SVCdump**
   This displays the dump status information for each system in the sysplex. The function allows you to issue a multisystem dump of up to 15 address spaces for a single system, including their data spaces and structures.

3 **INGPLEX SLIP**
   This displays SLIP (serviceability level indication processing) traps that have been set for all the systems in the sysplex. You can view, enable, disable, or delete each trap.

**INGPLEX SDUMP:**
If you select option 1, the INGPLEX SDUMP panel is displayed.

The following command code is available:

C  **change**
   Invokes the modification panel by providing the options of the selected system as input

The fields on the command dialog panel display the following information:
INGPLEX Dump

- The **Sysplex** field shows the name of the sysplex.
- The **System** field shows the name of the system in the sysplex.
- The **Permission** field shows your authorization level.
- The **Dump options** field shows the default SDUMP options of all systems in the sysplex. For each system the following details are displayed:
  
  **Q=** Shows whether SDUMP quiesces the system while dumping the contents of the SQA or CSA.
  
  **TYPE=**
  Causes SVC dump to dump the cross memory address spaces that the caller has when SVC dump gets control (XMEM) or when the error causing the dump occurs (XMEME).
  
  **BUFFERS=**
  Shows the reserved storage exclusively used by SVC dump. This storage can be used when capturing the contents of the common area storage.
  
  **MaxSpace**
  Shows the maximum amount of virtual storage that SVC dump can use to capture volatile virtual storage data, summary dump data, and component-specific data before writing the dump to DASD.
  
  **MsgTime**
  Shows how long (mm) the message IEA793A is shown at the console. When the system deletes the message, it also deletes the captured dump.

The FIND and RFIND commands are supported.

If you specify command code C, a modification panel is displayed, which allows you to modify all SDUMP options. You can also delete SDUMP options. After entering your changes you can set the new options for:

- The selected system
- All systems in the sysplex
- Selected systems in the sysplex

To set the options press the appropriate F-key. If you want to modify selected systems in the sysplex, you are prompted for the systems that the SDUMP options are being changed on. To reset the options to the state when the modification panel was invoked press F5 Undo all.

**Note:** The user must be authorized to change any SDUMP option. The authorization can be any of those that are used for controlling coupling facilities and couple data sets.

For further information about the panel fields refer to the online help.

**INGPLEX SVCDUMP:**
With the INGPLEX SVCDUMP command you can display the dump status information of each system in the sysplex. The INGPLEX SVCDUMP function allows you to issue a multisystem dump of up to 15 address spaces of a single system including their data spaces and structures.

On the INGPLEX SVCDUMP target system selection panel:

- The **Sel** field lets you select a system that a dump should be taken for.
- The **System** field shows the name of the system having joined the same XCF group the operator is logged on to.
For further information about the panel fields refer to the online help. After selecting a system and pressing Enter, the INGPLEX SVCDUMP Address Space Selection panel is displayed.

If you select the VTAM address space and the WATS address space, which is a user, press Enter and then PF5, the INGPLEX SVCDUMP Address Space Detail panel is displayed.

The VTAM address space has some data spaces (D), one list structure (L) and some XCF group members (M). TSO user WATS has nothing.

The following command codes are supported:
- D  Deselects the previous selection.
- S  Selects a local address space, data space, structure, or XCF group member address space for the SVC dump.

If you press PF5, the dump option selection panel is displayed. It shows the default dump options that are set on invocation. After specifying the dump title, press F5 to issue the dump. When the dump is taken, the function returns to the address space selection panel with all selections cleared.

The SORT, FIND and RFIND commands are supported for selection panels only. For further information about the panel fields refer to the online help.

**INGPLEX SLIP:**

With the INGPLEX SLIP command you can display serviceability level indication processing (SLIP) traps being set at all systems in the sysplex. With INGPLEX SLIP you can view, enable, disable, and delete the SLIP trap defined in the sysplex.

The following command codes are available on this panel:
- +  Shows the settings of the SLIP trap.
- -  Hides the settings of the SLIP trap.
- D  Disables the SLIP trap.
- E  Enables the SLIP trap.
- R  Deletes the SLIP trap.

The SORT, FIND and RFIND commands are supported.

**Note:** The user must be authorized to enable, disable, and delete a SLIP trap. The authorization can be any of those that are used for controlling coupling facilities and couple data sets.

For information about the panel fields refer to the online help.

**How to Work with INGSTR**

If you enter the INGSTR command without any parameters, a panel with all the coupling facilities of the sysplex is displayed.

The structure display shows all structures regardless of their allocation status or the coupling facility. You can limit the display by specifying a structure name pattern.
How to Work with INGSTR

- The **P** column indicates whether a policy change is pending for the structure. Rebuilding the structure, if possible, will remove the pending status.
- The **D** column shows what type of duplexing is supported:
  - **U** Indicates that user-managed duplexing is supported.
  - **S** System-managed duplexing.
  - **B** Both, where user-managed is preferred when possible.

  Note that, for performance reasons, this status does not include a check of the SMDUPLEX flag in the CFRM couple data set. However, this flag is checked when you use command code **S**.
- The **Old** column shows where the structure is allocated or where it was first allocated when it is being duplexed.
- The **New** column shows where the duplexed structure is allocated.
- The **Pref. Location(s)** column shows the locations where the structure should preferably be allocated. When the structure is allocated to the preferred coupling facility this column is blank. A “*” in front of the coupling facility name (or names) indicates that the structure does not allow XCF to perform a reallocation.
- The following command codes are available:
  - **D** Display details of the selected structure.
  - **F** Force the deletion of the selected structure.
  - **P** Stop duplexing of the selected structure.
  - **R** Rebuild the selected structure.
  - **S** Start duplexing of the selected structure.

When the **Include unallocated** option is set to YES, all structures that are defined in the policy are shown. When the **Include condition** option is set to YES, the structure’s current condition is also shown. Specifying this option increases the response time required to build the display.

Depending on the status of the CFs and the systems in the sysplex you can use the PF10 key to perform the XCF REALLOCATE command to move the allocated structures to their preferred location. When you press PF10 to move the structures, a panel asking you to confirm the action is displayed. Press PF10 to confirm the action, or PF11 to cancel the reallocation.

**Note:** You must be authorized to perform the FORCE, REBUILD, START, STOP, or REALLOC action.

---

**Monitoring and Controlling Guest Machines and Logical Partitions**

SA z/OS provides support for you to use processor operations (also known as ProcOps) to control and monitor:
- Guest machines running under VM, see “VM Second Level Systems Support”
- The logical partitions of your System z and zSeries processor hardware, see “LPAR Management Functions” on page 168

**VM Second Level Systems Support**

This feature provides processor operations support to control and monitor guest machines running under VM.
VM Second Level Systems Support

Processor operations allows an operating system to be IPLed into a processor, amongst other facilities. One such operating system is VM. Other operating systems can be IPLed within VM as guest machines. Of particular interest are LINUX guest machines, but MVS, VSE, and even VM guest machines may be possible. (Lower levels of guest machines are not considered). Thus SA z/OS offers an effective way to enter commands to and receive messages from such a guest target system in order to validate that it had IPLed correctly, or that it is behaving correctly.

With second level guest machine support you can:
• Capture messages issued by the guest machine itself and route these back to the ProcOps process for display or automated processing, or both
• Send commands to the guest machine from ProcOps, either as operator requests or automated actions

Processor Operations Dialog
The processor operations dialogs flow provides you with information about the guest systems hierarchy (see Figure 39):
• Hosting systems are flagged on the Status Summary panel, ISQESUM.
• The ProcOps Service Machines (PSMs) of a hosting target system can be displayed by selecting the Target System Summary panel, ISQETARG, and the following PSM panel, ISQETRGG. The Target Hardware Summary panel for the PSM shows the initialized guest systems by PSM.
• You can display the hosting target system of a guest system by selecting the Target Hardware Summary panel, ISQEHARP.

Figure 39. Processor Operations Panel Flow

Status Summary Panel (ISQESUM):
The Status Summary Panel, ISQESUM (Figure 40 on page 162), displays status information about processor operations target systems.
Targets that host other target systems are marked with (*), that is, they are hosting systems.

You can enter the command codes that are shown at the top of the panel in the Cmd field for a system to invoke related commands.

You can also use the PF keys that are given at the bottom of the panel to navigate to related panels.

**Target Hardware Summary Panel (ISQEHARP):**
Selecting the Target Hardware Summary panel, ISQEHARP, displays different fields depending on whether the target hardware is a ProcOps Service Machine (PSM) or not:

- [Figure 41 on page 163](#)
- [Figure 42 on page 163](#)

You can also use the PF keys that are given at the bottom of these panels to navigate to related panels.
In Figure 41 PSM in the Type field indicates that the processor is a ProcOps Service Machine (PSM) and that the initialized target systems are guest systems. For PSMs this panel:

- Displays VM Host System Name
- Does not display the Mode, Model, or Serial Number fields

For real hardware, the panel (Figure 42) displays:

- The Mode, Model, and Serial Number fields
- Initialized Target Systems that are hosting target systems are flagged with (*)
**Processor Operations Dialog**

**Target System Summary Panel (ISQETARG):**
Selecting the Target System Summary panel, ISQETARG, displays a panel similar to [Figure 43](#).

<table>
<thead>
<tr>
<th>ISQETARG</th>
<th>SA z/OS Proc-Ops Target System Summary</th>
<th>Updates: Dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Target System</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>KEY6____ (*)</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>O.S.</td>
<td>MVS</td>
<td></td>
</tr>
<tr>
<td>Load Profile</td>
<td>INITIALIZED</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>EXCEPTIONS</td>
<td></td>
</tr>
<tr>
<td><strong>Target Hardware</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>YORAMA</td>
<td>Mode: LPAR</td>
</tr>
<tr>
<td>Attention</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Path Status</td>
<td>ACTIVE</td>
<td></td>
</tr>
<tr>
<td><strong>Target Hardware LPAR</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>KEY6</td>
<td>Mode: ESA</td>
</tr>
<tr>
<td>Image Profile</td>
<td>KEY6</td>
<td></td>
</tr>
<tr>
<td>Last Significant Message:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Enter=Static  PF1=Help LF3=Exit  PF6=Roll  PF7=Oper List  PF8=Guests  PF9=Target Hardware  PF11=Path Detail  PF12=Quit

**Figure 43. Processor Operations Target System Summary Panel (ISQETARG)**

If the target system is a hosting system:
- (*) after the Name field indicates that this is a hosting system
- PF8=Guests is displayed: Pressing PF8 displays the ISQETRGG panel.

**Guests Summary Panel (ISQETRGG):**
Selecting the Guests Summary panel, ISQETRGG, displays a panel similar to [Figure 44](#).

<table>
<thead>
<tr>
<th>ISQETRGG</th>
<th>SA z/OS Proc-Ops PSMs/Guests Summary</th>
<th>Updates: Dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VM Host System Name:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Defined ProcOps Service Machines:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSM01</td>
<td>PSM02</td>
<td></td>
</tr>
<tr>
<td><strong>Initialized Target Systems:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSM01/Linux1</td>
<td>PSM01/Linux2</td>
<td>PSM01/Linux3</td>
</tr>
<tr>
<td>PSM02/VSE1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Enter=Static  PF1=Help LF3=Exit  PF6=Roll  PF7=Up  PF8=Down  PF12=Quit

**Figure 44. Processor Operations PSMs/Guests Summary Panel (ISQETRGG)**

This panel displays the guest (that is, hosted) systems for a target system that is a hosting system. The guest systems are displayed for each associated PSM with the following:

**VM Host System Name**
- Displays the name of the hosting system.
Defined ProcOps Service Machines
Lists PSMs defined in the control file to run on the hosting system.

The number of PSMs per hosting VM system is limited to 12. (This is checked by the configuration dialog.)

Initialized Target Systems
Lists target systems currently associated with the hosting system. The target systems are identified by PSM/Target.

The number of guest systems being controlled by a PSM is limited to 60. (This is checked by the configuration dialog.) Therefore the total number of guest systems running on a VM host is limited to 720.

PATH Detail Panel (ISQEPNVC):
Selecting the PATH Detail panel, ISQEPNVC, displays a panel similar to Figure 45.

---

---

For PSM hardware TCPIP address, Message Server port, Command Server port, IP stack, and Security resource name are displayed.

Security Resource Name
Identifier used for SAF checking (for example, by RACF®).

PSM Console Mode
Normally the PSM runs disconnected. In this mode, its regular, unsolicited messages are not seen, even though they are recorded by the Logger thread in its log data set. To assist problem determination, these regular messages are captured by the Message Handler, queued to the Message Queue, and routed to the current ProcOps focal point by the Message Server. They should then be seen in the NetView log.

When a user that is logged on to the PSM issues a PSM request, a CP command or a CMS command, the responses are always displayed to the user. So, if a user logs on to the PSM no unsolicited messages are seen. These unsolicited messages can be made visible by issuing the following command on the VM console:
PSM Console Mode

CP SET VMCONIO OFF

If the Message Handler is restarted for any reason (for example, when the first guest machine on the host is initialized using the ISQXIII command), the unsolicited messages to the VM console may be again lost. Reissue the CP SET command to restore message display at the VM console.

If an ISQTRACE command is issued that sets any thread trace to the ON state, the messages issued by the PSM are automatically displayed at the VM console, and not sent to NetView. This is to avoid possible message loops, because each message sent may cause additional messages to be created.

Note that if indentation of messages is active, the PSM messages returned to NetView also appear indented. If any PSM message spills to another line, that new line is a separate message sent to NetView.

If an ISQTRACE command is issued, and all traces are set to the OFF state, the flow of unsolicited messages is once again routed to NetView. This means that the effect of any earlier CP SET VMCONIO OFF command is then lost.

When messages are displayed on the VM console, they may also be spooled to a VM output file or a reader file. This includes any trace messages.

When a user is logged on to the PSM virtual machine, the following points should be noted.

- If messages are sent to the VM console they may cause the PSM to halt execution temporarily when or shortly after the screen fills with messages. Consider entering the CP TERMINAL MORE command to control the effect of this (possibly as a command in an initial EXEC that invokes the PSM Control Program). For example, issuing CP TERMINAL MORE 0 0 means that the VM console never locks up when regular messages are issued.

- Some messages are highlighted and held. For example, the XAUTOLOG command causes such messages. When the screen fills (even if CP TERMINAL MORE has been issued) the session goes into HOLDING mode and requires operator intervention.

- Any CP command can be executed by prefixing it with CP. Any CMS command can be executed by prefixing it with CMS. However any command is executed by the Command Server thread and serializes that thread. If such a command requires an extended period of time to execute it can prevent the Command Server thread from responding to requests. If a full screen command (such as XEDIT) is entered, all threads of the PSM Control Program stop executing until the user leaves the full screen command.

Starting the PSM Control Program

The PSM Control Program runs in its own virtual machine. To start the PSM Control Program manually, logon to the virtual machine in the usual way. The PSM Control Program can be started by simply entering its program name, as selected by the programmer at installation time. For example, if the programmer named the program ISQPSM, simply enter this in response to a Ready prompt:

Ready; T=0.04/0.05 19:52:14
isqpsm
ISQMA0001I PSM Main Thread entered
ISQMA0101I Thread 3 started as CSERV
ISQMA0001I Waiting for Command Server thread to initialize
ISQCS0710I Command Server thread entered
ISQMA0101I Thread 4 started as CNSERVER
ISQMA0101I Thread 5 started as LOGGER
Starting the PSM Control Program

Initially PSM messages are displayed on the VM console. These may be captured to a disk file by spooling the console file to a reader file. To initiate spooling enter a CMS SPOOL command; for example:

```
spool console start *
```

Then start the PSM, as above. When you wish to create a complete file in the reader, enter a command:

```
spool console close
```

Spooling continues, and you may use the above spool console close command to create successive reader files. When spooling is to stop, enter the command:

```
spool console stop
```

(This does not write the remaining records to a reader file.)

If the above commands are to be entered while the PSM Control Program is running, the spool commands should be prefixed with cms:

```
cms spool console close
```

To automate the process of starting the PSM Control Program when the PSM virtual machine starts, add the ISQPSM command to the PROFILE EXEC of the PSM virtual machine.

On occasions, you may wish to start the PSM Control Program with traces active for threads that are started automatically. Before issuing the command to start the PSM, enter a CMS command of the following form:

```
GLOBALV ISQTrace.comp ON
```

Where comp is the trace option as used in the ISQTRACE request. (Usually one of CSERV, TCPIP, LOGGER or CNSERVER). Enter one command for each trace required.

## Stopping the PSM Control Program

To stop the PSM Control Program normally, first logon to the PSM virtual machine. Because the PSM virtual machine runs disconnected, the first action is to issue a BEGIN command to resume operation.

```
LOGON PSM

ICH700011 PSM LAST ACCESS AT 17:08:54 ON MONDAY, OCTOBER 20, 2003
z/VM Version 4 Release 3.0, Service Level 0202 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES: 0038 RDR, 0003 PRT, NO PUN
RECONNECTED AT 20:25:10 CST WEDNESDAY 10/20/03

begin
```

Then enter the STOPALL request. The PSM control program should then close its threads.

```
stopall
ISQCN0402I Command <stopall> entered
HCPMFS057I PSM not receiving; not authorized
```
Stopping the PSM Control Program

Sometimes it is not possible to use STOPALL to stop the PSM. In these cases, enter the HX command (halt execution) followed by B (BEGIN), repeatedly until the Ready prompt is obtained:

```
hx
CMS
b
```

The DMSRXS1419E message is normal

**Restarting the PSM Control Program**

After a PSM Control Program has stopped control returns to CMS and a CMS Ready prompt is issued to a logged on user. At this point it is advisable to enter a HX command to terminate any running threads.

The PSM Control Program can be restarted by entering its program name, for example, ISQPSM.

When the program restarts any trace setting that were set from the last execution are retained and are still in effect. If you wish to set off a trace before resuming execution enter the CMS command:

```
GLOBALV ISQTrace.comp OFF
```

Where `comp` is the trace options as used in the ISQTRACE request. Enter such a command for each trace that is to be set off.

**LPAR Management Functions**

The two SA z/OS hardware interfaces, BCP internal interface (BCPii) and Processor Operations (ProcOps), offer a common set of hardware commands to manage and control the logical partitions of your System z and zSeries processor hardware. In addition, the management of processor activation profiles and querying of CPC and LPAR information is available.

**Functional Overview**

This section describes the following functions:

- “LPAR Management”
- “Activation Profile Management” on page 169
- “CPC and LPAR Information Queries” on page 169

For a detailed description of the ISQCCMD common commands listed in the following sections, see *IBM Tivoli System Automation for z/OS Operator's Commands*.

**LPAR Management:**

This function allows you to query and set the following:

- The defined LPAR capacity
The current LPAR minimum and maximum weights
The LPAR weight capping Enabled flag
The Workload Management Enabled flag
The name of a Capacity Group Profile

The ISQCCMD common command ICNTL provides these functions.

**Activation Profile Management:**
This function allows you to query and set values of the following CPC activation profiles:
- Reset
- Image
- Load
- Group

The ISQCCMD common command PROFILE provides these functions.

**CPC and LPAR Information Queries:**
This collection of queries allows you to obtain information about:
- Consolidated CPC and LPAR details for all configured partitions
- CPC type, model, serial number, mode, status, or degraded mode details
- CPC image (LPAR) name, mode, status, active operating system type and version
- CPC neighborhood information

The following ISQCCMD common commands provide this information:
- CCNTL
- CONDATA
- CPCDATA
- ICNTL
- GETCLUSTER
- GETIINFO
- GETIPSW
- GETISTAT
- GETSINFO
- GETSDGR
- GETSSTAT
- PROFILE
- STPDATA
- TCDATA

**Enabling the Functions**
Before you can enable the LPAR management functions, make sure all hardware interface-specific installation and customization steps are complete:
- For ProcOps SNMP connections, see “Understanding the Processor Operations SNMP Interface” and the installation step “Preparing the Hardware” in IBM Tivoli System Automation for z/OS Planning and Installation.
- For BCPii connections, see “Understanding the BCP Internal Interface” and the installation step “Preparing the Hardware” in IBM Tivoli System Automation for z/OS Planning and Installation.

**Processor Operations:**
LPAR Management Functions

The common command sets are implemented with the ISQCCMD command, the ProcOps common hardware command interface.

If you are a ProcOps user, nothing extra needs to be enabled. You must however do the following:
1. Define your processor hardware and LPARs to use the SNMP connection type and make ProcOps-specific definitions for each of your systems in the SA z/OS policy database. Select the correct processor LPAR for each system.
2. Add PROCOPS to the TOWER.SA statement in the NetView style sheet, for example:
   
   TOWER.SA = SYSOPS PROCOPS

BCP Internal Interface:
With the BCP internal interface, you use either GDPS® or the SA z/OS Parallel Sysplex enhancement functions, or both. You must do the following:
1. Define your processor hardware and LPARs to use the INTERNAL connection type.
2. Make sure each of your systems in the SA z/OS policy database has the correct processor and LPAR.
3. In a sysplex environment or system cluster, you also need to make sure that all systems that need to communicate with each other over the BCPii are members of the same SYSPLEX or STANDARD group (or both). You must therefore define corresponding SA z/OS groups of type SYSPLEX or STANDARD, or both.
4. If you want to enable the LPAR Management function, add PROCOPS to the TOWER.SA statement in the NetView style sheet, for example:
   
   TOWER.SA = SYSOPS PROCOPS

No other ProcOps-specific definitions are needed.

You do not need to start ProcOps to use the ISQCCMD command for LPAR management functions over the BCPii.

Usage Considerations
Bear in mind the following:

- If you activated a ProcOps FP environment for SNMP-connected processor hardware without enabling the BCPii for GDPS or the Parallel Sysplex enhancement applications, you can use the ISQCCMD LPAR management functions like any other ProcOps commands.
- If you have defined and activated a BCPii environment for GDPS or the Parallel Sysplex enhancements, without defining and starting ProcOps, you can use the ISQCCMD LPAR management functions that are documented in this chapter, but no other ProcOps commands or functions. As the only prerequisite, the PROCOPS SA z/OS subtower must be enabled.
- If you have defined both the ProcOps FP and the BCPii on the same NetView, operation in parallel is supported. The following rule applies:
  - If a ProcOps or LPAR management command that is valid for both connection types is entered with a processor name or target system name as a parameter, the ProcOps resource tables are searched first. If the resource name cannot be found in the ProcOps tables, the BCPii tables are then scanned to resolve the name.
Monitoring and controlling the resources of a Blade Extension (zBX) in a zEnterprise ensemble environment

SA z/OS provides support for you to use processor operations (also known as ProcOps) to monitor and control the resources of a zBX unit attached to a zEnterprise CPC, defined as a zEnterprise ensemble node. For more general information about zEnterprise and zBX, please refer to the publication "Introduction to Ensembles" (GC27-2609-xx) available on IBM Resource Link.

With processor operations, the following zBX resources can be managed:
- Blade Centers
- Blades
- Blade based resources
  - Virtual Hosts
  - Virtual Servers
  - Workloads

Preparation Work

Before you can manage zBX resources with processor operations, you must define and set up a connection between processor operations and the HMC used for zEnterprise ensemble management. This is described in the following chapters of *IBM Tivoli System Automation for z/OS Planning and Installation*:
- Planning the HW Interfaces - 'Understanding the Processor Operations HTTP Interface'
- Installing z/OS on Host Systems - 'Step: Preparing Ensemble HMC Communication'
- Setting up the HMC for use with SA z/OS

Now, the ensemble connection must be defined to processor operations using the SA z/OS Customization Dialog. This is described in *IBM Tivoli System Automation for z/OS Defining Automation Policy*. Use the chapter 'Entry Type References for Processors and for Ensembles' for more information.

After the ensemble connection is defined to SA z/OS and the zEnterprise ensemble HMC is prepared for processor operations, as explained here, use the SA z/OS Dialog Build function to generate the ACF to be used by the SA z/OS Manager and Agent components. Make sure, all necessary processor and ensemble entry definitions are included in this build.

zBX Resource Monitoring and Controlling

After successful SA z/OS initialization with an ACF containing your zEnterprise processor and ensemble definitions, use the ISQSTART ACF command to start processor operations. This establishes the processor operations environment. Depending on what you have defined in the SA z/OS Customization Dialogs: Target Systems with their associated Target Hardware, zEnterprise ensemble connections, or both; after a successful ISQSTART completion you are ready to manage all ready processor operations resources.
Starting/Stopping zEnterprise Ensemble Connections

Use the ISQXIII command to start, and the ISQXCLS command to stop communication with an ensemble HMC. Note, that in order to manage zBX resources, the ensemble HMC connection must be active. For command details refer to IBM Tivoli System Automation for z/OS Operator’s Commands or use online help.

Command Interface to monitor and control zBX resources

Use the ISQECMD command if you want to manage the zBX resources from the NCCF screen. Since this command is implemented as an API, you can also use it in your automation scripts. For command details refer to IBM Tivoli System Automation for z/OS Operator’s Commands or use the Online Help.

SA z/OS User Interfaces to monitor and control zBX Resources

There are two interfaces available, the processor operations resource status dialog and Status Display Facility SDF, which both can be used for that purpose.

If you have configured SA z/OS in your automation policy to use SDF, each time you start processor operations, SDF panels will be dynamically generated. The panels represent the active processor operations resource hierarchy and include the connection status to the defined ensembles. Whenever the status of a processor operations controlled resource changes, the SDF panel views will be updated. The new status will also be propagated in the SDF panel hierarchy. For zBX resources note, that in SDF only the ensemble connection status will be updated dynamically. From the SDF ensemble connection status views you use defined PF-keys to invoke the ensemble summary view of the processor operations resource status dialog. From there you can access the zBX views, which will show the dynamic status updates for blade centers and blades of a blade extension unit.

The ISQXDST processor operations resource status dialog

The dialog can either be called directly from the NCCF screen or may be invoked from a SDF processor operations view. For command syntax and details, refer to IBM Tivoli System Automation for z/OS Operator’s Commands or use the Online Help.

Like SDF, the ISQXDST dialog panels are updated dynamically, whenever the status of a processor operations resources changes. For ensembles the dialog provides an ensemble summary view, which allows you to navigate between all defined ensembles of your processor operations configuration.

From the current selected ensemble view, you can invoke the zBX view of an ensemble CPC node. On a single panel, the ensemble, rack, blade center, and blade status information of a zBX is displayed. You can use this panel to manage blade resources directly (activate/deactivate). Note that this affects the zBX resources running on the blades, such as the virtual hosts and virtual servers. By deactivating a blade, the dependent resources will also be deactivated. Virtual hosts, virtual servers, and workloads cannot be managed directly using SDF or the processor operations resource status dialog. Use the processor operations ISQECMD command to do this.

Using the SA z/OS Discovery Library Adapter

The SA z/OS Discovery Library Adapter (DLA) discovers automated resources and their relationships within a sysplex (that is, the scope of the automation manager) and generates output XML files. These files, often referred to as Books, conform to the Discovery Library IdML XML schema and Common Data Model.
Monitoring zEnterprise Resources

(CDM). The Books can be loaded by various products, including Tivoli Change and Configuration Management Database (CCMDB) and Tivoli Business Service Management (TBSM) products.

Overview

The SA z/OS Discovery Library Adapter (DLA), as shown in Figure 46, discovers SA z/OS configuration data, which is primarily SA z/OS resources and their relationships, for all systems that are controlled by the automation manager within a sysplex (including systems that are running a previous version of SA z/OS). It discovers this data for the following resources:

- SA z/OS subsystems (APLs)
- Application Groups (APGs)
- Monitor resources (MTRs)

Figure 46. SA z/OS Discovery Library Adapter

Discovery runs in NetView via the INGDLA command issued from a system running SA z/OS 3.2. The command can be invoked either at will by an operator, for example, after a major configuration change, or it can be performed periodically (for example, daily or weekly), driven by a NetView timer.

The process flow of the SA z/OS DLA is as follows:
1. The INGDLA command queries the automation manager to obtain the list of automated resources (APLs, APGs, and MTRs) and their relationships.
2. Next, the command makes trips to the various automation agents to obtain information about the APL resources.
3. This data is then used to compose the IdML book that describes the SA z/OS configuration.
4. The IdML book is then written into the output data set (along with a few other members that control the generation of the book).
5. The IdML book is then downloaded to a system running Tivoli Application Dependency Discovery Manager (TADDM), where it can be loaded into CCMDB, along with data from other DLAs.

The SA z/OS DLA be run in conjunction with the z/OS DLA, which provides data about operating systems and LPARs.
Using the SA z/OS Discovery Library Adapter

Once you have installed the SA z/OS DLA, you can configure it using the INGCNTL command. This allows you to set the following defaults:

- The output data set name to contain the IdML book. It must be a pre-allocated (catalogued) PDS with attribute VB=3000. NetView must have UPDATE access to it.
- The name of the member that contains the IdML data (the default is INGBOOK).
- The name of the organization.
- The codepage that is used by NetView.

For more details, see IBM Tivoli System Automation for z/OS Programmer’s Reference.

To discover your configuration data with the SA z/OS DLA, issue the INGDLA command from the NetView command line on any system in the sysplex. You can, for example, issue the command after major configuration changes or periodically, using NetView timers. See IBM Tivoli System Automation for z/OS Operator’s Commands for more details.

Once you have run the SA z/OS DLA, you will need to transfer the IdML book (that is, the member that contains the book) to a system that is running Tivoli Application Dependency Discovery Manager (TADDM). The member is generated in UTF-8 format, and can be downloaded in binary format using FTP.

It is recommended that the name of the file on the TADDM system begins with ING320 (this is also referred to as the AppCode). The AppCode is used for naming IdML books, so that DLA administrators can easily sort books by product.

How SA z/OS Uses Error Thresholds

Error thresholds influence whether SA z/OS recovers from an error situation. For applications, you can define a critical threshold for restarting these applications. This is a number of error conditions within a certain time interval, for example, five error conditions requiring restart within one hour. During a condition requiring restart, SA z/OS checks whether the number of occurrences of the condition reaches the critical threshold. If it is reached, SA z/OS does not attempt to restart a resource.

For z/OS components, such as dump data sets or log data sets, you can define thresholds to limit the frequency of how often they may be deleted after they have filled up without an action being taken or a notification being sent to the operator.

Error thresholds also determine when you should be alerted to problems. The primary use of error thresholds is to track subsystem abends and ensure that the abend and restart cycle does not become an infinite loop, but they may also be customized for other uses.

Refer to IBM Tivoli System Automation for z/OS Defining Automation Policy for information on how to define error thresholds. The following sections describe how to obtain information about them.

Displaying Threshold Settings and Occurrences

SA z/OS counts the number of times that a subsystem abends. Each time that an abending subsystem is restarted, a check is made to see whether the restart is made too often. SA z/OS allows you to specify thresholds in terms of error rates.
How SA z/OS Uses Error Thresholds

If the measured error rate reaches the critical threshold, SA z/OS stops to recover the resource and posts it to a broken status.

You have the following options to obtain information about threshold settings and to view when thresholds have been reached:

**INGTHRES**

If you want to display all defined thresholds or add, change, or delete threshold settings for a particular resource, use the INGTHRES command. This displays the related Command Dialogs panel.

**DISPERRS**

If you want to check the error counts for resources on a specific system, issue a DISPERRS command to this target system. SA z/OS displays the DISPERRS Command Dialogs panel with a list of all the errors that have been recorded for resources on the target system.

Table 13 provides details of how to use commands to display threshold settings and occurrences.

<table>
<thead>
<tr>
<th>Task</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display Threshold Settings for a Selected Resource: INGTHRES</td>
<td>Enter INGTHRES on any command line to display a panel that shows the settings for the critical, frequent, and infrequent thresholds for the specified resource.</td>
</tr>
<tr>
<td>Display Occurred Errors: DISPERRS</td>
<td>If you specify the DISPERRS command, you get the DISPERRS command dialog, which displays the number of errors for all the applications on the target system and also when an application has reached its threshold. Scroll through the panel to find the resource that has failed. If a subsystem is in the BROKEN status and it has not exceeded its critical threshold, it has probably encountered an unrecoverable error. If you view either SDF or DISPINFO for it, you should find the text of the AOF571I status change notification message that explains why the subsystem status was posted as BROKEN.</td>
</tr>
<tr>
<td>Display Occurred Errors for a Specific Resource: DISPASF</td>
<td>Although DISPERRS displays the errors that have occurred for all of a system’s resources, DISPASF lets you see detailed information about errors for a resource on a specific system. For example, if you want to see more information about the error with resource MVSDUMP on system AOC7 from the DISPERRS command dialog panel, issue the following command: DISPASF MVSDUMP TARGET=AOC7</td>
</tr>
</tbody>
</table>

How to Set or Change Error Thresholds

You can set or change thresholds for a resource, the resource group defaults, or the system defaults using the INGTHRES command dialog.

1. Enter INGTHRES resname at a command line to display the INGTHRES panel with the settings of the critical, frequent, and infrequent thresholds for the specified resource.

2. Use the following command codes against the resource to change the Critical, Frequent, or Infrequent thresholds:
   - **A** Add thresholds
   - **C** Change thresholds
How to Reset Occurrences

You can reset occurrences of threshold errors by updating records in the automation status file. To do this enter the following command:

```
ASF ID=resource_name REQ=REPL ERRORDT='' THRSHLD='' 
```

Where `resource_name` is the name of the resource that you want to reset the occurrence data for. Message AOF001I informs you that the update was completed:

```
AOF001I REQUEST "REPLACE" WAS SUCCESSFUL FOR "resource_name" 
```

Setting Up Thresholds Using the Customization Dialog

`SA z/OS` uses error thresholds to determine when to stop recovering from an error situation. You can define error thresholds to track errors for MVS components and applications. You can set error thresholds for different policy objects using either the `THRESHOLDS` or `MINOR RESOURCES` policy item (for details see IBM Tivoli System Automation for `z/OS` Defining Automation Policy).

To set error thresholds for all `MVSCOMP` resources, use the `THRESHOLDS` policy item of the `MVSCOMP` Defaults policy object. To set error thresholds for all applications, use the `THRESHOLDS` policy item of the Application Defaults policy object. If you do not specify error threshold defaults for MVS components or applications, `SA z/OS` uses the error thresholds for all monitored resources that are defined in the System Defaults policy object.

You define thresholds by specifying the number of errors in a particular time interval, for example three times in one hour.

Note: Setting a critical threshold to 1 disables the recovery process at the first occurrence of the error. This is valid independent of any time interval that is specified.

When the number of occurrences of a situation reaches the infrequent and frequent error thresholds, `SA z/OS` notifies the operator and logs a message in the NetView log. When the number of occurrences of a situation reaches the critical threshold, `SA z/OS` stops the recovery from that abend situation.

Once processing has stopped, operator intervention is required to restart it. This is because the critical error threshold is used as a trigger to stop `SA z/OS` from attempting to restart the application. After the operator has restarted the application, the thresholds will next be analyzed when the application next abnormally ends. If this occurs within the time interval specified for the critical threshold, it may result in the critical threshold being reached again. The error density will decrease with time, but this will not cause `SA z/OS` to restart the application.

For example, suppose an application has a critical error threshold of two per hour:

- The application will be restarted if it abnormally ends at 10:30.
- The application will be restarted if it abnormally ends again at 12:20.
How SA z/OS Uses Error Thresholds

- The application will not be restarted by SA z/OS if it abnormally ends again at 12:45. Operator intervention is required to restart the application.
- If an operator restarts the application at 13:05 and the application abnormally ends at 13:15, the error count will again reach two per hour and the application will not be restarted by SA z/OS.
- If the operator restarts the application at 15:00, the error count within the last hour will be zero again. The application will have to abnormally end twice within the hour in order to reach the critical error threshold again.

The primary use of error thresholds in SA z/OS automated operations is to track application abnormal ends and ensure that the abend-and-restart cycle does not continue indefinitely. You can also set error thresholds for events other than the occurrence of errors. For example, you can set thresholds to act as counters of specified events on your system.

**Thresholds for Applications**

Error threshold definitions for an application define how many abend errors can occur before a message is logged or restart processing is stopped. You define such a threshold using the THRESHOLDS policy item of an Application policy object. SA z/OS displays the Thresholds Definition panel.

If no error thresholds are set here, the application uses the thresholds that are defined in the Application Defaults policy object. Determine whether existing default threshold values are appropriate for this application. If they are not appropriate, create a unique threshold entry for the application.

**Thresholds for Minor Resources**

You can define thresholds for minor resources (such as messages or statuses) that can trigger automation for applications and MVSESA resources. You can do this using the MINOR RESOURCES policy item of an APL or MVC policy object.

For example, a threshold for MVSESA.MVSDUMP specifies how often a full dump data set should be recovered in a particular time interval before the operator gets a message indicating that something has gone permanently wrong.

Thresholds can be set that let the operator know if certain errors are occurring infrequently, frequently or have reached a critical stage where the recovery process should be ended (to avoid endless loops). This is done by specifying how many times an error must happen in a certain time period for each error situation.

**Handling Jobs That Are Not Controlled by SA z/OS**

The INGLKUP command dialog displays all active jobs, started tasks (STC), and APPC/MVS transaction programs that are not controlled by SA z/OS. You can either stop, cancel, or force these jobs. Refer to the *IBM Tivoli System Automation for z/OS Operator’s Commands* for full details.

For each job the following information is displayed:
- Job name
- Step name
- Procedure name
- Job type
- Address space ID (this is only shown when the job name is not unique)
Communicating with Other Systems

This chapter discusses the following topics:

- “Sending Commands within a Sysplex”
- “Sending Commands to a Linux Target System” on page 180
- “Terminal Access Facility Full-Screen Sessions” on page 180
- “Gateway Sessions” on page 182

Sending Commands within a Sysplex

This section provides details of the prerequisites for communicating with systems in the following environments:

- Within a sysplex: this is called a single system image.
- Within your complete enterprise: this is called a single point of control for the enterprise.

There is also a series of commands that is directly processed by the automation manager of a sysplex. You do not need a target parameter for these commands. You can retrieve sysplexwide information, such as the statuses of resources within the sysplex. For a complete list of sysplexwide commands, refer to the IBM Tivoli System Automation for z/OS Operator’s Commands.

The two environments allow you operate differently:

**Single system image**

You can use a system within the sysplex to monitor and manipulate all other systems in the sysplex. Then each system in the sysplex looks like the one you are currently working on.

**Single point of control**

If your system is the focal point system, you can use it to manipulate all the resources of your enterprise from this focal point system by issuing NetView commands. The target systems do not need to be members of your sysplex.

To exploit these features, the following prerequisites and conditions need to be fulfilled:

- **Single system image**: The target system is within the local sysplex.
- **Single point of control**: The target system has a direct gateway connection with the local system.

The following prerequisites also need to be fulfilled for both cases:

- The target is active.
- SA z/OS on the target has been fully initialized.
- The target can be reached either by XCF or RMTCMD.

SA z/OS searches your environment for the target system in the following order:

1. System name within the local sysplex
2. Domain ID within the local sysplex
3. Local sysplex name
4. Domain ID within the enterprise
5. System name within the enterprise
6. Sysplex name within the enterprise
Communicating with Other Systems

You have several choices as to how to route commands to other systems within the sysplex:

- "INGEXEC"
- "INGSEND"
- "Single System Image and Single Point of Control"
- "Cross-Domain Sessions"

**INGEXEC**

You can use the INGEXEC command as a system utility to process a specified command on the system, or systems, where the specified resource or resources reside without having to specify the systems. The INGEXEC command interrogates the automation manager to determine the list of resources that are affected. The INGEXEC command operates sysplex-wide.

For more information about the INGEXEC command, see *IBM Tivoli System Automation for z/OS Programmer’s Reference*.

**INGSEND**

This will assist you in sending a command over a semi-shared RMTCMD session (this requires some configuration).

1. Enter INGSEND without parameters at the command line of any NetView or system operations panel to display the INGSEND panel.
2. Complete the other fields and press Enter. (For more information on the individual fields, refer to *IBM Tivoli System Automation for z/OS Operator’s Commands*).

**Single System Image and Single Point of Control**

If the command you want to send is an SA z/OS command and is supported by single system image (most of them are), you can just issue it, specifying the destination system as the target. This only requires that NetView is active on both systems: VTAM is not used.

The following example shows the usage of single system image to determine the status of APPL1 on every system within the sysplex (AOCA, AOCB, AOCC, AOCD) and then to restart it on AOCC where it is BROKEN. All operations are carried out from the IPUFA system.

```
DISPSTAT APPL1,TARGET=*ALL
```

<table>
<thead>
<tr>
<th>CMD</th>
<th>RESOURCE</th>
<th>STATUS</th>
<th>SYSTEM</th>
<th>JOB NAME</th>
<th>A</th>
<th>S</th>
<th>R</th>
<th>T</th>
<th>RS</th>
<th>TYPE</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>APPL1</td>
<td>AUTODOWN</td>
<td>AOCA</td>
<td>APPL1</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>TRANS</td>
<td>--none--</td>
</tr>
<tr>
<td>-</td>
<td>APPL1</td>
<td>DOWN</td>
<td>AOCB</td>
<td>APPL1</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>TRANS</td>
<td>--none--</td>
</tr>
<tr>
<td>-</td>
<td>APPL1</td>
<td>BROKEN</td>
<td>AOCC</td>
<td>APPL1</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>TRANS</td>
<td>--none--</td>
</tr>
<tr>
<td>-</td>
<td>APPL1</td>
<td>CTLDOWN</td>
<td>AOCD</td>
<td>APPL1</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>TRANS</td>
<td>--none--</td>
</tr>
</tbody>
</table>

Type B against APPL1 on AOCC and press Enter.

Complete the command parameters (STATE=RESTART, START=YES) and press Enter to issue the command. It will be sent to the target system and executed.

**Cross-Domain Sessions**

Cross-domain sessions let you send commands to other domains using NetView RMTCMD, gateways, or Terminal Access Facility (TAF).
Communicating with Other Systems

To display cross domain sessions, from the NetView command line, type AOC to invoke the SA z/OS main menu.

From the SA z/OS main menu, type 1 (Operator Interface) and press the Enter key. This takes you to the Main Operator Menu. At the command line, type 1 (CROSS DOMAIN) and press the Enter key. This leads to the Cross Domain Menu. From this menu you can display cross-domain sessions.

Sending Commands to a Linux Target System

Use the ISQSEND command of processor operations to send a command to a Linux target system. The system previously must have been initialized successfully, using the ISQXIII command. In order to see the responses on the NetView screen, the command ISQXMON must have been used to add you to the interested operator list of the target system.

Since the Linux operating system handles commands case sensitive, you have to use the "netvasis" prefix. In the following example, the Linux command ps -x is issued to show Linux process related information. Note, that NetView suppresses the "netvasis" prefix when redisplaying the entered command.

Terminal Access Facility Full-Screen Sessions

The NetView terminal access facility (TAF) allows you to log on to multiple applications either on your system or other systems. Tivoli NetView for z/OS Command Reference describes in detail how to use TAF.
Communicating with Other Systems

SA z/OS allows you to define TAF sessions in the customization dialog so that you do not have to set them up each time you want to use them. Refer to IBM Tivoli System Automation for z/OS Defining Automation Policy for more details.

Determining the Status of TAF Full-Screen Sessions
Using the TAF Fullscreen Menu panel you can see the list of TAF fullscreen sessions that have been defined to SA z/OS for your operator ID, and determine which of these are active.

To see the list of TAF fullscreen sessions available to you:
1. From the NetView command line, type Oper. You see the Main Operator Menu.
2. On the Selection line, type 1 (CROSS DOMAIN) and press the Enter key. You see the Cross Domain Menu.
3. On the Action line, type 2 and press the Enter key. You see the TAF Fullscreen Menu.

   Note: If no TAF sessions are defined for your operator ID, you will see a line mode message
   AOF041I UNABLE TO FIND FULL SESSION ENTRIES

   AOFK2DF SA z/OS - Command Dialogs Line
   Domain ID = IPUFM --- TAF Fullscreen Display --- Date = 14/11/04
   Operator ID = OPER1 System = AOC7 Time = 16:34:53
   Cmd: I Initialize R Return to T Terminate
   Cmd Name Application System Status SRCLU Session id
   --- ------ -------- ------ -------- -------- --------
   TSO IPSFM KEY3 ACTIVE TAFFJF35
   - IMS IPSFN KEY4 INACTIVE

Figure 48. TAF Fullscreen Display Panel

   This panel shows the settings of all fullscreen sessions that are defined in the policy database.

Managing TAF Full-Screen Sessions
You can use the TAF Fullscreen Display panel to log on to other applications on the current or another system to return to the session or to terminate the session.

For each TAF session the following information is displayed:

Name Is the name of the session.

Application Is the name of the application as defined in the VTAMLST data set.

System Is the name of the system where the application runs. This name is defined in the policy database.

Status Is the current status of the TAF session.
Communicating with Other Systems

SRCLU
Is the logical unit name of the session.

Session ID
Is the session ID as assigned by NetView.

You can use the following command codes:
I  Initializes the fullscreen session.
R  Returns to session.
T  Terminates the fullscreen session.

Gateway Sessions
SA z/OS does not inform an operator through a message when an outbound
session with a remote system is established or not. Therefore, you need to monitor
session status yourself.

In SA z/OS you can monitor other systems and issue commands to those systems
using gateway sessions.

Gateways are useful because they:
• Save you the step of logging on to the NetView on the other system
• Let you communicate with another system even if you do not have an ID for
that system
• Can provide a second NetView session with a system if you have an ID for that
system

SA z/OS provides several panels that let you check your communication links
with other systems and send commands to those systems. You can:
• Determine the status of gateway sessions
• Determine the status of paths. A path uses an intermediate system to connect
two systems that are not directly linked
• Issue commands to other systems using gateway sessions

Determining the Status of Gateway Sessions
You can determine which gateway sessions have been defined and see which of
those are active. You may want to do this if you use SDF to monitor other systems,
since SDF uses gateway sessions to receive information from remote systems.

You also need to check the status of a gateway session if you want to receive
notification messages from other systems. (Notification messages are messages that
document significant actions that SA z/OS has detected or taken.) Finally, you
need to check the status of a gateway session if you want to issue a command to
another system using a gateway session.

You can obtain information on gateway sessions from the Gateway Display panel
or from SDF, if SDF has been customized to do this.

To access the Gateway Display panel, type DISPGW from any command line and
press Enter.

For further information on the DISPGW command, see IBM Tivoli System
Automation for z/OS Operator’s Commands.
Chapter 8. Solving Problems with Resources

For most operators, the key resource to be monitored is the application. You want to be sure that your enterprise’s critical jobs and subsystems are getting the service they require and are operating satisfactorily.

This chapter describes general steps that you can take if resources do not start or stop as expected. It also contains a scenario that shows how you can use SA z/OS commands to discover and react to problems affecting your important applications.

If Resources Do Not Start

If you have attempt to start a subsystem and SA z/OS does not do it, there are some things you can check:

1. Locate and examine the AOF313I message for the resource. This should clearly state why the startup request was rejected.
2. If there isn’t a AOF313I message, search your netlog for more information.
3. Use the INGLKUP resource REQ=COLLECT command to collect diagnostic information about the resource from several SA z/OS commands (such as INGAMS, INGINFO, DISPMTR, DISPSTAT, and so on). You can also specify a data set for the output of the INGLKUP command. If you do not specify a data set, the output is written to the netlog. For more details, see IBM Tivoli System Automation for z/OS Operator’s Commands.

Other issues to consider include:

- **Supporting Resources Not Up**
  If supporting resources are not up, SA z/OS tries to start them. If you wait for a short period of time they should come up and then the target subsystem will be started.
  You can check on the state of the parents through the DISPTREE display, including status, or through messages issued to the Network Communications Control Facility (NCCF) console. If for example, it turns out that a parent is in a status of BROKEN, it is not possible to start the desired subsystem until this problem is fixed.

- **Automation Flags Turned Off**
  Automation flags may be turned off for the subsystem or for just the INITSTART or RESTART actions. If the INITSTART flag is turned off, the subsystem cannot be started from a DOWN status. If the RESTART flag is turned off, the subsystem cannot be started from a RESTART status.
  To see the flag settings for a subsystem you can use either DISPFLGS or DISPINFO. DISPFLGS will show you both the current settings in force for the subsystem and, perhaps more importantly, the settings that were made that resulted in those values.

- **Trigger Not Satisfied**
  Triggers tell SA z/OS when it is allowed to stop and start subsystems. If a trigger has been linked to a subsystem, SA z/OS will not stop or start that subsystem unless the trigger says it is OK to do so. Normally this function is fine, but there may be times when you need to override it.
To find out about the trigger linked to a subsystem, use the DISPTRG command. For example, the following command shows the triggers and events that are associated with the resource TEST0:

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

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

- **The Automation Manager Automation Flag Is Off:**
  Use the INGINFO command or the INGLIST command to find out what the current setting is. If the automation flag is off, the automation manager will not send the order to start the subsystem to the automation agent.

### If Resources Do Not Stop

If you have tried to stop a subsystem and SA z/OS does not do so, check for the following:

- **Supporting resource cannot stop:**
  The supporting resources cannot be brought into the condition fulfilling the stop dependency, so the dependent resource that you want to stop also cannot stop. This is the case if the compound status indicates AWAITING.

- **Request with higher priority to have the subsystem UP:**
  Use the INGVOTE command to determine whether a higher priority request exists for the subsystem. If so, you can either reissue the stop request with a higher priority, thus overruling the other request, or cancel the other request.

- **The shutdown trigger is not satisfied:**
  Use the DISPTRG command to find out the trigger conditions that are defined for the subsystem. View the shutdown conditions that are defined for the subsystem. You can either set one or more events that bring the shutdown condition into a satisfied state or use the INGREQ OVERRIDE=TRG parameter to bypass the trigger check.

- **Automation flags turned off:**
  The automation flag may be turned off for the subsystem or the termination action. To view the flag settings, use DISPSTAT, DISPFLGS, or DISPINFO.

- **The automation manager automation flag is off:**
  Use the INGINFO command or the INGLIST command to find out what the current setting of the automation manager automation flag is. If the automation flag is off, the automation manager will not send the order to stop the subsystem to the automation agent.

- **Check history:**
  Use the INGINFO command to analyze the history of the resource.

### Availability Scenario

**Consider the following scenario:**
You get a phone call from one of your users: Resource MSCSK1 is not available. Find out why and start it.

Here are the steps to make the resource available:
1. Use the INGLIST command to find out details (especially the status information) about resource MSCSK1. Because you know the name of the resource, you can issue the command as follows on your NetView console:
   
   INGLIST MSCSK1

2. Suppose you get an output screen as shown in Figure 49 that shows that the compound status for MSCSK1 is SATISFACTORY. This results from the fact that the observed status, SOFTDOWN, fulfills the desired status of UNAVAILABLE even though you and your user expect it to be available. This needs further investigation.

   ![INGLIST Output](image)

   Figure 49. Scenario: INGLIST Output

3. To find out what votes have caused the resource to be unexpectedly unavailable (SOFTDOWN), use the INGVOTE command dialog. Enter the action code e in the CMD column next to the MSCSK1 resource in the INGLIST command dialog panel, as shown in Figure 49.

4. From the INGVOTE command dialog panel, as shown in Figure 50 on page 186, you can see that there has been an operator request (MakeUnAvailable) with the highest priority (X'005A0000') to stop the resource, which has obviously been satisfied.
5. Because this request is persistent and can only be beaten by another request with priority FORCE, you must cancel it to allow the resource to become available again. (The request obviously stems from an operator’s INGREQ command with high priority that is the strongest request you can issue with SA z/OS. See also “Automation Concepts” on page 11 for more information about the hierarchy of requests). Enter action code c in the CMD column next to the request as indicated in Figure 50.

6. On the INGVOTE command dialog panel, you see confirmation that the request has been canceled. Press PF3 to go back to the INGLIST command dialog and press the PF9 key to refresh the information for resource MSCSK1. Assume that you see the output as shown in Figure 51.

![Figure 50. Scenario: INGVOTE Output](image)

![Figure 51. Scenario: Refreshed INGLIST Output.](image)

What you now see is the compound status for MSCSK1 is AWAITING (awaiting automation) and the observed status is still SOFTDOWN, which is not what you might have wanted or expected. You want the observed and
desired statuses to be AVAILABLE. There must be something else that is inhibiting this. Maybe there are dependency relationships to supporting resources that are not fulfilled.

7. Use INGINFO (action code F) to display more information about why MSCSK1 is still awaiting automation. Scroll down the INGINFO output panel until you reach the section that provides details about relationships, similar to Figure 52 (You can also use the FIND command by entering for example, find backware.)

From this output panel you find out that MSCSK1 has a HasParent dependency on the supporting resource MSCSK2/APL/AOC9/. The current situation as you have investigated it until now using various commands is:

- An operator has made MSCSK1 unavailable by issuing a SOFTDOWN request.
- MSCSK1 is dependent upon MSCSK2.

8. You now need to find out more about MSCSK2. To do this, proceed as in Step 1 on page 185 using the INGLIST command:

INGLIST MSCSK*

You get output similar to Figure 53 on page 188
This panel informs you that MSCSK2 is also not available (its observed status is SOFTDOWN), and that this even conforms to its desired status. This is obviously not what you want, so now you must find out the reason for this.

**9.** As in Step 3 on page 185, you can use action code e to find out what votes unexpectedly caused the MSCSK2 resource to be unavailable (SOFTDOWN). SA z/OS might display a panel as shown in Figure 54.

From this panel you can see that there has been a request (MakeUnAvailable) generated by a service period (SCHEDULE) to stop the resource MSCSK2. This obviously has been satisfied.

**10.** To find out more about the service period that has been defined for MSCSK2, on the INGLIST panel, enter action code i in the CMD column next to MSCSK2. Suppose that you see a display as shown in Figure 55 on page 189. From this you can see that the resource MSCSK2/APL/AOC9 is scheduled with high priority to be down on Tuesdays between 14:10 and 16:00, and that the current time is 14:21:50, which is actually within the requested downtime.
(and indeed, the indicated date of 09/15/09 is a Tuesday).

11. To allow automation to make the MSCSK2 available and keep it available for the rest of the day, you now display an additional uptime schedule for Tuesdays, starting from now (14:26 PM) until midnight (24:00) as shown in Figure 56. This method is called a schedule override (because it overrides a schedule defined by automation policy).

   Note: Schedule overrides made by INGSCHED are permanent.

12. Using PF3, go back to the INGLIST command dialog and press PF9 to see if any changes have occurred. From Figure 57 on page 190 you can see that MSCSK2 is now available, but MSCSK1 is still SOFTDOWN, AWAITING
It is possible that an automation flag is inhibiting startup or automation for this resource.

13. Next you check the automation flags for SMCSK1. You do this by scrolling the INGLIST output from Figure 57 to the right with the PF11 key until you reach a display similar to Figure 58.

However, the result shows that MSCSK1 is startable and automation is allowed for it (Auto=YES and Hold=NO). So what else could be preventing MSCSK1 from starting?

14. There might be a trigger for MSCSK1 whose events for its startup condition are not set, thus preventing the resource from starting. Scroll horizontally in your current INGLIST output until you see the Trigger column as shown in Figure 59 on page 191.

Note: The DISPSTAT command cannot be used because it only displays service periods and triggers from downlevel SA z/OS systems.
From this panel, you see that a trigger, MSCTRG1 is connected to MSCSK1:

- MSCSK1 is dependent on MSCSK2. So MSCSK2 must be up before MSCSK1 can be started.
- The availability of MSCSK2 is controlled by a schedule. So the time of the day must be within one of the uptime schedules that are defined for this resource.
- MSCSK1 has a trigger connected to it. Among all the other prerequisites for starting this resource, at least one startup condition for that trigger must be fulfilled to MSCSK1 available.

15. In this step, you check the startup conditions for trigger MSCTRG1. You do this by entering action h (for DISPTRG) in the CMD column next to MSCSK1. This displays triggers and their events for the selected resource. In this scenario, we get the DISPTRG output shown in [Figure 60 on page 192]
In this output, you can see that trigger MSCTRG1 has a startup condition that is fulfilled if event MSCEVTUP is set. This is what you need to find out in the next step.

16. To see more details about an event (or the condition), enter action code s in the CMD column next to the startup condition (event MSCEVTUP). This produces the output shown in Figure 61.

17. From Figure 61, you can see that the event MSCEVTUP is not set (the Status column contains UNSET). This sets the event to fulfill the startup condition for trigger MSCTRG1, thus allowing the resource MSCSK1 to start. SA z/OS will now put your environment into the status you desired. After a few minutes you should be able to verify this. View the INGLIST command dialog for both resources involved (enter INGLIST MSCSK*), as shown in Figure 62 on page 193.
Both resources now have AVAILABLE for the observed and desired status and thus a compound status of SATISFACTORY.
Chapter 9. Using I/O Operations

I/O operations is a highly flexible connection environment that expands the distance for channel and control unit communication.

You can use the SA z/OS I/O operations component:

- To get a single point of logical control for managing connectivity in your active I/O configurations.
- To have information presented on the operator console that you issue commands from or from the I/O Operations ISPF dialog. You can also use the I/O operations application program interface (API) to get information via a user-written call.
- As a tool for automating these I/O configurations. I/O operations takes an active role in detecting unusual conditions and lets you view and change paths between a processor and an I/O device, using dynamic switching (the ESCON/FICON director).

To fulfill the tasks of managing and automating I/O configurations, various commands are available, like for example, a command to define a dynamic connection between ports, commands to block or permit data transfer through ports or a series of commands to display I/O configuration information.

You can issue these commands either through the ISPF dialog (see “Using the I/O Operations ISPF Dialog” on page 198) or from the operator console.

More features of the I/O operations are, for example:

- For an I/O resource that passes through a dynamic switch, I/O operations can detect and report several types of I/O configuration mismatches. It reports these mismatches when the program is started or when the configuration is dynamically reconfigured.
- You can use I/O operations as a single point of control:

  I/O operations lets you change the connectivity of I/O resources. From any one system where I/O operations is running, you may affect the I/O connectivity of a target system that I/O operations is also operating on, thereby providing a single point of control.

  I/O operations performs these functions when you enter, or send, one of its connectivity commands to one of the hosts that the program is running on. That host issues the commands to the affected hosts. The hosts that can be affected by the command depend on several factors, including the domain of the issuing host.

  With the feature of a single point of control, you can:

  - Remove the dependency on a dynamic switch
  - Manage the devices across hosts; that is, you can vary devices on or off
  - Remove and restore channel paths among system images and have the pathing status automatically adjusted
  - Manage switchable paths in a system-integrated manner

With I/O operations, you can display and control the following I/O resources:

- **Channels** of the following types:
ESCON/FICON channels
Parallel channels
Coupling facility channels
- Dynamic switches, such as the ESCON/FICON Directors
- Ports, which are the interfaces on a dynamic switch
- Control units
- I/O devices
- and other objects

Making Connectivity Changes

With I/O operations you can easily perform the following connectivity changes:
- Remove and restore channel paths: These commands not only instruct z/OS to reconfigure a CHPID, they also issue the appropriate vary path requests. In these commands, you can specify any CHPID that is defined to one of the hosts in the issuing host's group. The CHPID can be either switchable or point-to-point, ESCON/FICON or parallel.
- Remove and restore devices: These commands allow you to remove one, a range, or an array of devices from - and restore them to - one or an array of hosts in a group. These commands therefore expand the capabilities of the analogous Vary Device system commands, so that devices can be varied across system images-all within the issuing host's group.
- Use switching commands, such as:
  - Commands that affect only ports
  - Remove Switch
  - Restore Switch
  - Commands that define a chained path and remove these definitions

Display Commands To Obtain Data About I/O Resources

With such commands, you can for example, retrieve data about one or all channels (CHPIDs) defined to the host image that the command was issued on. Or you can retrieve data about one, a range, or all the devices and their paths defined to the host image. You can also display data about the addressable port(s) on one or all switches allocated, or attached, to the issuing host.

Examples for this category of commands are:
- DISPLAY CHP
- DISPLAY DEVICE
- DISPLAY PORT
- DISPLAY RESULTS

Connectivity Commands That Affect a Switch

With commands of this category you can for example permit or prohibit dynamic communication between two specified ports or between one port and all other ports on the same switch or the same ports on all the switches allocated to, or attached to, the issuing host.
Connectivity Commands That Affect a Switch

Or, you can establish or remove a static, or dedicated, connection between two ports on the same switch or the same two ports on all the switches allocated to, or attached to, the issuing host.

Examples for this category of commands are:
- ALLOW
- PROHIBIT
- CONNECT
- DISCONNECT
- BLOCK
- UNBLOCK
- CHAIN
- UNCHAIN

Display and Utility Commands Related to Connectivity Commands

With commands of this category you can for example display the specified number or all of failed vary path requests as a result of the last connectivity command.

Or you can build a list of the vary path requests that would be needed to synchronize path status with the physical settings of the specified switch.

You can also display or reset the number of seconds that the issuing hosts (I/O operations) will wait to receive votes from the I/O operations that from its voting set.

Examples for this category of commands are:
- DISPLAY VARY
- SYNC SWITCH
- RESET HOST
- DISPLAY TIMEOUT
- RESET TIMEOUT

Utility Commands for General Processing

With such commands, you can for example, acquire or release exclusive control of I/O operations command processing for a specified user ID.

Or you can instruct the switch(es) to send their log data to the host that the issuing I/O operations is running on.

Another usage of such a command would be to ensure that the issuing I/O operations's configuration map is correlated with the active configuration on the switch(es).

Examples for this category of commands are:
- GETLOCK
- UNLOCK
- LOGREC
- RESET SWITCH
Using the I/O Operations ISPF Dialog

A set of REXX EXECs is shipped with SA z/OS that – when installed as the I/O operations Interactive System Productivity Facility (ISPF) dialog – offers you a simple way to enter the same commands to I/O operations that you can enter at an z/OS operator console.

In addition to entering I/O operations commands, you can create entire switch configurations, starting with the default matrix of switch ports that is shipped with the dialog.

Detailed information about the I/O operations ISPF dialog is included in Appendix B, “Using the I/O Operations ISPF Dialog,” on page 227. The information in this section helps you get started using the dialog. For more information, refer to the online tutorial and online help panels supplied with the dialog.

Using the z/OS Operator Console for I/O Operations Commands

Some I/O operations commands can be entered at the z/OS operator console. To enter a command in the z/OS environment, use the Modify command, which can be shortened to F. Specify the I/O operations procedure name that has been created as a member in your SYS1.PROCLIB. The member name for the procedure as supplied by SA z/OS is INGEIO.

- If you did not append an z/OS identifier to the Start command when I/O operations was started, the syntax is as follows:

  /modiFy procname,I/O_operations_command

- If you did append an z/OS identifier to the Start command when I/O operations was started, the syntax is as follows. (NOTE: if the SA z/OS I/O operations component was started automatically by the automation supplied with SA z/OS, an identifier of "IO" was used by the Start command.)

  /modiFy identifier,I/O_operations_command

For example, /F IO,D N CU.C00* * instructs I/O operations to display names of, and salient data about, the port or ports that have been assigned names with the common character string 'CU.C00' on the dynamic switches allocated to the issuing instance of I/O operations.

The NetView program provides an z/OS interface that allows a NetView REXX EXEC or an operator at a NetView console to issue z/OS operator commands. For example, the following Display Switch command can be coded in a REXX EXEC or issued at a NetView console:

  MVS /F IO,D S *

(The output of this Display Switch command is displayed on the NetView console, not the z/OS operator console.)

When you use the NetView console to enter I/O operations commands, keep the following points in mind:

- Control authorization is not checked. The same I/O operations commands can be entered that can be entered at the z/OS operator console.
Using the z/OS Operator Console for I/O Operations Commands

- Commands can be entered from different consoles to the same system operator ID. You should, therefore, have a central point of control for I/O operations processing, especially for its connectivity and utility commands.
Using the z/OS Operator Console for I/O Operations Commands
Part 3. Using SA z/OS on Workstations

This part describes how to use SA z/OS on workstations. It contains the following chapter:

- Chapter 10, “Using the NetView Management Console for SA z/OS,” on page 203
Chapter 10. Using the NetView Management Console for SA z/OS

The NetView Management Console (NMC) is a part of the NetView product and is used for monitoring network resources. As SA z/OS is based on NetView, it also exploits the graphical capabilities of the NMC.

You can monitor and control your whole enterprise with the NMC. It is also possible to monitor your enterprise from a single workstation (single point of control). This is valid for system operations as well as for processor operations. For general information about the NMC see NetView Management Console User’s Guide. This chapter only describes the extensions that have been added for monitoring SA z/OS resources.

Introduction

The communication between SA z/OS and the NMC is maintained by the SA z/OS topology manager. An SA z/OS topology agent on each target system retrieves the enterprise data from the automation manager. The SA z/OS topology manager on the focal point provides the information into RODM. GMFHS takes the information from RODM and presents it in a graphical form on the NMC workstation. The following sketches the main tasks of the SA z/OS topology manager for system operations. For processor operations, the mode of operation is similar.

At initialization time, the SA z/OS topology manager retrieves SA z/OS information for one or more sysplexes from the corresponding automation managers. To do that, it contacts an SA z/OS topology agent that resides on a system within the sysplex to be monitored. The SA z/OS topology agent obtains the information from the automation manager of the sysplex and passes it to the SA z/OS topology manager.

The retrieved information includes all resources, for example, applications (type APL), application groups (type APG), systems (SYS), and groups of systems (GRP) with their mutual dependencies, and relationships. The SA z/OS topology manager stores these resources together with their dependency information within RODM.

The SA z/OS topology manager calls BLDVIEWS to build the views that are necessary for these resources to be displayed on the NMC. The definitions for the views must be supplied by the users.

This process is illustrated in Figure 63 on page 204. The arrows with broken lines represent requests, those with solid lines stand for delivery of information. This figure shows an idealized system configuration. Any constellation of systems is supported according to the abilities of SA z/OS. For example, all components can be installed on a single system.
Note: In SA z/OS, a single system can be regarded as a special sysplex with one system only.

At runtime, the SA z/OS topology manager performs the following tasks:
- It updates the NMC whenever the status of a displayed resource changes.
- It dynamically displays events like WTORs or tape mount requests on the NMC.
- Users can issue SA z/OS commands from the NMC that will be routed by SA z/OS to the appropriate system for processing.
- It resynchronizes its resource structure for one or more sysplexes with the corresponding automation managers. When this happens, the views that are effected by possible changes are rebuilt.

Multiple Focal Point Support

Typically two NMC focal points are used to survey the NMC targets. Each focal point is the backup of the other one. They always show the same information about the targets (hot backup). Both focal points have to be configured within the SA z/OS customization dialogs (> 2 GRP > NMC DEFINITIONS) in the Primary field of the NMC Focal Point Domains section.

Separate, independent heartbeats are required for multiple NMC focal points. Each focal point needs its own heartbeat (a heartbeat timer on the target system and deadman timer on the focal point system).

Starting the NMC

Before you start the NMC, make sure that the following conditions are satisfied:
- The NMC server must be up and running and must be communicating with NetView.
- In order to issue interactive commands (see “Interactive Commands” on page 213), you must have set up your NetView 3270 Management Console correctly. See IBM Tivoli System Automation for z/OS Planning and Installation for details.

After you have started the NMC, the Sign On panel is displayed. Enter the user ID and the password in the corresponding fields. The user ID must be identical to that of an active NetView host session. In the Host Machine field you must enter the address of the NMC server. If you do not know the correct address, contact your system administrator.
Starting the NMC

When you have signed on successfully, the Tivoli NetView for z/OS Management Console is displayed. The left part of that window contains the Business Tree. If you select a view from the Business Tree, it is displayed in the right part. The SA z/OS views are descendants of the Networking view, as shown in Figure 64.

![Figure 64. Tivoli NetView for z/OS Management Console](image)

Monitoring SA z/OS on the NMC

On the NMC workstation, you can see the graphical visualization of what is automated in system operations and processor operations.

For system operations, the automation manager knows about the configuration of the sysplex and the resources running in it. The SA z/OS topology manager communicates with these automation managers.

For processor operations, it is the processor operations focal point that knows about the processor operations configuration and therefore, the SA z/OS topology manager communicates with it.

In addition to the resource information retrieved from the automation managers, the SA z/OS topology manager maintains SA z/OS objects that have no counterpart within the automation manager. The following sections explain the representation of these two types of SA z/OS resources on the NMC.

Regular Resources

SA z/OS resources in RODM that also exist in the automation manager are called regular. For example, systems (SYS resource type), system groups (GRP), applications (APL), application groups (APG), and Monitor resources (MTR) are regular resources. Figure 65 on page 206 shows a network view of regular resources.
Monitoring SA z/OS on the NMC

Note that the figures in this chapter are based on the system AOC7 and sysplex SS7.

![Network View for Regular Resources](figure65)

The members of a group are represented on the NMC by subviews. For instance, if you double click on the application group SS7.AOC7.APG.CICSGRPJSHP of Figure 65 the following view will be displayed:

![Subview with Applications of an Application Group](figure66)

### Minor Resources, Anchors and Tokens

The SA z/OS topology manager distinguishes between *major* and *minor* resources. Major resources, such as applications or systems, exist independently. Minor resources are subcomponents of applications (or MVS components) that cannot exist independently, but are always associated with a major resource. Typical examples for minor resources would be the spool of a JES application or transactions of a CICS application. Minor resources have their own automation flags so that you can, for example, switch on automation for an application, but switch it off for one of its subcomponents. This is one main reason for introducing the distinction between major and minor resources.

The minor resources that are defined in the policy database and known to SA z/OS are a special type of regular resources. They are retrieved by the SA z/OS topology manager from an automation manager (either during initialization of the SA z/OS topology manager or when the information in RODM must be resynchronized with the automation manager).

In addition to regular minor resources, the SA z/OS topology manager also knows *dynamic* minor resources. Dynamic minor resources are posted to the SA z/OS...
Monitoring SA z/OS on the NMC

topology manager at runtime with the INGPOST command. You can also change
the state of an existing dynamic resource with INGPOST. For more information on
the INGPOST command, see IBM Tivoli System Automation for z/OS Programmer’s
Reference. In contrast to regular resources, you can delete dynamic minor resources
in RODM from the NMC.

To post a dynamic minor resource to the SA z/OS topology manager, you must
specify the major resource that this minor resource belongs to. When the SA z/OS
topology manager finds the major resource within RODM it connects the new
minor resource with the major resource and displays it immediately in all
(network) views that contain the major resource.

The major resource specified to INGPOST can be a regular resource or an anchor.
Anchors are special objects that exist in RODM, but not in the automation
manager. They play the role of major resources and serve two purposes:

- They are used to display dynamic minor resources that either the specified
  major resource or the specified sysplex does not exist for in RODM. See
  "'Lost&Found'-Anchors" on page 209.
- Anchors are used to display and group minor resources that derive from events
  such as WTORS and tape mount requests.

Anchors and Events
The SA z/OS topology manager is able to represent events such as WTORS or tape
mount requests on the NMC. The events are posted to it with the INGPOST
command. The SA z/OS topology manager treats them as dynamic minor
resources; these resources are called event resources. Event resources are connected
to an anchor as their major resource. All events connected to one anchor belong to
the same event type. Thus there are, for example, anchors for WTORS, and anchors
for tape mount requests.

If specified in the Customization Dialog the SA z/OS topology manager creates
one anchor per sysplex for several event types, for example for WTORS and tape
mount requests. Whenever an event of such a type is posted to the SA z/OS
topology manager, it is attached to the appropriate anchor of the corresponding
sysplex, and all views that include this anchor are immediately updated.

Thus, when a WTOR originating from system AOC7 is posted to the SA z/OS
topology manager and AOC7 belongs to sysplex SS7, this WTOR is attached as a
minor resource to the (generated) WTOR anchor of sysplex SS7 and dynamically
added to any view that includes the WTOR anchor of SS7.

In addition to this you can define your own anchors for any events in the
customization dialogs. You must post these events to the SA z/OS topology
manager with the INGPOST command.

Figure 67 on page 208 shows a view that contains WTOR, TAPE, and
SYSPLEX anchors for the sysplex SS7 with event resources attached to some of
them. It also shows the HEARTBEAT anchor that is automatically created.
The anchors and their minor resources have the following function:

**Heartbeat**

These resources represent the connection between the SA z/OS topology manager and the sysplex. Usually, two minor resources are attached to the heartbeat anchor. One icon represents the heartbeat for the sysplex (the right one in Figure 67), the other is the heartbeat for the system that is currently being used to monitor the sysplex (the left one in Figure 67).

For more information on heartbeats, see *IBM Tivoli System Automation for z/OS Defining Automation Policy*.

**Note:** When the communication between the SA z/OS topology manager and the target system(s) fails, icons are created to represent the status: you will find a heartbeat icon with a corresponding label, such as `APLX.HEARTBEAT.NOTARGET` and a sysplex group labelled for example, `APLX.GRP.NoTarget.NoCommunication`. The heartbeat icon is there to represent the state. On the sysplex group you can initiate an `INGTOPO REBUILD <system>` or an `INGTOPO SWEEPSYS <system>`.

**SYSPLEX**

Minor resources like coupling facilities, couple data sets or sysplex timers are attached to the SYSPLEX anchor if they have been defined in the customization dialogs.

Additionally, all minor resources that are not related to a specific major resource are attached to the SYSPLEX anchor.

**WTOR**

Write To Operator with Reply (WTOR) messages are z/OS messages that inform you of various conditions and require a reply of the operator. In Figure 67 one WTOR is attached to the WTOR anchor.

In the label of this WTOR, 'WTOR' is the major resource name, and 'SYS1.0001' the minor resource name. The additional information is the message content.

By defining a view that contains the WTOR anchors of all sysplexes in the enterprise, you can monitor all WTORs at a single glance. For defining views, see “Creating Views” on page 216.
Tape mount request
When a tape mount request for sysplex SS7 is posted to the SA z/OS topology manager, it is attached to the tape mount request anchor of this sysplex.

'Lost&Found'-Anchors
To post a new minor resource to the SA z/OS topology manager, you must specify the major resource that this minor resource belongs to. When the SA z/OS topology manager cannot find the major resource in RODM, it attaches the minor resource to one of several special anchors that serve to display such unconnected minor resources.

There is one such anchor at sysplex level, which collects minor resources whose major resource does not exist in RODM, but whose sysplex is known to the SA z/OS topology manager. If even the sysplex is unknown, the SA z/OS topology manager attaches the minor resource to an anchor with the prefix 'Lost&Found'. There are 'Lost&Found'-anchors for WTORs, tape mount requests, heartbeats, and for sysplex related resources.

Figure 68 displays a view that comprises the 'Lost&Found'-anchors.

Tokens
A token is an object that is generated automatically by the SA z/OS topology manager. The date and time when the SA z/OS topology manager was initialized is appended to the name of the token. In the Data1 field of the resource properties, you can find the original ACF token or the processor operations token of the monitored system. This can be used for diagnostic purposes.

Properties of SA z/OS Objects on the NMC Workstation
All resources are represented by icons. Dependencies are displayed as lines; no distinction is made between start and stop dependencies. Association of a minor resource to a major resource is also represented with a line. The state of the resources is indicated by the color of the icon. The colors are mapped to the compound states of the resource as follows:

<table>
<thead>
<tr>
<th>Color</th>
<th>Compound State of Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>GREEN</td>
<td>Satisfactory</td>
</tr>
<tr>
<td>LIME</td>
<td>InAutomation</td>
</tr>
<tr>
<td>CYAN</td>
<td>Awaiting</td>
</tr>
<tr>
<td>WHITE</td>
<td>Inhibited</td>
</tr>
<tr>
<td>YELLOW</td>
<td>Degraded</td>
</tr>
<tr>
<td>PINK</td>
<td>Denied</td>
</tr>
<tr>
<td>RED</td>
<td>Problem</td>
</tr>
</tbody>
</table>
### Monitoring SA z/OS on the NMC

<table>
<thead>
<tr>
<th>Color</th>
<th>Compound State of Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>GREY</td>
<td>Status not known (NMC specific)</td>
</tr>
<tr>
<td>BLACK</td>
<td>Resource deleted in RODM (NMC specific)</td>
</tr>
</tbody>
</table>

For more information on the compound status, see “Compound Status” on page 47.

The labels of the icons consist of three (or four) parts:

1. The sysplex name as defined in the INGTOPOF file.
   
   For more information about the INGTOPOF file, see IBM Tivoli System Automation for z/OS Planning and Installation.

2. A derivative of the major resource name.
   
   This part of the label differs from the resource name within the automation manager in that the order of its components is reversed and the slashes are replaced by dots (for example, the standard resource name JES2/APL/SYS1 becomes SYS1.APL.JES2). The reason for this reversal is explained in “Names” on page 217.

3. Possibly a minor resource name.
   
   The minor resource name of the JES2 spool in Figure 65 on page 206 is SPOOL. When a minor resource name contains more than one part, the separator slashes are also replaced by dots, but no reversal of the components takes place (for example, TRAN/ABC123 would become TRAN.ABC123).

4. Additional information.
   
   With major resources, this additional information is essentially the compound status. In most cases, it is supplemented by a suffix that indicates whether the status refers to starting or stopping the resource. Thus there is an INHIBITED_START state and an INHIBITED_STOP state. The SATISFACTORY compound state is replaced by the pair AVAILABLE and UNAVAILABLE.

   For the minor resource in Figure 65 on page 206, the additional information is ‘OK’.

For more information on the labels, see “Names” on page 217.

If you have the appropriate CPE profile setup for your user ID, you can view and possibly edit the **Resource Properties** of the displayed resources (see also *NetView Management Console User’s Guide*).
For regular major resources, the **Data 1** field contains detailed status information, and the **Data 2** field can contain a comment, for example from an operator. For dynamic minor resources (see "Minor Resources, Anchors and Tokens" on page 206), the **Data 1** field corresponds to the *message* parameter of the INGPOST command, **Data 2** corresponds to the *userdata* parameter of the INGPOST command, and **Data 3** can be changed by using the B3 parameter of the INGPOST command; see *IBM Tivoli System Automation for z/OS Programmer’s Reference* for more information on INGPOST. **Data 4** is not currently used.

SA z/OS information is presented on the NMC in *network* views or in *exception* views. Network views represent the topology of resources, that is, grouping of resources and their connections. Figure 65 on page 206 is an example for a network view. Exception views display resources that have certain properties in common.

**Dynamic Behavior of the SA z/OS Topology Manager**

In a running SA z/OS enterprise, you can imagine scenarios where new systems are IPLed while the SA z/OS topology manager is already running, or new systems are added to a sysplex and so on. The following sections describe how the SA z/OS topology manager behaves in these events.

**SA z/OS Topology Manager Initialization: INGTOPO INIT**

If the systems are set up accordingly, SA z/OS automatically issues the INGTOPO INIT command after it has received the completion code of the start of the NetView MultiSystem Manager. If this is not the case, you have to start the SA z/OS topology manager manually by issuing the INGTOPO INIT command.

With the INGTOPO INIT call, the SA z/OS topology manager queries the INGTOPOF file and contacts each of the specified target systems (the specified
Dynamic Behavior of the SA z/OS Topology Manager

sysplexes or the processor operations focal point). For the contacted target systems, it creates the corresponding objects in RODM.

System Detection

When a system that SA z/OS is installed on becomes active, it sends a signal to the SA z/OS topology manager. The SA z/OS topology manager, which is already running, checks whether the system belongs to either a sysplex or a processor operations focal point it knows (that is, one that is defined in the INGTOPOF file) and whether this sysplex or processor operations focal point was incommunicative until now. If both conditions are fulfilled, the SA z/OS topology manager performs the initialization process for this sysplex or processor operations focal point. In all other cases the SA z/OS topology manager ignores the signal. Note that these signals never reach any operator when the SA z/OS topology manager is not present.

The Heartbeat Function

When the SA z/OS topology manager makes contact with a system that can act as the representative of a sysplex, it launches a heartbeat timer on that system and a deadman’s timer for that sysplex on the focal point system. The heartbeat timer sends information to the focal point system at a regular interval to indicate that the target system is still in good health. This causes the deadman’s timer on the focal point to be reset. If the reset does not occur before the deadman's timer expires, the SA z/OS topology manager assumes that the current representative cannot respond. Processing in such a case is described in "Takeover when a System Goes Down."

Takeover when a System Goes Down

If the SA z/OS topology manager needs to assume that the sysplex representative cannot respond, it tries to find another one according to the specified systems in the in the SYSPLEX statement in the INGTOPOF file. Therefore the sequence of the specified SYSPLEX and PROCOPS statements and the sequence of the specified systems is relevant. If it finds a new representative, communication continues with the new one and representation on the NMC continues. If it does not find one, it considers communication with the corresponding sysplex to be interrupted; all resources of the sysplex assume the color representation of the status Unknown.

Issuing Commands from the NMC

In order to issue a command against a resource, click on this resource with the right mouse button to open a context menu. This menu contains only those system operations or processor operations commands that are applicable to the corresponding resource. Some SA z/OS commands are not provided because they cannot be reasonably connected to a resource.

When the state of the resource is affected by the command, the state information (event component of the label, color of the icon) is actualized on the NMC.

The following sections describe the different command types.

Direct Commands

Some commands, for example INGINFO, need no user specified parameters. They are immediately executed after having been selected in a command submenu. Their output is displayed in the log of the NMC.
Dialog Commands

Most SA z/OS commands accept or need parameters that you must specify. When you select such a command from a submenu, a window will open where you can enter the appropriate values. Their output is displayed in the log of the NMC.

If you select, for example, the command INGTHRES for the resource SS7.AOC7.APL.VTAMJSHIP in the view of Figure 65 on page 206, the appropriate parameter window is opened as shown in Figure 70.

![Figure 70. Parameter Window for INGTHRES](image)

All parameter windows consist of two parts. The upper part contains the fields where you can enter the parameter values. The lower part displays the operator command that is generated from your input in the upper part. All you have to do is to enter the required values or choose the required option in the parameter fields (which has already been done in Figure 70) and issue the command by clicking on one of the two send buttons.

You can edit the command string in the Command string box at the bottom of the window. However, this is not necessary and may even lead to errors if you do not have a thorough understanding of the INGRTCMD command. "The INGRTCMD Command" on page 215 contains a brief summary of the INGRTCMD mechanism.

Interactive Commands

There are commands that require a continued dialog before they can be processed. Such commands are routed to a NetView 3270 Management Console and are processed there; this presupposes that you have set up the 3270 console correctly.

The INGREQ STOP command provides an example:
After clicking on the resource SS7.AOC7.APL.VTAMJSHP in Figure 65 on page 206 and selecting the entry *SA Operations Commands*, a submenu is opened that includes the **INGREQ Stop** entry. If you select this command, the request is forwarded to a NetView 3270 console, which would look as follows:

![NetView Console with INGREQ Panel](Figure 71. NetView Console with INGREQ Panel)

After pressing Enter you are presented with a list of the resources affected by the command (SS7.AOC7.APL.VTAMJSHP and SS7.AOC7.APG.CICSGRPJSHP). If you confirm the request the resources are stopped. The view of Figure 65 on page 206 is actualized, and now looks as follows:
Issuing Commands from the NMC

Note the modified event components in the label of the resources SS7.AOC7.APL.VTAMJSHP and SS7.AOC7.APG.CICSRPJSHP.

Note: Depending on your setup, at the initial start of the NetView 3270 Management Console, triggered by an SA z/OS command that needs to be transferred to the NetView 3270 Management Console, you might be confronted with messages, or even problems (e.g. failing logon), that are not related to the original SA z/OS command. Handle these messages or problems as appropriate and retry the command.

The INGRTCMD Command

When you select a resource and then choose an operator command, a template of that command is generated. In this template, a place holder value is specified for all parameters of the command that an attribute of the selected resource can be assigned to as its value. The place holder points to this attribute. The place holders all have the format ?name?. Thus, for example, ?RSC. refers to the resource name (which would be VTAMJSHP/APL/AOC7 in our example) of the object you clicked on.

Before the command is forwarded to the appropriate target system, these place holders must be replaced by real values. This is done by the command INGRTCMD. In order to be able to substitute the place holders, INGRTCMD must be passed the ID of the object whose attributes are to supply the parameter values, so that the format of the command is as follows:

```
INGRTCMD object_id command_string
```

In the parameter window, the object_id is specified in the form %OBJECTID%, which is a reference to the RODM object ID of the selected resource. The command string displayed in Figure 70 on page 213 would thus be converted by INGRTCMD into

```
INGTHRES VTAMJSHP/APL/AOC7,CRIT=(2,02:00) outmode=line
```

and would be routed to the system representing the sysplex SS7.
Creating Views

The RODM objects representing SA z/OS resources must be included in a view in order to be displayed on the NMC. The definition of these views is up to you. The SA z/OS topology manager, however, offers some support for executing view definitions and updating the views whenever necessary. In order to exploit this feature, you must specify your view definitions in the format required by the NetView BLDVIEWS tool. Some familiarity with this tool and with the terminology and class structure of RODM (see Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer’s Guide) will be assumed in the following description.

The files containing your BLDVIEWS view definitions can be introduced to the SA z/OS topology manager in the INGTOPOF configuration file (for information on the INGTOPOF file, see IBM Tivoli System Automation for z/OS Planning and Installation). When the SA z/OS topology manager is started, it will, after constructing the SA z/OS objects, create the views in RODM by calling BLDVIEWS for all the definition files that it finds in the INGTOPOF file. BLDVIEWS will create a view object in RODM for every definition and connect every resource that meets the selection criteria specified in the definition with this view object.

Note: You can also call the BLDVIEWS tool independently of the SA z/OS topology manager to update views or create further views. However, these will not be automatically updated when new major resources are created in RODM.

Network Views and Exception Views

NetView provides different types of views. For displaying SA z/OS resources, only network views and exception views are used.

Network views are intended to display the structure of a network. Therefore, all relationships between resources are displayed on the NMC. Network views are also static with regard to their structure; unless the link between a view element and the view object is removed or a new link is established, they always contain the same elements.

Exception views, on the other hand, display resources that have certain properties in common, for example, all resources that are in an exception state and marked by an operator. The relationships are not displayed in an exception view. Exception views are dynamic; although the underlying structure does not change, new elements are included in the view when a resource acquires the appropriate combination of properties (for example, when an operator marks a resource that is in an exception state), and elements are removed when they change any of their relevant properties (for example, when the resource is unmarked again).

A BLDVIEWS definition of a view typically consists of two parts. With the VIEW (for network views) or EVIEW (for exception views) statement you specify the type and properties of the view.

The second part of the definition specifies the resources that are to appear in the view. All SA z/OS resources in RODM are Non-SNA (GMFHS Managed Real) resources. Thus, you can use either the MyName or the DisplayResourceName field as a selection criterion for the Non-SNA resources to be included in the view. Note that you can specify more than one selection criterion.
The following are a typical network view and a typical exception view definition.
First the network view:

```plaintext
VIEW=ING.SS7.APPL,ANNOTATION='Sysplex SS7'
NONSNA='SS7.AOC7.*',QUERYFIELD=DRN
```

The NONSNA statement must be read as "Include in the view all Non-SNA resources whose DisplayResourceName field contains a value beginning with SS7.AOC7." For more information on view definitions for the BLDVIEWS tool see the online help that is shipped with NetView.

A typical exception view:

```plaintext
EVIEW=ING.SS7.EXC,ANNOTATION='Exceptions SS7',DSF=NOXCPT,MF=OFF
NONSNA='SS7.AOC7.APL.*',QUERYFIELD=DRN
```

Note the interplay between the selection criterion of the NONSNA statement and the filter parameters of the EVIEW statement. In contrast to the network view, the NONSNA statement for an exception view only specifies the candidates for appearance in the view. Which of these candidates are in fact included depends on the filter parameters. In the example definition, the exception state filter (DSF parameter) and the mark filter (MF parameter) are set so that only those candidate resources are displayed that are in an exception state and marked by an operator.

In order to be able to specify the selection criteria correctly, you must know how the names of the SA z/OS resources in the RODM MyName and DisplayResourceName fields are constructed. This is explained in the following section.

**Names**

Every SA z/OS resource in RODM has two names: an object name (RODM MyName field) and a display name (RODM DisplayResourceName field). The object name identifies the object uniquely within RODM; the display name appears as the label of the icon that represents the object on the NMC. When defining a view to BLDVIEWS, you can use either of these names as a selection criterion for the objects to be included in the view.

Because the names for system operations and processor operations objects are somewhat different, they are described separately.

**Names for System Operations Objects**

The object names for system operations objects have the following format:

```plaintext
sysplex_name.{name/type/system \ name/type | ANCHOR/MJR/anchor_name}

[minor_res_name]
```

Here `sysplex_name` is the name of a sysplex as defined by a SYSPLEX statement in the INGTOPOF file.

The display names have the following format:

```plaintext
sysplex_name.{system.type.name | type.name | anchor_name.MJR.ANCHOR}

[minor_res_name] [ev_summ]
```

Here again, `sysplex_name` is the name of a sysplex as defined in the INGTOPOF file. Note that the order of the component names for major resources is reversed in the display name. This is done to provide more flexibility when specifying filter criteria for views. Note further that slashes (/) are replaced by dots in the names.
of minor resources. \textit{ev\_summ} stands, in the case of a regular major resource, for the current compound status of the resource. With minor resources it stands for various information, for example for the message content of a WTOR or the timestamp of a heart beat.

The following are some examples for both name types:
1. Application TSO on system AOC7 in sysplex SS7:
   \begin{itemize}
   \item \textbf{Object Name:}
     \begin{verbatim}
     SS7.TSO/APL/AOC7
     \end{verbatim}
   \item \textbf{Display Name:}
     \begin{verbatim}
     SS7.AOC7.APL.TSO
     \end{verbatim}
   \end{itemize}

2. Regular minor resource SPOOL of application JES2JSHP on AOC7 in sysplex SS7 in state OK:
   \begin{itemize}
   \item \textbf{Object Name:}
     \begin{verbatim}
     SS7.JES2JSHP/APL/AOC7.SPOOL
     \end{verbatim}
   \item \textbf{Display Name:}
     \begin{verbatim}
     SS7.AOC7.APL.JES2.SPOOL OK
     \end{verbatim}
   \end{itemize}

3. WTOR outstanding on system AOC7 in sysplex SS7:
   \begin{itemize}
   \item \textbf{Object Name:}
     \begin{verbatim}
     SS7.ANCHOR/MJR/WTOR.AOC7/0007
     \end{verbatim}
   \item \textbf{Display Name:}
     \begin{verbatim}
     SS7.WTOR.MJR.ANCHOR.AOC7.0007 NETCTST7_DSI802A
     \end{verbatim}
   \end{itemize}

4. Coupling facility on system AOC7 in sysplex SS7:
   \begin{itemize}
   \item \textbf{Object Name:}
     \begin{verbatim}
     SS7.ANCHOR/MJR/SYSPLEX.CF/CF01/AOC7
     \end{verbatim}
   \item \textbf{Display Name:}
     \begin{verbatim}
     SS7.SYSPLEX.MJR.ANCHOR.CF/CF01/AOC7 Connected
     \end{verbatim}
   \end{itemize}

\textbf{Note:} The display name can be customized. For further information, refer to "Appendix B, Syntax for INGTOPOF File" in \textit{IBM Tivoli System Automation for z/OS Planning and Installation}.

\textbf{Names for Processor Operations Objects}

The object names for processor operations objects have the following format:

\begin{verbatim}
procops_name.name[/focal_point_system]/1st_level_min[/2nd_level_min
[/{3rd_level_min}]
\end{verbatim}

Here \textit{procops_name} is the name of a focal point configuration as defined by a PROCOPS statement in the INGTOPOF file.

The display names have the following format:

\begin{verbatim}
procops_name.[focal_point_system.]type.name.1st_level_min.2nd_level_min
[.3rd_level_min][ev\_summ]
\end{verbatim}

Here again, \textit{procops_name} is the name of a focal point configuration as defined in the INGTOPOF file. As with the system operations names, the order of the major resource component names is reversed, and slashes (/) are replaced by dots. \textit{ev\_summ} stands for the current processor operations state of the resource.
Creating Views

SA z/OS Topology Manager Commands

There are two commands that you can use to communicate with the SA z/OS topology manager: INGTOPO and INGPOST. For more information on these commands, refer to IBM Tivoli System Automation for z/OS Programmer’s Reference and IBM Tivoli System Automation for z/OS Operator’s Commands.

Processor Operations on the NMC

Processor operations deals with controlling operations of the processor hardware. One or more of these processors are controlled from a focal point. There can be more than one focal point. It is these focal points that the SA z/OS topology manager contacts for gathering information. A focal point with its associated target processors is called a processor operations focal point configuration or, if no confusion is likely to arise, simply a focal point configuration.

Processor operations is independent from system operations. Accordingly, when the same enterprise component (for example, a system) is integrated into processor operations and also into system operations, the SA z/OS topology manager creates two representations of it in RODM. For these reasons it will usually be advisable to keep the processor operations objects apart from the system operations objects in the NMC views. See “Creating Views” on page 216 for defining views.

The following figure shows a network view of a processor operations configuration.

Figure 73. Processor Operations View

The SA z/OS topology manager knows four kinds of processor operations objects that can be distinguished by the type component of their labels. This label is constructed as follows:

1. The name of a focal point configuration as defined in the INGTOPOF file.
   For the INGTOPOF file, see IBM Tivoli System Automation for z/OS Planning and Installation.

2. The resource name.
   This consists of the name itself, the type identifier, and optionally a system name.
There are four resource types, whose names are derived (with one exception) from the corresponding processor operations type names by omitting the prefix ISQ. The names have the following meanings:

- **TGT**: This characterizes a target system.
- **THW**: This characterizes a target processor.
- **PGR**: This characterizes a group of target systems.
- **SBG**: This characterizes a subgroup of target systems.

**Note:** This type identifier is different from the second part of the corresponding processor operations type name, which is ISQGRP. The reason for this deviation is that GRP is already used within system operations.

3. A minor resource name consisting of one, two or three parts. The parts are separated by dots.

The states of processor operations objects are interpreted in terms of NMC states. You can view the complete processor operations status in the **Data 2** field of the **Resource Properties** window.

**Monitoring Sysplex Resources**

The SA z/OS topology manager detects and provides objects that represent sysplex resources (for example, coupling facilities and couple data sets). Also sysplex resource specific commands are provided so that you can issue them in the same way as described in [“Issuing Commands from the NMC” on page 212](#).

**Problem Management**

Problem management facilities offered by the NMC workstation consist of the following aspects described in the subsequent sections:

- **“Exception Views”**
- **“Marked and Unmarked Resources” on page 221**
- **“Updating Operator Data” on page 221**
- **“Problem Management of Minor Resources” on page 221**

These sections are an outline of suggestions how you can use the combination of these facilities to establish an efficient problem management.

**Exception Views**

Exception views are typically used for problem management. It is recommended that you define your exception views in pairs, where the difference between the two views is that one of them contains only unmarked resources, and the other only marked resources. For example, the marked resources could be those being worked upon by an operator, the unmarked ones are not yet handled. The exception views are actualized dynamically; if you mark a resource within a view that contains only unmarked resources, the resource is immediately removed from that view and added to the corresponding view with marked resources.

The following view, for example, shows one of the resources also occurring in [Figure 65 on page 206](#) that is in an exception state.
For more information on network and exception views, see “Network Views and Exception Views” on page 216.

Marked and Unmarked Resources

The feature of marked and unmarked resource as introduced in “Exception Views” on page 220 is provided for every resource via the resources command menu. The NMC workstation just lets you mark resources. Marked resources are displayed with a flag. You define within your enterprise what the meaning of the mark should be. So for example, you can distinguish resources in different states of attendance. Marked and unmarked resources dynamically change within the appropriately defined views according to your marking actions.

Markings processed through the SA z/OS marking commands (offered in the popup menu) can be persistently stored by SA z/OS. This is the advantage over the standard marking function as provided by native NMC.

Updating Operator Data

You can write messages into the Data 2 field of the Resource Properties window from the NMC. This can be used for problem management. For example, when you mark a resource that is in an exception state in order to indicate that you are working on the problem you can add in this way a comment for your colleagues.

You call this function by selecting SA Icon Commands and then Update Operator Data. A window appears where you can enter your comment. This will immediately be written into the Data 2 field of the selected resource.

The advantage of this function, as compared to the standard function for information exchange between operators in the lower part of the Resource Properties window is that the comments can be persistently stored.

Problem Management of Minor Resources

Minor resources inherit the view properties of the related major resources. Thus they appear in the same views as the major resource.
For example, if a major resource were able to appear in an exception view, and a minor resource of this major resource runs into a problem, this minor resource will appear in the exception view (independently of whether the major resource is visible). However, because minor resources are bound to a major resource, they can only appear in views where you would expect the major resource.

You can apply problem management on minor resources just as you would do for major resources; that is, you can mark and unmark minor resources and you can also append a comment to them as described in “Updating Operator Data” on page 221. Additionally, in contrast to major resources, you can delete minor resources.
Part 4. Appendixes
Appendix A. Automation Initialization Options

At the start of SA z/OS initialization a multiline write-to-operator (MLWTO) message, AOF767I, is issued describing the available options. The lines of the MLWTO define the options that can be specified for SA z/OS initialization. Each line is listed and explained below.

AOF767I is followed by either an AOF603D or an AOF606D WTOR. SA z/OS replies automatically to an AOF603D after two minutes unless this has been changed as part of your site’s automation policy. SA z/OS does not automatically reply to an AOF606D message.

The appearance of AOF767I, AOF603D and AOF606D can be suppressed by setting the variable AOFINITREPLY to 0 (see Appendix “Global Variables to Enable Advanced Automation” in IBM Tivoli System Automation for z/OS Customizing and Programming).

A description of the options follows:

- **STOP - CANCEL AUTOMATION**
  This line is always present. It indicates that you can reply **STOP**. Doing so causes SA z/OS to issue an AOF605I and stop running. You must recycle NetView to restart SA z/OS.

- **PAUSE - SUSPEND AUTOMATION**
  This line is present unless SA z/OS initialization has previously been paused. It indicates that you can reply **PAUSE**. Doing so causes SA z/OS to go into a paused state. An AOF604I message is issued, the AOF767I message is reissued, and an AOF606D WTOR is issued. You should not automate the reply to the AOF606D WTOR, as this disables the pause function.

  If the automation manager is started, restarted, or refreshed (with the INGAMS or MVS modify command) while the automation agent is paused, the reply is cancelled and the automation agent initialization is redriven from the beginning.

- **NOSTART - DO NOT AUTOMATE SUBSYSTEM STARTUP**
  This line is always present. It indicates that you can reply **NOSTART**. Doing so causes SA z/OS to suspend the automation agent (see INGAMS) to avoid that any start or stop orders are sent by the automation manager.

- **RUNMODE=x - SET RUNMODE(CURRENTLY mode)**
  This line is present when runmodes are used in the policy or when a runmode request is present in the automation manager. It indicates that can reply RUNMODE= followed by a valid runmode. Doing so causes SA z/OS to set the specified runmode before the automation agent is resumed after loading the configuration. You can specify RUNMODE=? to get a list of all valid runmodes for the current system.

- **ENTER - CONTINUE**
  The initialization of SA z/OS continues.

A sample of how the AOF767I message could look, is in [Figure 75 on page 226](#).
Automation Initialization Options

AOF767I AUTOMATION OPTIONS:
.STOP   - CANCEL AUTOMATION
.PAUSE   - SUSPEND AUTOMATION
.NOSTART - DO NOT AUTOMATE SUBSYSTEM STARTUP
.RUNMODE=x - SET RUNMODE (CURRENTLY *ALL)
.ENTER   - CONTINUE

Figure 75. Sample AOF767I Message

If the message on your screen looks significantly different from the sample shown, ask your automation programmer whether they used the AOFEXI01 initialization exit to modify the message.
Appendix B. Using the I/O Operations ISPF Dialog

Notes:
1. Refer to IBM Tivoli System Automation for z/OS Planning and Installation for instructions on how to install and to modify the ISPF dialog.
2. The IBM-supplied REXX code, including the EXECs, panels, messages, and tables, is supported, provided it has not been modified. Therefore, if you intend to modify IBM-supplied code, it is advisable to retain two copies: the IBM-supplied copy and the modified.
3. For more information on ISPF, refer to the ISPF publications that can be found in the z/OS Internet Library at http://www.ibm.com/systems/z/os/zos/bkserv/.

Getting Started

To use the I/O operations ISPF dialog:
- You must already be logged on to your user ID and using ISPF.
- The I/O operations ISPF dialog must be available.

To see whether the I/O operations ISPF dialog is available, display the ISPF/PDF Primary Option panel on the ISPF main menu.
- If I/O operations is displayed as an option on the ISPF main menu, select it.
  The I/O operations ISPF main menu, which is shown in Figure 76 on page 228 should now be displayed. If it is not, consult your system administrator or IBM Tivoli System Automation for z/OS Planning and Installation for further instructions.
- If I/O operations is not displayed on the ISPF/PDF Primary Option panel:
  1. Select option 6 (Command) on the ISPF/PDF Primary Option panel and press Enter.
  2. If the library that is required by the customization dialog (ING.CUSTOM.IHVCONF) has been allocated using the TSO logon procedure or using a CLIST of your own, enter INGDLG SELECT(IOCONNECT) ALLOCATE(NO).
     (To issue the INGDLG command, the ING.SINGIREX data set must be allocated in the SYSPROC DD statement of your logon procedure.)
     The I/O operations ISPF main menu, which is shown in Figure 76 on page 228 should now be displayed. If it is not, or if your TSO logon procedure is not set up to allocate the data set that is required by the ISPF dialog, consult your system administrator or IBM Tivoli System Automation for z/OS Planning and Installation for further instructions.

Starting with the I/O Operations ISPF Main Menu

This section briefly describes the I/O operations ISPF main menu, which is shown in Figure 76 on page 228.
Starting with the I/O Operations ISPF Main Menu

1. **Menu Bar**
   - that you select menus A through E from.

2. **Modify Menu**
   - Lets you modify a switch configuration in matrix format and issue a change to all switches that are allocated to the issuing SA z/OS I/O operations. Place the cursor in Modify and press Enter. See “The Modify Menu” on page 229 for more information.

3. **View Menu**
   - Prompts you for the operands of the Display commands and then displays the data you request. Place the cursor in View and press Enter. See “The View Menu” on page 231 for more information.

4. **Options Menu**
   - Lets you customize the IBM-supplied defaults for your user ID. Place the cursor in Options and press Enter. See “The Options Menu” on page 234 for more information.

5. **Help Menu**
   - Shows you how to display the online help and the online tutorial. Place the cursor in Help and press Enter. See “The Help Menu” on page 235 for more information.

6. **Locking Menu**
   - This is a simple way to enter the Getlock and Unlock commands. Enter 1 or 2. If access is controlled by RACF, the UNLOCK command requires only Read authorization, but the GETLOCK command requires the Control level. See IBM Tivoli System Automation for z/OS Operator’s Commands for
Starting with the I/O Operations ISPF Main Menu

Information about authorization required for I/O operations commands and for information about the GETLOCK and UNLOCK commands.

2 ISPF Command Line

On the ISPF command line, you can enter any applicable ISPF command. For more information on ISPF, refer to the ISPF publications that can be found in the z/OS Internet Library at http://www.ibm.com/systems/z/os/zos/bkserv/.

3 I/O Operations Command Line

On the I/O operations command line, you can enter any SA z/OS I/O operations command. See IBM Tivoli System Automation for z/OS Operator’s Commands for a complete list of these commands.

The Modify Menu

Modify

<table>
<thead>
<tr>
<th>1. Work with active...</th>
<th>Action (Press Enter when Needed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Work with active (matrix)</td>
<td>a. Select a switch from the list in the Select SWCH panel. b. Go to “A Sample Way to Modify a Switch Matrix” on page 230 for more information.</td>
</tr>
</tbody>
</table>

| 2. Work with saved (matrix) | |
|-----------------------------| a. Select a saved configuration from either: |
|                             | – Host (saved in ISPF) |
|                             | – Switch |
|                             | b. Follow directions. |

<table>
<thead>
<tr>
<th>3. Work with a default Matrix</th>
<th>Go to “A Sample Way to Modify a Switch Matrix” on page 230 for more information.</th>
</tr>
</thead>
</table>

| 4. Single change to all SWCHs | |
|-------------------------------| a. Select a command on the Single Change to All SWCHs panel. |
|                               | b. Enter the required information in the fields on the panel (for example, enter a port address or name for the Block command). |
|                               | d. If the Verify panel is displayed, verify the information and press Enter. |
|                               | e. When processing is complete, do one of the following: |
|                               | – Repeat this sequence of steps to make another change. |
|                               | – Press F3 to exit. |

After you have created or changed a switch configuration in matrix format, you can activate it or make it operational on the designated switch, or you may save it on the host or on a selected ESCON/FICON Director. The issuing SA z/OS I/O operations must be running and the switch must be allocated to it.
The Modify Menu

You cannot activate the IBM-supplied default matrix until you have changed its default switch device number of 0000 to a number that is valid for your configuration. (See the last entry in the table under “Sample Changes You Can Make to a Port” on page 231.)

It is very advisable to save an active switch matrix, even if this is not mandatory.

A Sample Way to Modify a Switch Matrix

When a switch matrix is displayed, you can make the following changes on it; you do not have to enter commands at the command line.

1. If you want to assign a name to the switch, enter the name in the SWCH Name field.

   Note: Press Enter if you want to see the change highlighted.

2. To make a change to a port, check whether that port is horizontal row is displayed on the matrix. If it is not:
   - To scroll down the left column of ports, use F8 to scroll to the higher port addresses and F7 to scroll in the other direction (backwards).
   - To scroll across the column of ports in the matrix, use F11 to the right across the screen and press PF10 to scroll to the left.

   Some sample changes are shown under “Sample Changes You Can Make to a Port” on page 231.

3. After you have finished making changes, consider whether you want to save or to activate the switch configuration, or both.

   Note: It is strongly recommended that you save a switch configuration before activating it.

4. To save a switch configuration:
   a. Press F5.
   b. On the Select Destination of Save Configuration panel, first choose between saving at a switch or at the host. Then, either select a name from the list that is presented, or enter a new name and description.
   c. Press Enter to save the matrix of the switch configuration. If you are replacing another switch configuration, another panel will prompt you to verify the change. In that case, press Enter again.
   d. If you want to activate the switch configuration, refer to the next step.

5. To activate the switch configuration:
   a. Press F6 on the screen of the matrix.
   b. Verify the information.
   c. Press Enter.

   Note: If the saving or the activation of a switch configuration fails you can press the function key F2 to get the reason (or reasons) why your action failed. As an alternative to the function key you can issue the command RESULTS from the command line.
## Sample Changes You Can Make to a Port

<table>
<thead>
<tr>
<th>I/O operations Command</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block a port</td>
<td>• Type B at the intersection point between the port row and the B column. For example, if you type B in the B column at D5’s row, D5 will be blocked.</td>
</tr>
<tr>
<td>Unblock a blocked port</td>
<td>• Type U or a space at the intersection point between the port row and the B column. For example, type U over the B in the B column at D5’s row to unblock port D5.</td>
</tr>
<tr>
<td>Connect two ports statically (dedicated)</td>
<td>• Type the port address of the 2nd port at the intersection point between the 1st port and the C column. For example, type C3 in the C column at port row E4 to connect C3 and E4 statically.</td>
</tr>
<tr>
<td>Disconnect a static (dedicated) connection</td>
<td>• Type spaces over the port address at the intersection point between the port row and the C column. For example, type spaces over C3 in the C column at port row E4 to disconnect C3 and E4 from each other.</td>
</tr>
<tr>
<td>Prohibit a dynamic connection</td>
<td>• Using PF10 and F11 to scroll right and left, display the port column of the 2nd affected port. • Type P at the intersection point between the 1st port’s row and the 2nd port’s column. For example, type P at the intersection of port row D7 and port column E1 to prohibit these ports from connecting dynamically to each other.</td>
</tr>
<tr>
<td>Allow a dynamic connection</td>
<td>• Using PF10 and F11 to scroll right and left, display the port column of the 2nd affected port. • Type A at the intersection point between the 1st port’s row and the 2nd port’s column. For example, type A at the intersection of port row D7 and port column E1 to allow these ports to connect dynamically to each other.</td>
</tr>
<tr>
<td>Copy Allows and Prohibits from one port row to another port row</td>
<td>• Type C on the blank line to the left of the port row you plan to use as a source. • Type T on the blank line to the left of the port row you plan to use as a target.</td>
</tr>
<tr>
<td>Name or rename a port</td>
<td>• Type a port name at the intersection point between the port row and the Name column. For example, type PRINTER1 in the Name column of port row C5 to name C5 PRINTER1. It is recommended that the name be in uppercase letters. (Refer to IBM Tivoli System Automation for z/OS Planning and Installation for suggestions for naming ESCON/FICON Director ports.) You can remove a name by typing blanks over it.</td>
</tr>
<tr>
<td>Change the switch device number</td>
<td>• Type a switch device number in the Target SWCH Number field. (This is a required action if you start with the IBM-supplied default matrix of 0000.)</td>
</tr>
</tbody>
</table>

### The View Menu
If you select the View menu, the subtasks shown on the left are displayed. When the selected view is displayed:

- Use F7 to scroll backward and F8 to scroll forward to view the information.
- Press F3 to return to the display selection screen.

<table>
<thead>
<tr>
<th>Selection</th>
<th>Option</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Hosts</td>
<td>The SA z/OS I/O operations base programs that are known to the issuing SA z/OS I/O operations.</td>
<td>Type * on the 1st blank line. See the DISPLAY HOST command in IBM Tivoli System Automation for z/OS Operator’s Commands.</td>
</tr>
<tr>
<td></td>
<td>Detailed information about the SA z/OS I/O operations base program specified.</td>
<td>Type the VTAM name of an SA z/OS I/O operations on the 1st blank line. See the DISPLAY HOST command in IBM Tivoli System Automation for z/OS Operator’s Commands.</td>
</tr>
<tr>
<td></td>
<td>A list of all SA z/OS I/O operations base programs that have the specified switch allocated to them.</td>
<td>Type a switch device number on the 2nd blank line. See the DISPLAY HOST command in IBM Tivoli System Automation for z/OS Operator’s Commands.</td>
</tr>
<tr>
<td>2. CHPIDs</td>
<td>A list of all CHPIDs with their associated information</td>
<td>Type * on the blank line. See the DISPLAY CHP command in IBM Tivoli System Automation for z/OS Operator’s Commands.</td>
</tr>
<tr>
<td></td>
<td>Detailed information about a specified CHPID</td>
<td>Type a CHPID number on the blank line. See the DISPLAY CHP command in IBM Tivoli System Automation for z/OS Operator’s Commands.</td>
</tr>
</tbody>
</table>
### The View Menu

<table>
<thead>
<tr>
<th>Selection</th>
<th>Option</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. Ports</td>
<td>A list of all ports with the same name</td>
<td>Type the name of a named port on the 1st blank line. A generic port name may be used. See the DISPLAY NAME command in IBM Tivoli System Automation for z/OS Operator’s Commands.</td>
</tr>
<tr>
<td></td>
<td>Details about a port name or port address on a specified switch</td>
<td>Type either a port address on the 2nd blank line or a port name on the 3rd blank line. Then, type a switch device number on the 4th blank line. See the DISPLAY PORT command in IBM Tivoli System Automation for z/OS Operator’s Commands.</td>
</tr>
<tr>
<td></td>
<td>Information about all ports, only blocked ports, or only statically connected ports on a specified switch</td>
<td>Type A, B, or C on the 5th blank line. Then, type a switch device number on the 6th blank line. See the DISPLAY PORT command in IBM Tivoli System Automation for z/OS Operator’s Commands.</td>
</tr>
<tr>
<td>4. Devices</td>
<td>A list of all devices attached to a specified switch</td>
<td>Type * on the 1st blank line. Then, type a switch device number on the 2nd blank line. See the DISPLAY DEVICE command in IBM Tivoli System Automation for z/OS Operator’s Commands.</td>
</tr>
<tr>
<td></td>
<td>Detailed information about a specific device with relation to all switches</td>
<td>Type a device number on the 1st blank line. Then, type * on the 2nd blank line. See the DISPLAY DEVICE command in IBM Tivoli System Automation for z/OS Operator’s Commands.</td>
</tr>
<tr>
<td></td>
<td>Detailed information about a specified device relative to a specified switch</td>
<td>Type a device number on the 1st blank line. Then, type a switch device number on the 2nd blank line. See the DISPLAY DEVICE command in IBM Tivoli System Automation for z/OS Operator’s Commands.</td>
</tr>
<tr>
<td></td>
<td>Multiple devices on a specified switch</td>
<td>Type a range of device numbers on the 1st blank line. Then, type a switch device number on the 2nd blank line. See the DISPLAY DEVICE command in IBM Tivoli System Automation for z/OS Operator’s Commands.</td>
</tr>
</tbody>
</table>
The Options Menu

In the I/O operations ISPF dialog, a number of options are shipped with IBM-supplied defaults. You can customize these defaults to your ISPF user ID. Except for the Verify option, your customized options remain in effect for your user ID across SA z/OS I/O operations sessions.

1. Do either of the following:
   - On the dialog’s main menu, select the Options, and select one of the entries.
   - Enter OPTIONS in the command area of a dialog panel.

2. Make a selection (1–6).

3. If you want to save your change, press F3 or enter END.

1. Assist
   Describes the return codes and reason codes in the messages. Selections are:
   1. To receive explanatory text and specific messages (IBM-supplied default)
   2. Not to receive explanatory text on the return codes and reason codes.

2. Command
   Specifies a set of default connectivity options. Selections are:
   1. NOForce, Vary, BAckout (IBM-supplied defaults)
   2. NOForce, Vary, NOBackout
   3. NOCheck, Vary, BAckout

---

The View Menu

<table>
<thead>
<tr>
<th>Selection</th>
<th>Option</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>5. Command Results</td>
<td>A list of command results with relation to each SA z/OS I/O operations</td>
<td>See the DISPLAY RESULTS command in IBM Tivoli System Automation for z/OS Operator’s Commands.</td>
</tr>
<tr>
<td>6. Varys</td>
<td>A list of failed vary path requests with relation to a specific SA z/OS I/O operations host</td>
<td>Type the SA z/OS I/O operations VTAM name on the blank line. See the DISPLAY VARY command in IBM Tivoli System Automation for z/OS Operator’s Commands.</td>
</tr>
<tr>
<td>7. SWCH</td>
<td>A list of all switches allocated, or attached, to the issuing instance of I/O operations.</td>
<td>Type * on the blank line. See the DISPLAY SWITCH command in IBM Tivoli System Automation for z/OS Operator’s Commands.</td>
</tr>
<tr>
<td>Details about a specific switch</td>
<td></td>
<td>Type a switch device number on the blank line. See the DISPLAY SWITCH command in IBM Tivoli System Automation for z/OS Operator’s Commands.</td>
</tr>
</tbody>
</table>
The Options Menu

4. NOCheck, Vary, NOBackout
5. Force, Vary, NOBackout
6. Force NOVary, NOBackout

3. Getlock
Obtains processing control. (See the GETLOCK command in IBM Tivoli System Automation for z/OS Operator’s Commands for more information.)
Selections are:
1. Never take control from another user.
2. Be prompted with the Getlock panel so you can decide whether you want to take control (IBM-supplied default).
3. Always take control. If a command is being processed by another user, however, processing can be interrupted and the command can fail.

4. Matrix
Specifies the changes on the switch configuration that will be activated by SA z/OS I/O operations. (Select them with any character.) Selections are:
___ ALL commands (overrides other options; IBM-supplied default)
___ Allow and Prohibit
___ Connect and Disconnect
___ Block and Unblock
___ Port names
___ Switch name

5. Unlock
Relinquishes control of SA z/OS I/O operations command processing. (See the UNLOCK command in IBM Tivoli System Automation for z/OS Operator’s Commands for further information.) Selections are:
1. No automatic unlock. You must enter the Unlock command.
2. Automatic unlock when you press F3 or END from the I/O operations ISPF main menu (IBM-supplied default).
3. Automatic unlock after every command.

6. Verify
Requires you to verify a command before the dialog activates it. Contrary to the other options, you can tailor the Verify option for the remainder of the session only. The default for this option resets to the IBM-supplied default at the beginning of each session. Selections are:
1. Verify before activating (IBM-supplied default).
   • If you press Enter after entering a connectivity command, the dialog will continue.
   • If you press F3 or END, command processing is halted, and the preceding screen will be displayed.
2. Do not verify.

The Help Menu
1. How to get help:
   - To get the entries shown above, select the Help menu from the I/O operations ISPF main menu.
   - To get field-oriented help, press F1 on any input field on a screen.
   - To get screen-oriented help, press F1 on any full screen. If you press F1 twice in a row, an ISPF help panel, not an I/O operations ISPF help panel, is displayed.

2. Help on keys... explains the I/O operations ISPF dialog settings of the following function keys: F1 (Help); F3 (End); F5 (Save); F6 (Activate).

3. Navigation Help... helps you navigate through the panel, or screen, flow of the I/O operations ISPF dialog.

4. Help Index... lets you choose among a selected number of topics that are listed alphabetically.

5. Tutorial ... Do one of the following:
   - Select the Help menu on the main menu. Then:
     a. Select Tutorial
     b. Select one of the topics shown in on the screen below.
   - Enter TUTORIAL on a panel of the dialog. Then:
     a. Press Enter
     b. Select one of the topics on the tutorial main menu shown below.

**SA z/OS - IO-Ops ISPF Tutorial Main Menu**
Select a specific topic by typing the topic number on the command line and pressing Enter.

1. Overview of IO-Ops and SWCHs
2. Displaying and Changing SWCH Configuration Data
3. Displaying Other Types of Information
4. Setting Options
5. Guidelines and Precautions
6. Understanding and Using the Matrix Screen
7. Making a Single Change to All SWCHs
8. Connectivity Command Options

**COMMAND ====>**
F1= Help   F2=   F3= END   F4= RETURN   F5=   F6=
F7= Help   F8=   F9=   PF10=   F11= Index   F12=
Appendix C. Problem Determination

This appendix contains various subtopics that deal with methods of avoiding or detecting problems during SA z/OS automation. The following topics are discussed:

- "Maintaining System Status during a Status Forwarding Path Failure"
- "Debugging and Tracing"

Maintaining System Status during a Status Forwarding Path Failure

SA z/OS uses processor operations functions and XCF to attempt to maintain the current status of a system, even if there is a failure in the status forwarding path. SA z/OS system operations uses XCF for both passive and active monitoring of system status. As long as there is one SA z/OS in a sysplex with an active status forwarding path, SA z/OS is able to use XCF to maintain system status for all systems in the sysplex.

SA z/OS system operations receives alerts generated by processor operations for target systems monitored by processor operations. If, for example, a target system enters a wait state or a DCCF condition, SA z/OS will use the resulting alert from the processor operations functions to update the system status, even if the status forwarding path between SA z/OS and the target system is down.

Debugging and Tracing

This information describes the tracing and debugging facilities that are available with SA z/OS. The following topics are discussed:

- "Automation Manager State Trace Table" on page 238
- "Using Trace Services for the Automation Manager" on page 239
- "Using AOCTRACE" on page 240
- "How to Use a Log with INGHIST" on page 240
- "How to Use the Diagnostic Option of INGAMS" on page 240
- "Command Logging" on page 82
- "Tracing and Debugging for ProcOps and the BCP Internal Interface" on page 247
- "Using Trace Services for I/O Operations" on page 249

To collect debugging information you can also use the command INGLKUP REQ=COLLECT.

Automation Manager State Trace Table

The event handler trace back table is written to SYSLOG or to the Message Logger. It is the most important debug tool when you want to understand the event flow.

The table is written using the HSAM1399I message due to one of the following:

- Address space termination (written automatically to the Message Logger)
- /F jobname,$TRACESTATE is written to SYSLOG
Using Trace Services for the Automation Manager

To trace the control flow of a process, SA z/OS uses the MVS Component Trace Facility with all its capabilities.

This requires an External Writer as soon as the trace is started. The writer is internally started if it is not already active.

Before the trace can be activated the trace data set must be allocated. The recommended data set name is 'SYS1.HSA.sysname.CTRACE01'. An appropriate JCL is provided in the SINGSAMP SA z/OS sample library as the member HSACTWR. In addition, the JCL of the component trace's external writer must be cataloged in SYS1.PROCLIB. For more information on customizing the component trace refer to IBM Tivoli System Automation for z/OS Planning and Installation.

To collect the trace data in a data set, the external writer must be started. Enter the following command on a z/OS console:

```
TRACE CT,WTRSTART=HSACTWR
```

To activate the trace, enter the following command on a z/OS console:

```
TRACE CT,ON,COMP=HSAAM
```

When the system responds with a WTOR for the trace options, reply as follows:

```
xx,WTR=HSACTWR,OPTIONS=(ALL|set_of_options),END
```

Where `set_of_options` is one of the options in Table 14.

Table 14. SA z/OS Trace Activation Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Activates the Trace In:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>All of the following options</td>
</tr>
<tr>
<td>Base</td>
<td>SA z/OS service routines using z/OS system services</td>
</tr>
<tr>
<td>COMmunication</td>
<td>The communications framework</td>
</tr>
<tr>
<td>CONtrol</td>
<td>The base framework</td>
</tr>
<tr>
<td>Event</td>
<td>The Sysplex Event Handler</td>
</tr>
<tr>
<td>Framework</td>
<td>The service sustaining the LE environment</td>
</tr>
<tr>
<td>Hardware</td>
<td>Processor operations processing</td>
</tr>
<tr>
<td>Instruction</td>
<td>The services encoding or decoding the instruction streams</td>
</tr>
<tr>
<td>Lock</td>
<td>The service creating and deleting locks</td>
</tr>
<tr>
<td>Request</td>
<td>The beginning and ending of a work item (request) processing</td>
</tr>
<tr>
<td>Rexx (functions)</td>
<td>REXX function routines</td>
</tr>
<tr>
<td>Schedule</td>
<td>The service overriding service period definitions</td>
</tr>
<tr>
<td>SDF</td>
<td>SDF control blocks, such as the 3270 data stream buffer and control flow</td>
</tr>
<tr>
<td>Workitem</td>
<td>All components working on work items</td>
</tr>
</tbody>
</table>

Deactivate the trace with the command:

```
TRACE CT,OFF,COMP=HSAAM
```

This command is automatically issued when the last automation manager in a single image terminates while the trace is active.
Before the trace records can be analyzed by IPCS the trace data set or sets must be made available to IPCS. This is done with the operator command:

```
TRACE CT,WTRSTOP=HSACTWR
```

If the related message indicates that more than one data set contains trace data, you must use the IPCS command COPYTRC to combine the data sets into a new one.

To analyze the trace data, you must use the IPCS command CTRACE. A JCL is provided for the IPCS batch interface. The command supports several parameters and, at the very least, you must adapt the following:

- COMP(HSAAM)
- USEREXIT(HSAPSTUE)
- DSNAMETrace_data_set_name
- OPTIONS((filter_options)
  where filter_options is a combination of the options listed in Table 14 on page 238 and the following items, separated with a comma:
  - AM=xcf_member_name
  - CALLID=(callid[,callid])
  - KEY=(key[,key])
  - METHOD=(method[,method])
  - MODULE=(module[,module])
  - TASKID=(taskid[,taskid])
  - TOKEN=(token[,token])

The keyword options are connected by a logical AND. The values of a keyword and the remaining options are connected by a logical OR. Apart from AM, each keyword option supports up to 5 different values separated by a comma or a space. Imbedded spaces or commas in a value require delimiters, such as '...' or "...".

**JCL Sample for Printing the Trace Records**

The JCL sample member HSAJPTRC is stored in the SA z/OS SINGSAMP library. The data set names of the STEPLIB, IPCSDDIR, and TRACEDS DDnames must be changed to their actual values before the job is submitted. This is also true for all of the OPTIONS parameters that are listed because the values of the keywords are only sample values.

**Using AOCTRACE**

You can use the AOCTRACE command to enable or disable the automation debugging facility, either globally or for specific clists (REXX routines).

You can enable tracing for a particular message ID by issuing AOCTRACE at the command line with the MSG/id option, for example:

```
AOCTRACE MSG/AOF313I ON
```

You can find the syntax of the AOCTRACE command and additional information in *IBM Tivoli System Automation for z/OS Operator’s Commands*.

Information about using AOCTRACE to trace user-written automation procedures is provided in *IBM Tivoli System Automation for z/OS Customizing and Programming*.

For more information about trace settings, see TRACE in *TSO/E REXX/MVS Reference*. 
How to Use a Log with INGHIST

With the INGHIST command you can display a log with automation manager messages by entering INGHIST REQ=LOG at the command line.

The INGHIST panel displays automation manager messages that are written to the system logger. Two or more lines are displayed for each message depending on the size of the message text. The first line contains the date and time when the message was issued. It also shows the XCF member name of the automation manager and the XCF group name the automation manager belongs to. The second and following line, or lines, contain the message.

How to Use the Diagnostic Option of INGAMS

From the INGAMS command dialog, you can select option D Diagnostic.

This leads to the panel shown in Figure 77 where you can initiate the following diagnostic functions:

- Writing a snapshot of the state image queue
- Starting or stopping recording
- Displaying work item statistics
- Starting or stopping work-item queue monitoring

Figure 77. INGAMS Diagnostic Functions Panel

In this panel's fields, enter the following information:

Action

Specify the action to be done. For options 1–3, you need to fill in the Data set name and System fields.

Enter 4 to display work item statistics. For details about enabling and using work item statistics, see “Work Item Statistics” on page 241.
Debugging and Tracing

Enter 5 and a value for the monitoring period (10–999 seconds) in the **Interval** field to activate work item queue monitoring. Enter 6 to stop queue monitoring. For more details, see “Workitem Queue Monitoring” on page 246.

**Data set name**

Specify the name of the data set that will hold the snapshot data.

The snapshot data set and the recording data set should be preallocated with RECFM FB and LRECL 80. The snapshot and recording data sets must be sequential files.

It is strongly recommended that you make your allocation of the snapshot data set according to the snapshot size information that you receive when you select option **B Show Details** for an automation manager from the INGAMS command dialog. This displays the information shown in Figure 78 showing the snapshot size (highlighted in the example).

<table>
<thead>
<tr>
<th>Diagnostic Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>Snapshot size : 3492224</td>
</tr>
<tr>
<td>Number of resources : 225</td>
</tr>
<tr>
<td>Number of requests : 66</td>
</tr>
<tr>
<td>Number group requests : 63</td>
</tr>
<tr>
<td>History records : 65422</td>
</tr>
<tr>
<td>Max History records : 262144</td>
</tr>
</tbody>
</table>

*Figure 78. INGAMS Command Dialog, Show Details Panel*

**Note:**

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

**System**

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

**Work Item Statistics**

To display work item statistics, you enter 4 in the **Action** field of the INGAMS Diagnostic Functions Panel (Figure 77 on page 240) without specifying a data set name or system. This displays a panel similar to Figure 79 on page 242.
This panel shows history information about the work items processed by the automation manager. The automation manager keeps track of the last 300 work items processed by each of the tasks that build the automation manager kernel.

**Note:** The history information is not persistent or available when the automation manager runs on a system that is running an earlier version of SA z/OS. The following data is shown:

**Snapshot**

The snapshot timestamp shows the time of the query.

**Task**

The name of the task.

**Status**

The status of the task. It is one of the following:
- **BUSY** The task is processing a work item.
- **IDLE** The task is waiting for work.
- **NOTIDLE** The task has completed work but is not waiting for work.
- **UNKNOWN** History records have not yet been defined for this task.

**Since**

This is the number of seconds that the task has been processing the work item (elapsed time). If this number is unexpectedly high, it is an indication that something is wrong.

The columns to the right of the **Since** column show the number of work items that have been processed in one minute, starting from the time the history query (INGAMS DIAG REQ=STATS) was made or refreshed.

You can use the S command code to display details of the work item history for the selected task, such as the starting time and total processing time of the work item, as shown in Figure 80 on page 243.

**Figure 79. INGAMS Work Item Statistics Display**

This panel shows history information about the work items processed by the automation manager. The automation manager keeps track of the last 300 work items processed by each of the tasks that build the automation manager kernel.

**Note:** The history information is not persistent or available when the automation manager runs on a system that is running an earlier version of SA z/OS. The following data is shown:

**Snapshot**

The snapshot timestamp shows the time of the query.

**Task**

The name of the task.

**Status**

The status of the task. It is one of the following:
- **BUSY** The task is processing a work item.
- **IDLE** The task is waiting for work.
- **NOTIDLE** The task has completed work but is not waiting for work.
- **UNKNOWN** History records have not yet been defined for this task.

**Since**

This is the number of seconds that the task has been processing the work item (elapsed time). If this number is unexpectedly high, it is an indication that something is wrong.

The columns to the right of the **Since** column show the number of work items that have been processed in one minute, starting from the time the history query (INGAMS DIAG REQ=STATS) was made or refreshed.

You can use the S command code to display details of the work item history for the selected task, such as the starting time and total processing time of the work item, as shown in Figure 80 on page 243.
Work Item Lifecycle Recording

Work item lifecycle recording is an internal diagnostic tool that you should use only if required by SA z/OS service.

Lifecycle recording provides enhanced debugging to track down lost requests during automation agent-automation manager communication and other automation manager-related problems.

Lifecycle recording records checkpoints along the flow of a work item. The following flows are reported:

- Work items that flow from an automation agent to the primary automation manager (for example, status updates)
- Work items that flow from an automation agent to the primary automation manager plus the response (for example, INGLIST)
- Orders that flow from the PAM to an automation agent

Using Lifecycle Recording: By default, lifecycle recording is disabled. When enabled, the automation agent and the primary automation manager (PAM) write life cycle records to a data space. Each automation agent and the PAM have their own data space. Lifecycle recording must be enabled for the PAM and at least one automation agent. It can be enabled for the PAM from any automation agent, however it must be enabled for the automation agent from that automation agent.

Before lifecycle recording can be enabled, the size of the data space must be defined. This can be done automatically with the INGRLCR command that enables lifecycle recording.

When the problem has been captured, the data spaces must be off-loaded to external data sets of the same size with the INGRLCR command. A sequential data set is therefore required for the PAM and for each automation agent that lifecycle recording was enabled for.
The allocation of the data sets might be done automatically if the appropriate RACF definitions are given to the NetView that issues the lifecycle recording command.

**Managing Lifecycle Recording with the INGRLCR Command:**

You can enable or disable lifecycle recording by issuing the following INGRLCR commands in NetView:

- **INGRLCR AA ON SIZE=size DSN=dsname**
  Switches on lifecycle recording for the local automation agent.

- **INGRLCR AA OFF**
  Switches off lifecycle recording for the local automation agent.

- **INGRLCR AM ON SIZE=size DSN=dsname**
  Switches on lifecycle recording for the PAM.

- **INGRLCR AM OFF**
  Switches off lifecycle recording for the PAM.

**Note:** If you issue the command with the size parameter, a new data space is created and the old data space is deleted. The dsname parameter specifies the data set that should be used later when the data space is saved. The total number of characters for the name of the data set is limited to 55 characters.

You can offload the data collected for lifecycle recording for the local automation agent or the primary automation manager, as follows:

- **Local automation agent:**
  - Use **INGRLCR AA SAVE DSN=dsname** if you used **INGRLCR AA ON SIZE=size** and did not specify a data set name or you want to save the data to a new data set.
  - Use **INGRLCR AA SAVE** if you used **INGRLCR AA ON SIZE=size DSN=dsname** to save the data to the data set that you specified.

- **Primary automation manager:**
  - Use **INGRLCR AM SAVE DSN=dsname** if you used **INGRLCR AM ON SIZE=size** and did not specify a data set name or you want to save the data to a new data set.
  - Use **INGRLCR AM SAVE** if you used **INGRLCR AM ON SIZE=size DSN=dsname** to save the data to the data set that you specified.

**Notes:**

1. When you offload the data space to a data set make sure that LCR is still ON before you issue SAVE.

2. Only the actual data from data space is copied. Small primary and multiple secondary allocations of the data set reduce the amount of DASD space.

3. If **dsname** does not exist it will be automatically created with the appropriate size. If the data set already exists it is used as it is. You should therefore make sure that it is large enough.

4. If **dsname** ends with *, for example, **MY.HLQ.***, a new data set with a unique name is created using following pattern:

   ```
   MY.HLQ,<AA|AM>,<domain>,<Yymmdd>,<Thhmmss>
   ```

5. NetView must have RACF ALTER access to the data set. The PAM must have RACF UPDATE access to the data set. Allocation of the new data set is done automatically via the NetView ALLOCATE command. The corresponding NetView return codes are passed back to caller in case of an error.
6. The size of the data space can range from 1–2098 MB. It specifies the maximum size of the data space. Initially a smaller data space is allocated that is extended until the maximum is reached. A value of 500 is recommended and should be sufficient for all situations.

Examples:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INGRCR AA ON SIZE=500 DSN=MY_AGENT_DATA_SET</td>
<td>Create data space for automation agent</td>
</tr>
<tr>
<td>INGRCR AA ON SIZE=500 DSN=MY_AGENT.*</td>
<td>Extend data space</td>
</tr>
<tr>
<td>INGRCR AA SAVE</td>
<td>Save data space</td>
</tr>
<tr>
<td>INGRCR AA SAVE DSN=MY.NEW_AGENT.DATA_SET</td>
<td>Save data space with new name</td>
</tr>
<tr>
<td>INGRCR AA SAVE DSN=MY.HLQ.*</td>
<td>Save data space with new name</td>
</tr>
<tr>
<td>INGRCR AM SAVE DSN=MY.NEW.PAM.DATA_SET WAIT=30</td>
<td>Save data space on primary automation manager</td>
</tr>
</tbody>
</table>

Changes to HSAPRMxx and INGXINIT:
The LIFECYCLE parameter in the INGXINIT and HSAPRMxx members defines the size of the data space and the data set name for offloading the lifecycle recording data:

```plaintext
LIFECYCLE=nnnn;dsname
```

This parameter is not mandatory for using lifecycle recording. The same information can be provided later with the INGRCR command when lifecycle recording is actually enabled.

Notes:

1. The LIFECYCLE parameter does not switch on lifecycle recording. This must be done with the INGRCR AA|AM ON command.
2. The LIFECYCLE parameter does not allocate data sets automatically. This is done with the INGRCR command only.
3. You must use a semicolon to separate `nnnn` and `dsname`. No blanks are allowed.

Performing Lifecycle Recording:

Perform the following steps to perform lifecycle recording:

1. **Allocate data sets**: If necessary, allocate sequential data sets for the primary automation manager and each automation agent to be monitored with the following attributes:
   - Organization PS
   - Record format FB
   - Record length 128
   - Block size 27904

   The data sets must be large enough to hold the data space. The recommendation is 500 MB. The automation agents and PAM must have write access to their data sets.

2. **Create the data space**: After the automation agent has completed initialization, enable lifecycle recording on each automation agent that is involved in debugging the problem with the following command:

   ```plaintext
   INGRCR AA ON SIZE=500 DSN=my.agent.dataset
   ```

   Create the data space on the PAM just once with the following command:

   ```plaintext
   INGRCR AM ON SIZE=500 DSN=my.pam.dataset
   ```

   If the SIZE parameter is present, the old data space is deleted and a new empty data space created. The size and the data set name can also be provided with LIFECYCLE parameter in the HSAPRMxx and INGXINIT members.
3. Check, enable or disable lifecycle

You can check the status of lifecycle recording with the following commands:

- INGRLCR AA QRY
- INGRLCR AM QRY

If the data space has already been created (through either HSAPRMxx and INGXINIT, or INGRLCR), you can switch of lifecycle recording with the following commands, as needed:

- INGRLCR AA ON
- INGRLCR AM ON

To switch off lifecycle recording, issue the following commands:

- INGRLCR AA OFF
- INGRLCR AM OFF

4. Logging checkpoints: When you have prepared and started lifecycle recording, recreate the problem situation. After the problem scenario has been duplicated, switch off lifecycle recording. Because the checkpoints are written in wraparound mode, do not unnecessarily delay switching off. This prevents important checkpoints from being lost or overwritten.

5. Save data to data sets and send to SA z/OS service: Use the following commands to copy the data space to the appropriate data sets that were previously allocated:

- INGRLCR AA SAVE on each automation agent that is involved in debugging the problem
- INGRLCR AM SAVE

Send the data sets to SA z/OS service for evaluation and analysis.

Displaying Lifecycle Recording Syntax:

Issue the command INGRLCR HELP to display a short description of the syntax of the lifecycle recording command. It looks similar to the following output:

INGRLCR : IBM Diagnostic Tool for SA Life Cycle Recording
INGRLCR : target: HELP function:
INGRLCR : extracted: size= dsn= wait=
INGRLCR : SYNTAX
INGRLCR : PARAMETERS: target function KEY=VALUE [ALLOC-parms]
INGRLCR : target : AA|AM
INGRLCR : function : SAVE|ON|OFF|QRY|FLUSH
INGRLCR : KEY=VALUE : SIZE=nn  DSN=a.b.c  WAIT=sec
INGRLCR : ALLOC-parms might be optional NetView ALLOC parameters
INGRLCR : Build Date: 19 Nov 2009
INGRLCR : DONE with return code 0

Workitem Queue Monitoring

Timeout messages (ING008I) can be the result of the fact that more work items are received than the automation manager is able to process. To better understand the nature of received timeout messages, you can now monitor the workitem input queue. This can be achieved with the INGAMS STARTMON/STOPMON parameter.

When the automation manager workitem queue monitoring routine detects that more work items are arriving than can be processed within the monitoring interval, message INGX1011I is issued.
In addition, INGRYAMD traps message INGX1011I with the NetView automation table and saves the relevant data. On request, INGRYAMD creates a file that contains the collected data. The data is separated by semicolons so that the file can be easily imported into a spreadsheet.

The syntax of the INGRYAMD command is as follows:

```
INGRYAMD
```

\textbf{REQ} Specifies the function to be performed. Valid values are:

\begin{itemize}
  \item \textbf{REPORT} Generates a report file that contains the captured data.
  \item \textbf{RESET} Resets the captured data.
\end{itemize}

\textbf{OUTDSN} Specifies the name of the data set that contains the report file. The data set must exist. The minimum record length should be 80.

The report file contains one record for each message that is captured. The format of the records is:

<table>
<thead>
<tr>
<th>Column #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Timestamp when INGX1011I message was captured. The format is yyyy-mm-dd hh:mm:ss</td>
</tr>
<tr>
<td>2</td>
<td>(DIF) The number of work items that were not processed within the last monitoring cycle. It is the number of work items in the queue at the current monitoring time minus the number of work items in the queue at the last monitoring period.</td>
</tr>
<tr>
<td>3</td>
<td>(ABS) The total number of work items in the input and output queue. This also includes the work items that have been processed but not yet run through the cleanup phase.</td>
</tr>
<tr>
<td>4</td>
<td>(QUE) The number of work items in the input queue.</td>
</tr>
<tr>
<td>5</td>
<td>(EXP) The number of expired work items in the input queue.</td>
</tr>
<tr>
<td>6</td>
<td>(CPU) The CPU time of the automation manager.</td>
</tr>
<tr>
<td>7</td>
<td>(OUT) The number of entries in the output queue.</td>
</tr>
<tr>
<td>8</td>
<td>(RSP) The number of response entries in the input queue (for example, Query).</td>
</tr>
<tr>
<td>9</td>
<td>(FAF) The number of fire and forget entries in the input queue.</td>
</tr>
</tbody>
</table>

### Tracing and Debugging for ProcOps and the BCP Internal Interface

The following trace facilities are provided for ProcOps SNMP connections and the BCP Internal Interface that you can use for problem determination purposes.

- “Using AOCTRACE” on page 248
- “Tracing Specific Connections” on page 248
- “First Failure Data Capture Support” on page 249
Using AOCTRACE
To enable the global tracing of all communications related parts of ProcOps-SNMP or BCP Internal Interface, issue the following command:

```
AOCTRACE compname ON
```

Where `compname` is one of the following:

**INGRX810**
INGHWCMD command list

**INGHWCOM**
Communications module

To disable the global trace mode issue command:

```
AOCTRACE compname OFF
```

In global trace mode all currently active connections to SEs or HMCs are traced. This may produce a large amount of data written to the netlog, depending on the number of connections or the amount event data returned from the hardware. In such cases, use the connection-specific trace, as described in "Tracing Specific Connections."

**Note:** For AOCTRACE INGHWCOM, it is important to consider the target hardware initialization from a time point of view as well. Keep in mind that a target hardware connection deals with two different sessions:

1. Event session - established with command INITCOM for connection type INTERNAL (BCPii) or command ISQXIII for connection type SNMP (Processor Operations); the event session remains active till the connection closed with command TERMCOM (BCPii) or ISQXCLS (Processor Operations). The event session handles hardware events and system messages.
2. User session - established for each distinct command issued by the user and terminated after command execution.

In order to get trace messages for BOTH sessions, command AOCTRACE INGHWCOM ON needs to be issued BEFORE command INITCOM or ISQXIII is executed. To stop AOCTRACE, two steps are required:

1. Issue AOCTRACE INGHWCOM OFF.
2. Stop any active target hardware connection (using command TERMCOM or ISQXCLS). If this step is not done, AOCTRACE messages are stopped only for User Command sessions, but will still be issued for Event sessions.

After target hardware connection stop completion, the stopped connections can be restarted immediately. If AOCTRACE INGHWCOM is issued AFTER target hardware connection initialization, trace messages will be written for User Command sessions. To stop tracing, issue command AOCTRACE INGHWCOM OFF.

**Tracing Specific Connections**
If you want to trace the communication to a specific processor or CPC, issue one of the following service commands:

```
INGHWCMD proname TRACE ON
INGHWCMD proname UTRACE ON
```

Where `proname` is the name of the target hardware or processor that is defined in the SA z/OS customization dialog, and specifying UTRACE traces only the user session of the operator that issued the command.
Use the following service commands to disable the trace:

INGHNCMD proname TRACE OFF
INGHNCMD proname UTRACE OFF

Although only a single processor connection is traced, the amount of data that is written to the netlog may be high, especially if the processor connection is through an HMC. In this case, data from other processors that are serviced by this HMC is additionally shown in the trace. It is therefore recommended to limit the tracing time.

**First Failure Data Capture Support**

The INGHWCOM communications module supports the First Failure Data Capture debug aid, provided by NetView for high level language command processors, written in PL/I or C. In case of an INGHWCOM abnormal end, message CNM983E, CNM998E, or CNM999E is issued.

Message CNM983E contains the HLL API user trace information that is maintained by INGHWCOM. This information allows you to determine the internal program flow at error time. If available, save this debug aid information before contacting IBM for service. For more information about the diagnostic messages, see the *Tivoli NetView for z/OS Diagnosis Guide*.

**Using Trace Services for I/O Operations**

To trace the control flow of I/O operations, SA z/OS uses the MVS Component Trace Facility with all its capabilities.

The external writer is not mandatory when the trace is activated. However, when you specify the external writer on the TRACE CT,ON,... command or the WTR option of the CTIIHVZZ PARMLIB member, the writer is internally started if it is not already active.

Before the external writer can be activated the trace data set must be allocated. An appropriate JCL for allocating the data set is provided in the SINGSAMP SA z/OS sample library as the member HSAJCTWR. The recommended data set name is SYS1.HSA.sysname.CTRACE01. An appropriate JCL for running the external writer is provided in the SINGSAMP SA z/OS sample library as the member HSACTWR. In addition, the JCL of the component trace’s external writer must be cataloged into SYS1.PROCLIB. For more information on customizing the component trace refer to *IBM Tivoli System Automation for z/OS Planning and Installation*.

If the trace has not been activated by the CTIIHVZZ PARMLIB member, you can activate it on a z/OS console using the command:

```
TRACE CT,ON,COMP=IHVAS
```

When the system responds with a WTOR expecting the trace options reply:

```
xx[,WTR=HSACTWR],OPTIONS=({MIN|MORE|MAX},ALL|set_of_options),END
```

Where *set_of_options* means a combination of the following options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Header</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM</td>
<td>Switch interface</td>
</tr>
<tr>
<td>API</td>
<td>Programming API</td>
</tr>
<tr>
<td>AU[DET]</td>
<td>Event recording</td>
</tr>
<tr>
<td>B[ASE]</td>
<td>Common routines</td>
</tr>
</tbody>
</table>
You must specify the external writer if you want analyze the trace using IPCS.

You can view the trace options that are currently active with the following command:

```
D TRACE,COMP=IHVAS
```

Unless you want to deactivate the trace automatically when I/O operations terminates, you can deactivate with the following command:

```
TRACE CT,OFF,COMP=IHVAS
```

Before the trace records can be analyzed by IPCS the trace data sets must be made available to IPCS. This is done with the operator command TRACE CT,WTRSTOP=HSACTWR. If the related message indicates that more than one data set contains trace data, the IPCS COPYTRC command must used to combine the data sets into a new one.

To analyze the trace data you need to run the IPCS CTRACE command. You can use either option 2.7.1.D or option 6 from the IPCS primary option menu. Or, you can use the IPCS batch interface (refer to "JCL Sample for Printing the Trace Records" on page 239). At least the following parameters that the command supports must be adapted:

- **COMP**(IHVAS)
- [FULL | SHORT | SUMMARY | TALLY]
- **DSNAME**(trace_data_set_name)

**Note:** In interactive IPCS this is only required when this is not the default data set name.

- **OPTIONS**(filter_options)
  
  Where filter_options is a combination of the following items, separated by a comma:
  - AM
  - API
  - Base
  - CMD
  - COMmunication
  - CONfig
  - DB
  - Query
  - Recovery
  - Schedule
  - XM

<table>
<thead>
<tr>
<th>Option</th>
<th>Header</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMD</td>
<td>Connectivity commands</td>
</tr>
<tr>
<td>COM[MUNICATION]</td>
<td>VTAM, TCP/IP and internal communication</td>
</tr>
<tr>
<td>CON[FIG]</td>
<td>Device and CHannel Path configuration</td>
</tr>
<tr>
<td>DB</td>
<td>Updates of the internal database</td>
</tr>
<tr>
<td>QUERY]</td>
<td>Display and Query commands</td>
</tr>
<tr>
<td>RECOVERY]</td>
<td>Recovery</td>
</tr>
<tr>
<td>SCHEDULE]</td>
<td>Command scheduling</td>
</tr>
<tr>
<td>XM</td>
<td>Cross-memory transport between the I/O operations address space and</td>
</tr>
<tr>
<td></td>
<td>other address spaces, such as IOS or TSO</td>
</tr>
</tbody>
</table>
Debugging and Tracing

- JOBNAME=(jobname[jobname]...)
- ASID=(asid[asid]...)
- TASKID=(taskid[taskid]...)
- MODULE=(module[module]...)
- POS=(pos[pos]...)

The keyword options are connected by a logical AND and the values of a keyword and the remaining options are connected by a logical OR. Each keyword option supports up to 10 different values separated by a comma or a space. Imbedded spaces or commas in a value require delimiters ’..’ or ’...’.

Available task IDs are:
  IAEI INIT
  MAIN
  SCHA SCHB SCHF
  UOBY UOIN
  ZDQM ZSCB ZSDF ZSPL

The trace records are stored intermediately in a data space before they are externalized by the external writer. The data space can be regarded as a wraparound buffer. If you are interested only in the latest trace records you can use the IHVJCT job rather than the external writer and IPCS. The program allows you to print selected trace records from the data space as defined by the filter options above. The JCL can be found in the SINGSAMP SA z/OS sample library.

Collecting the available BCPII session status messages

In case of BCPII session problems, make sure you have collected all SA z/OS WTO messages (msgid ING81*) at the time of the error, that are issued in the SA z/OS NetView domain, where the error occurred. Report these messages (Syslog,Netlog) in case you need to contact IBM Support.

The BCPII session status messages of interest are: ING811I, ING812I, ING813E, ING814E, ING815I, ING816I, ING817I, ING818I and ING819I.
Debugging and Tracing
Appendix D. Hardware Interfaces: Additional Recovery Information

BCP Internal Interface Recovery Information

BCP Internal Interface Recovery Actions

The following section explains manual recovery procedures that might be necessary in case the BCP internal interface communication module INGHWCOM is not responding or is unavailable.

Interface report messages that indicate that module INGHWCOM is unavailable are:

1. AOFA0000 Check Task: <tn>_task_module_INGHWCOM_not_running
2. AOFA0000 Check Task: <tn>_reached QueueLimit_<ql>
3. AOFA0000 Check Task: <tn>_task_is_not_available

The recovery procedure for each of these cases is outlined below.

Recovery Case 1: <tn>_task_module_INGHWCOM_not_running

The recovery procedure for this case is as follows:

1. Make a note of the autotask name <tn> and browse the netlog for messages to or from that task.
2. Search for DSI172I messages that have <tn> as the abending subtask and look for INGHWCOM as the failing module. If this is the case, note down the last report message (AOFA*) prior to the abend from that task.
3. Issue the NetView command EXCMD <tn> LOGOFF (where <tn> is the autotask name that you made a note of in step 1). This will terminate the autotask and free its task storage that was allocated by INGHWCOM. If the autotask has been defined as an HWOPERxx task in the SA z/OS AUTOOPS policy entry, it is automatically restarted and activates the communication module INGHWCOM.

Recovery Case 2: <tn>_reached QueueLimit_<ql>

Each Support Element connection uses a separate autotask. As soon as the "reached QueueLimit" message is returned, this indicates that the previously issued HW request has not yet sent a completion report. The normal recovery process would be to retry the HW command request at a later time.

There are, however, cases where module INGHWCOM has to wait for an indefinite period of time until the HW has completed the request. Usually, this occurs when a Support Element session has been started in SYNC processing mode while a long-running HW command such as ACTIVATE or CBU is still active.

If necessary, you can interrupt and terminate the session, which will not affect any HW command currently in progress on the Support Element. If you terminate the session, you will not receive any completion information for this command.

Issue the following NetView commands:

1. EXCMD <tn> RESET
2. TASKUTIL <tn>
3. EXCMD <tn> LOGOFF

© Copyright IBM Corp. 1996, 2014
BCP Internal Interface Recovery Actions

Command 1 on page 253 terminates the communication module INGHWCOM in task <tn>. The session between NetView and the SE/HMC will be closed normally and the allocated task storage will be freed.

With command 2 on page 253 you can control whether module INGHWCOM is no longer the active command of <tn> and the amount of storage still allocated by <tn>.

With command 3 on page 253 you terminate autotask <tn>, which will initiate an autotask restart and the activation of INGHWCOM.

If command 2 on page 253 indicates that command 1 on page 253 did not end INGHWCOM, command 1 on page 253 can be repeated with RESET IMMED. This will abnormally terminate INGHWCOM, but SE/HMC session termination will not be performed. Note that a 'STOP TASK' command will end the autotask, but will not invoke normal SE/HMC session termination.

Recovery Case 3: <tn>_task_is_not_available

There are cases where a BCP internal interface autotask, previously defined as an HWOPERxx task with an SA z/OS AUTOOPS policy entry, cannot be used. In these cases, a task definition problem or a mismatch of your current ACF configuration might be the reason for the problem.

The recovery procedure for this case is as follows:
1. Verify the AUTOOPS definition statements in your PDB.
2. Verify that the corresponding DSIOPF statements have AOFPRFHW defined as the operator profile name.
3. Perform a Build operation after the policy changes.
4. Activate your policy changes, for example by performing an ACF REFRESH.

Checking BCP Internal Interface Interlocks

If applications that exploit the BCP Internal Interface, such as GDPS, seem to hang or loop while a HW command or HW query is active, this may indicate that internal locks were not freed. Unconditional task interrupts during INGHWCMD processing or an unconditional task termination can cause such a situation.

You can use the LCHECK function of the IBM Service command INGHWSRV to clear blocking locks. For more information about the service command, use the SA z/OS command HELP INGHWSRV.

Terminating NetView and the BCP Internal Interface

When terminating NetView, a 'CLOSE NORMAL' command should first be issued to give INGHWCOM enough time to end all its SE/HMC sessions. Then, after a predefined period of time, a 'CLOSE IMMED' command could be used to limit NetView's overall termination time.

How to Switch the IP Address for Target Hardware Used by Processor Operations

This description is only valid for target hardware (defined as 'Processors' in the SA z/OS customization dialog) with a connection type of SNMP.

When defining a processor as the target hardware for Processor Operations, two IP addresses can be provided for communication with either the Support Element (SE) or the Hardware Management Console (HMC). At least one IP address for the
BCP Internal Interface Recovery Actions

SE or HMC must be provided; optionally, a secondary IP address can be provided for this processor. If two IP addresses are provided and both addresses are SE ones, ensure that the SE has an alternate network adapter card installed and that it is operational.

If you need to switch between these two addresses (for example, in case of a SE network adapter card failure), switching to the secondary IP address can be easily performed using Processor Operations command:

```
isqipswt <target-hardware-name>
```

For a detailed command description, refer to IBM Tivoli System Automation for z/OS Operator’s Commands.

---

How to Terminate BCP Internal Interface Connections or Clear Hardware Control Blocks after a Connection Failure

There are occasions when it may be necessary to terminate BCP internal interface connections, for example, prior to issuing a CLOSE IMMED to terminate IBM Tivoli NetView for z/OS and IBM Tivoli System Automation for z/OS.

This can be done using the following Systems Operations command:

```
inghwsrv term
```

This command performs a TERMCOM request for each CPC that is found and then attempts to disable the HSAET32 resource.

Occasionally it may be necessary to clear hardware control blocks after a connection failure with the target hardware.

This can be done using the following Systems Operations command:

```
inghwsrv target_hardware_name force
```

For a detailed command description, see IBM Tivoli System Automation for z/OS Operator’s Commands.

See also the appendix “Controlling Access to the Processor Hardware Functions” in IBM Tivoli System Automation for z/OS Planning and Installation.

---

Recovery of Processor Operations Target System Connections in the event of Hardware Management Console Outage

Processor Operations target system connections rely on a fully operational Support Element (SE) or a Hardware Management Console (HMC), in order to monitor active target system connections or to execute commands against defined target systems.

If the connection to the HMC is lost, there are 2 major reasons for the failure:
1. Shutdown of the HMC for maintenance reasons (planned outage).
2. Technical failure or accidental shutdown of the HMC (unplanned outage).

In case of a planned outage, it is strongly recommended to close any active target system connections for that HMC as follows:
1. Issue Processor Operations host-based command ISQXCLS, or
2. Issue Line command 'C' in the Processor Operations target system summary panel (see host-based command ISQXDST), or
3. Issue Processor Operations common command CTRLCONS (see the note at the end of this section regarding command CTRLCONS).

Once the HMC is active again, the previously closed target system connections can be re-initiated using host-based command ISQXIII, or line command 'T' in the Processor Operations target system summary panel (see host-based command ISQXDST).

For an unplanned outage, check on the duration of the outage and if Processor Operations has received an error notification (event message from the HMC).

If the interrupt is for a short period, and if Processor Operations receives an event failure from the HMC (messages AOFA0998 and AOFA0999 as described in Appendix D - Response Messages, Error Strings, Condition Codes of IBM Tivoli System Automation for z/OS Messages and Codes), Processor Operations might succeed in re-establishing the broken connection on its own immediately.

If Processor Operations does not receive any event failure message, the connection will remain in an INITIALIZED status, though it may change to, for example, TARGET HARDWARE PROBLEM if the status polling process detects there is a problem. If the target system connection has this problem status, AND if the reason obviously is due to an outage of the HMC, AND if Processor Operations did not succeed in automatically re-establishing the target system connection, the recommended recovery steps are as follows:
1. Wait till the affected HMC is available again,
2. Issue command ISQXCLS for all affected target systems to properly close and clean up the Processor Operations connection,
3. Issue command ISQXIII for all affected target systems to re-establish Processor Operations connection.

**Note:** In order to shutdown the HMC in a controlled manner, the Processor Operations common command CTRLCONS can be used (see IBM Tivoli System Automation for z/OS Operator’s Commands for a detailed description). CTRLCONS will automatically close any still active target system connections to that HMC.
Glossary

This glossary includes terms and definitions from:

- The *Information Technology Vocabulary* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.

The following cross-references are used in this glossary:

**Contrast with.** This refers to a term that has an opposed or substantively different meaning.

**Deprecated term for.** This indicates that the term should not be used. It refers to a preferred term, which is defined in its proper place in the glossary.

**See.** This refers the reader to multiple-word terms in which this term appears.

**See also.** This refers the reader to terms that have a related, but not synonymous, meaning.

**Synonym for.** This indicates that the term has the same meaning as a preferred term, which is defined in the glossary.

**Synonymous with.** This is a backward reference from a defined term to all other terms that have the same meaning.

A

ACF See *automation configuration file*

ACF/NCP Advanced Communications Function for the Network Control Program. See *Advanced Communications Function* and *Network Control Program*

ACF/VTAM Advanced Communications Function for the Virtual Telecommunications Access Method. Synonym for VTAM. See *Advanced Communications Function* and *Virtual Telecommunications Access Method*

active monitoring In SA z/OS automation control file, the acquiring of resource status information by soliciting such information at regular, user-defined intervals. See also *passive monitoring*

adapter Hardware card that enables a device, such as a workstation, to communicate with another device, such as a monitor, a printer, or some other I/O device.

adjacent hosts Systems connected in a peer relationship using adjacent NetView sessions for purposes of monitoring and control.

adjacent NetView In SA z/OS, the system defined as the communication path between two SA z/OS systems that do not have a direct link. An adjacent NetView is used for message forwarding and as a communication link between two SA z/OS systems. For example, the adjacent NetView is used when sending responses from a focal point to a remote system.

Advanced Communications Function (ACF) A group of IBM licensed programs (principally VTAM, TCAM, NCP, and SSP) that use the concepts of Systems Network Architecture (SNA), including distribution of function and resource sharing.

advanced program-to-program communication (APPC) A set of inter-program communication services that support cooperative transaction processing in a Systems...
Network Architecture (SNA) network. APPC is the implementation, on a given system, of SNA’s logical unit type 6.2.

Advanced Workload Analysis Reporter (zAware)
IBM analytics appliance which requires an exclusive user partition as runtime environment. The appliance is a combination of a LPAR running in a special mode, an embedded OS (Linux on System Z) and an application layer. Customers can use the appliance to monitor the console message streams of other LPARs running in the same System z cluster and create trend reports. Exploiting zAware and these trend reports can help to better predict OS outages or performance degradations and initiate proactive clusters.

alert
In SNA, a record sent to a system problem management focal point or to a collection point to communicate the existence of an alert condition.

In NetView, a high-priority event that warrants immediate attention. A database record is generated for certain event types that are defined by user-constructed filters.

alert condition
A problem or impending problem for which some or all of the process of problem determination, diagnosis, and resolution is expected to require action at a control point.

alert focal-point system
See NPDA focal point system

alert threshold
An application or volume service value that determines the level at which SA z/OS changes the associated icon in the graphical interface to the alert color. SA z/OS may also issue an alert. See warning threshold

AMC See Automation Manager Configuration

The Auto Msg Classes entry type.

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.

**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.

**basic mode**
A central processor mode that does not use logical partitioning. Contrast with [logically partitioned mode](#).

**BCP** See [Base Control Program](#).

**blade**
A hardware unit that provides application-specific services and components. The consistent size and shape (or form factor) of each blade allows it to fit in a BladeCenter chassis.

**BladeCenter chassis**
A modular chassis that can contain multiple blades, allowing the individual blades to share resources such as management, switch, power, and blower modules.

**BookManager®**
An IBM product that lets users view softcopy documents on their workstations.

**C**

**central processor (CP)**
The part of the computer that contains the sequencing and processing facilities for instruction execution, initial program load (IPL), and other machine operations.

**central processor complex (CPC)**
A physical collection of hardware that consists of central storage, (one or more) central processors, (one or more) timers, and (one or more) channels.

**central site**
In a distributed data processing network, the central site is usually defined as the focal point for alerts, application design, and remote system management tasks such as problem management.

**CFR/CFS and ISC/ISR**
I/O operations can display and return data about integrated system channels (ISC) connected to a coupling facility and coupling facility receiver (CFR) channels and coupling facility sender (CFS) channels.

**channel**
A path along which signals can be sent; for example, data channel, output channel. See also link.

**channel path identifier**
A system-unique value assigned to each channel path.

**channel-attached**
Attached directly by I/O channels to a host processor (for example, a channel-attached device).

Attached to a controlling unit by cables, rather than by telecommunication lines. Contrast with link-attached. Synonymous with local.

**CHPID**
In SA z/OS, channel path ID; the address of a channel.

**CHPID port**
A label that describes the system name, logical partitions, and channel paths.

**CI** See [console integration](#).

**CICS/VS**
Customer Information Control System for Virtual Storage. See [Customer Information Control System](#).

**CLIST** See [command list](#).

**clone**
A set of definitions for application
instances that are derived from a basic application definition by substituting a number of different system-specific values into the basic definition.

clone ID
A generic means of handling system-specific values such as the MVS SYSCLONE or the VTAM subarea number. Clone IDs can be substituted into application definitions and commands to customize a basic application definition for the system that it is to be instantiated on.

CNC
A channel path that transfers data between a host system image and an ESCON control unit. It can be point-to-point or switchable.

command
A request for the performance of an operation or the execution of a particular program.

command facility
The component of NetView that is a base for command processors that can monitor, control, automate, and improve the operation of a network. The successor to NCCF.

command list (CLIST)
A list of commands and statements, written in the NetView command list language or the REXX language, designed to perform a specific function for the user. In its simplest form, a command list is a list of commands. More complex command lists incorporate variable substitution and conditional logic, making the command list more like a conventional program. Command lists are typically interpreted rather than being compiled.

In SA z/OS, REXX command lists that can be used for automation procedures.

command procedure
In NetView, either a command list or a command processor.

command processor
A module designed to perform a specific function. Command processors, which can be written in assembler or a high-level language (HLL), are issued as commands.

command processor control block
An I/O operations internal control block that contains information about the command being processed.

Command Tree/2
An OS/2-based program that helps you build commands on an OS/2 window, then routes the commands to the destination you specify (such as a 3270 session, a file, a command line, or an application program). It provides the capability for operators to build commands and route them to a specified destination.

common commands
The SA z/OS subset of the CPC operations management commands.

Common User Access (CUA) architecture
Guidelines for the dialog between a human and a workstation or terminal.

communication controller
A type of communication control unit whose operations are controlled by one or more programs stored and executed in the unit or by a program executed in a processor to which the controller is connected. It manages the details of line control and the routing of data through a network.

communication line
Deprecated term for telecommunication line.

connectivity view
In SA z/OS, a display that uses graphic images for I/O devices and lines to show how they are connected.

console automation
The process of having NetView facilities provide the console input usually handled by the operator.

console connection
In SA z/OS, the 3270 or ASCII (serial) connection between a PS/2 computer and a target system. Through this connection, the workstation appears (to the target system) to be a console.

console integration (CI)
A hardware facility that if supported by an operating system, allows operating system messages to be transferred through an internal hardware interface for
display on a system console. Conversely,
it allows operating system commands
entered at a system console to be
transferred through an internal hardware
interface to the operating system for
processing.

consoles
Workstations and 3270-type devices that
manage your enterprise.

Control units
Hardware units that control I/O
operations for one or more devices. You
can view information about control units
through I/O operations, and can start or
stop data going to them by blocking and
unblocking ports.

ccontroller
A unit that controls I/O operations for
one or more devices.

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

couple data set
A data set that is created through the XCF
couple data set format utility and,
depending on its designated type, is
shared by some or all of the z/OS
systems in a sysplex. See also sysplex,
couple data set and XCF couple data set.

coupling facility
The hardware element that provides
high-speed caching, list processing, and
locking functions in a sysplex.

CP  See central processor

CPC  See central processor complex

CPC operations management commands
A set of commands and responses for
controlling the operation of System/390®
CPCs.

CPC subset
All or part of a CPC. It contains the
minimum resource to support a single
control program.

CPCB  See command processor control block

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

CPU  Central processing unit. Deprecated term
for processor

Glossary   263
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 system associated with IBM automation products.

**NMC focal point system**
The NMC focal point system is a NetView system with an attached workstation server and LAN that gathers information about the state of the network. This focal point system uses RODM to store the data it collects in the data model. The information stored in RODM can be accessed from any LAN-connected workstation with NetView Management Console installed.
NPDA focal point system. This is a NetView system that collects all the NPDA alerts that are generated within your enterprise. It is supported by NetView. If you have SA z/OS installed the NPDA focal point system must be the same as your NMC focal point system. The NPDA focal point system is also known as the alert focal point system.

SA z/OS Processor Operations focal point system. This is a NetView system that has SA z/OS host code installed. The SA z/OS Processor Operations focal point system receives messages from the systems and operator consoles of the machines that it controls. It provides full systems and operations console function for its target systems. It can be used to IPL these systems. Note that some restrictions apply to the Hardware Management Console for an S/390® microprocessor cluster.

SA z/OS SDF focal point system. The SA z/OS SDF focal point system is an SA z/OS NetView system that collects status information from other SA z/OS NetViews within your enterprise.

Status focal point system. In NetView, the system to which STATMON, VTAM and NLDLM send status information on network resources. If you have a NMC focal point, it must be on the same system as the Status focal point.

Hardware Management Console. Although not listed as a focal point, the Hardware Management Console acts as a focal point for the console functions of an S/390 microprocessor cluster. Unlike all the other focal points in this definition, the Hardware Management Console runs on a LAN-connected workstation.

frame For a System/390 microprocessor cluster, a frame contains one or two central processor complexes (CPCs), support elements, and AC power distribution.

full-screen mode In NetView, a form of panel presentation that makes it possible to display the contents of an entire workstation screen at once. Full-screen mode can be used for fill-in-the-blanks prompting. Contrast with line mode.

gateway session An NetView-NetView Task session with another system in which the SA z/OS outbound gateway operator logs onto the other NetView session without human operator intervention. Each end of a gateway session has both an inbound and outbound gateway operator.

generic alert Encoded alert information that uses code points (defined by IBM and possibly customized by users or application programs) stored at an alert receiver, such as NetView.

group A collection of target systems defined through configuration dialogs. An installation might set up a group to refer to a physical site or an organizational or application entity.

group entry A construct, created with the customization dialogs, used to represent and contain policy for a group.

group entry type A collection of target systems defined through the customization dialog. An installation might set up a group to refer to a physical site or an organizational entity. Groups can, for example, be of type STANDARD or SYSPLEX.

Hardware Management Console (HMC) A user interface through which data center personnel configure, control, monitor, and manage System z hardware and software resources. The HMC communicates with each central processor complex (CPC) through the Support Element. On an IBM zEnterprise 196 (z196), using the Unified Resource Manager on the HMCs or Support Elements, personnel can also create and manage an ensemble.

Hardware Management Console Application (HWMCA) A direct-manipulation object-oriented graphical user interface that provides a single point of control and single system image for hardware elements. The HWMCA provides grouping support, aggregated and real-time system status.
using colors, consolidated hardware messages support, consolidated operating system messages support, consolidated service support, and hardware commands targeted at a single system, multiple systems, or a group of systems.

**heartbeat**
In SA z/OS, a function that monitors the validity of the status forwarding path between remote systems and the NMC focal point, and monitors the availability of remote z/OS systems, to ensure that status information displayed on the SA z/OS workstation is current.

**help panel**
An online panel that tells you how to use a command or another aspect of a product.

**hierarchy**
In the NetView program, the resource types, display types, and data types that make up the organization, or levels, in a network.

**high-level language (HLL)**
A programming language that provides some level of abstraction from assembler language and independence from a particular type of machine. For the NetView program, the high-level languages are PL/I and C.

**HLL** See [high-level language](#)

**host (primary processor)**
The processor that you enter a command at (also known as the issuing processor).

**host system**
In a coupled system or distributed system environment, the system on which the facilities for centralized automation run. SA z/OS publications refer to target systems or focal-point systems instead of hosts.

**HWMCA**
See [Hardware Management Console Application](#)

**Hypervisor**
A program that allows multiple instances of operating systems or virtual servers to run simultaneously on the same hardware device. A hypervisor can run directly on the hardware, can run within an operating system, or can be imbedded in platform firmware. Examples of hypervisors include PR/SM, z/VM, and PowerVM Enterprise Edition.

**IBM blade**
A customer-acquired, customer-installed select blade to be managed by IBM zEnterprise Unified Resource Manager. One example of an IBM blade is a POWER7 blade.

**IBM Smart Analyzer for DB2 for z/OS**
An optimizer that processes certain types of data warehouse queries for DB2 for z/OS.

**IBM System z Application Assist Processor (zAAP)**
A specialized processor that provides a Java execution environment, which enables Java-based web applications to be integrated with core z/OS business applications and backend database systems.

**IBM System z Integrated Information Processor (zIIP)**
A specialized processor that provides computing capacity for selected data and transaction processing workloads and for selected network encryption workloads.

**IBM Websphere DataPower Integration Appliance X150 for zEnterprise (DataPower X150z)**
A purpose-built appliance that simplifies, helps secure, and optimizes XML and Web services processing.

**IBM 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 (LTK); name by which the I/O resource or object is known; stored in IODF.

IOCDS
See input/output configuration data set.

IOSP
See input/output support processor.

IPL
See initial program load.

ISPF
See Interactive System Productivity Facility.

ISPF console
You log on to ISPF from this 3270-type console to use the runtime panels for I/O operations and SA z/OS customization panels.

issuing host
The base program that you enter a command for processing with. See primary host.

J
JCL
See job control language.

JES
See job entry subsystem.

JES2
An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system. In an installation with more than one processor, each JES2 processor independently controls its job input, scheduling, and output processing. See also job entry subsystem and JES3.

JES3
An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system. In complexes that have several loosely coupled processing units, the JES3 program manages processors so that the global processor exercises centralized control over the local processors and distributes jobs to them using a common job queue. See also job entry subsystem and JES2.

job
A set of data that completely defines a unit of work for a computer. A job usually includes all necessary computer programs, linkages, files, and instructions to the operating system.

An address space.

job control language (JCL)
A problem-oriented language designed to express statements in a job that are used to identify the job or describe its requirements to an operating system.

job entry subsystem (JES)
An IBM licensed program that receives jobs into the system and processes all output data that is produced by jobs. In SA z/OS publications, JES refers to JES2 or JES3, unless otherwise stated. See also JES2 and JES3.

K
Kanji
An ideographic character set used in Japanese. See also double-byte character set.

L
LAN
See local area network.

line mode
A form of screen presentation in which the information is presented 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

**logicaly 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.

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.
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 (NMC)
A function of the NetView program that provides a graphic, topological presentation of a network that is controlled by the NetView program. It provides the operator different views of a network, multiple levels of graphical detail, and dynamic resource status of the network. This function consists of a series of graphic windows that allows you to manage the network interactively. Formerly known as the NetView Graphic Monitor Facility (NGMF).

NetView message table
See NetView automation table.

NetView paths via logical unit (LU 6.2)
A type of network-accessible port (VTAM connection) that enables end users to gain access to SNA network resources and communicate with each other. LU 6.2 permits communication between processor operations and the workstation. See logical unit 6.2.

NetView-NetView task (NNT)
The task that a cross-domain NetView operator session runs under. Each NetView program must have a NetView-NetView task to establish one NNT session. See also operator station task.

NetView-NetView task session
A session between two NetView programs that runs under a NetView-NetView task. In SA z/OS, NetView-NetView task
sessions are used for communication between focal point and remote systems.

network
An interconnected group of nodes.

In data processing, a user application network. See SNA network.

network accessible unit (NAU)
In SNA networking, any device on the network that has a network address, including a logical unit (LU), physical unit (PU), control point (CP), or system services control point (SSCP). It is the origin or the destination of information transmitted by the path control network. Synonymous with network addressable unit.

network addressable unit (NAU)
Synonym for network accessible unit.

Network Communications Control Facility (NCCF)
The operations control facility for the network. NCCF consists of a presentation service, command processors, automation based on command lists, and a transaction processing structure on which the network management applications NLDM and NPDA are built. NCCF is a precursor to the NetView command facility.

Network Control Program (NCP)
An IBM licensed program that provides communication controller support for single-domain, multiple-domain, and interconnected network capability. Its full name is Advanced Communications Function for the Network Control Program.

network control program (NCP)
A program that controls the operation of a communication controller.

A program used for requests and responses exchanged between physical units in a network for data flow control.

Network Problem Determination Application (NPDA)
An NCCF application that helps you identify network problems, such as hardware, software, and microcode, from a central control point using interactive display methods. The alert manager for the network. The precursor of the NetView hardware monitor.

Networking NetView
In SA z/OS the NetView that performs network management functions, such as managing the configuration of a network. In SA z/OS it is common to also route alerts to the Networking NetView.

NGMF
Deprecated term for NetView Management Console.

NGMF focal-point system
Deprecated term for NMC focal point system.

NIP See nucleus initialization program.

NMC focal point system
See focal point system.

NMC workstation
The NMC workstation is the primary way to dynamically monitor SA z/OS systems. From the windows, you see messages, monitor status, view trends, and react to changes before they cause problems for end users. You can use multiple windows to monitor multiple views of the system.

NNT See NetView-NetView task.

notification message
An SA z/OS message sent to a human notification operator to provide information about significant automation actions. Notification messages are defined using the customization dialogs.

notification operator
A NetView console operator who is authorized to receive SA z/OS notification messages. Authorization is made through the customization dialogs.

NPDA
See Network Problem Determination Application.

NPDA focal-point system
See focal point system.

NTRI See NCP/token ring interconnection.

nucleus initialization program (NIP)
The program that initializes the resident control program; it allows the operator to
request last-minute changes to certain options specified during system generation.

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

OPC/ESA

See Operations Planning and Control/Enterprise Systems Architecture

Open Systems Adapter (OSA)

I/O operations can display the Open System Adapter (OSA) channel logical definition, physical attachment, and status. You can configure an OSA channel on or off.

Operating System (OS)

Software that controls the execution of programs and that may provide services such as resource allocation, scheduling, input/output control, and data management. Although operating systems are predominantly software, partial hardware implementations are possible. (T)

Operations

The real-time control of a hardware device or software function.

Operations Planning and Control/Advanced (OPC/A)

A set of IBM licensed programs that automate, plan, and control batch workload. OPC/A analyzes system and workload status and submits jobs accordingly.

Operations Planning and Control/Enterprise Systems Architecture (OPC/ESA)

A set of IBM licensed programs that automate, plan, and control batch workload. OPC/ESA analyzes system and workload status and submits jobs accordingly. The successor to OPC/A.

Operator

A person who keeps a system running.

A person or program responsible for managing activities controlled by a given piece of software such as z/OS, the NetView program, or IMS.

A person who operates a device.

In a language statement, the lexical entity that indicates the action to be performed on operands.

Operator Console

A functional unit containing devices that are used for communications between a computer operator and a computer. (T)

A display console used for communication between the operator and the system, used primarily to specify information concerning application programs and I/O operations and to monitor system operation.

In SA z/OS, a console that displays output from and sends input to the operating system (z/OS, LINUX, VM, VSE). Also called operating system console.

In the SA z/OS operator commands and configuration dialogs, OC is used to designate a target system operator console.

Operator Station Task (OST)

The NetView task that establishes and maintains the online session with the network operator. There is one operator station task for each network operator who logs on to the NetView program.

Operator View

A set of group, system, and resource definitions that are associated together for monitoring purposes. An operator view appears as a graphic display in the graphical interface showing the status of the defined groups, systems, and resources.

OperatorView Entry

A construct, created with the
customization dialogs, used to represent and contain policy for an operator view.

**optimizer**
A special-purpose hardware component or appliance that can perform a limited set of specific functions with optimized performance when compared to a general-purpose processor. Because of its limited set of functions, an optimizer is an integrated part of a processing environment, rather than a stand-alone unit. One example of an optimizer is the IBM Smart Analytics Optimizer for DB2 for z/OS.

**OS** See [operating system](#).

**OSA** See [Open Systems Adapter](#).

**OST** See [operator station task](#).

**outbound**
In SA z/OS, messages or commands from the focal-point system to the target system.

**outbound gateway operator**
The automation operator that establishes connections to other systems. The outbound gateway operator handles communications with other systems through a gateway session. The automation operator sends messages, commands, and responses to the inbound gateway operator at the receiving system.

**P**

**page**
The portion of a panel that is shown on a display surface at one time. To transfer instructions, data, or both between real storage and external page or auxiliary storage.

**panel**
A formatted display of information that appears on a terminal screen. Panels are full-screen 3270-type displays with a monospaced font, limited color and graphics.

By using SA z/OS panels you can see status, type commands on a command line using a keyboard, configure your system, and pass thru 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, [MP](#), and [side](#).

**partitioned data set (PDS)**
A data set in direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

**passive monitoring**
In SA z/OS, the receiving of unsolicited messages from z/OS systems and their resources. These messages can prompt updates to resource status displays. See also [active monitoring](#).

**PCE**
A processor controller. Also known as the support processor or service processor in some processor families.

**PDB** See [policy database](#).

**PDS** See [partitioned data set](#).
physical partition
Part of a CPC that operates as a CPC in its own right, with its own copy of the operating system.

physical unit (PU)
In SNA, the component that manages and monitors the resources (such as attached links and adjacent link stations) of a node, as requested by a system services control point (SSCP) through an SSCP-PU session. An SSCP activates a session with the physical unit to indirectly manage, through the PU, resources of the node such as attached links.

physically partitioned (PP) configuration
A mode of operation that allows a multiprocessor (MP) system to function as two or more independent CPCs having separate power, utilities, and maintenance boundaries. Contrast with single-image mode.

PLEXID Group
PLEXID group or "extended XCF communication group" is a term used in conjunction with a sysplex. The PLEXID group includes System Automation Agents for a subset of a sysplex or for the entire sysplex. It is used to provide XCF communication beyond the SAplex boundaries. For a detailed description, refer to "Defining the Extended XCF Communication Group" in IBM Tivoli System Automation for z/OS Planning and Installation.

POI See program operator interface

policy
The automation and monitoring specifications for an SA z/OS enterprise. See IBM Tivoli System Automation for z/OS Defining Automation Policy.

policy database
The automation definitions (automation policy) that the automation programmer specifies using the customization dialog is stored in the policy database. Also known as the PDB. See also automation policy.

POR See power-on reset

port System hardware that the I/O devices are attached to.
In an ESCON switch, a port is an addressable connection. The switch routes data through the ports to the channel or control unit. Each port has a name that can be entered into a switch matrix, and you can use commands to change the switch configuration.

An access point (for example, a logical unit) for data entry or exit.

A functional unit of a node that data can enter or leave a data network through.

In data communication, that part of a data processor that is dedicated to a single data channel for the purpose of receiving data from or transmitting data to one or more external, remote devices.

time

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. ProOps communicates via the PSM with the VM guest systems that are defined as target systems within ProcOps.

**product automation**
Automation integrated into the base of SA z/OS for the products CICS, DB2, IMS, TWS (formerly called features).

**program operator interface (POI)**
A NetView facility for receiving VTAM messages.

**program to program interface (PPI)**
A NetView function that allows user programs to send or receive data buffers from other user programs and to send alerts to the NetView hardware monitor from system and application programs.

**protocol**
In SNA, the meanings of, and the sequencing rules for, requests and responses used for managing the network, transferring data, and synchronizing the states of network components.

**proxy resource**
A resource defined like an entry type APL representing a processor operations target system.

**PSM** See ProcOps Service Machine.

**PU** See physical unit.

**R**

**RACF** See Resource Access Control Facility.

**remote system**
A system that receives resource status information from an SA z/OS focal-point system. An SA z/OS remote system is defined as part of the same SA z/OS enterprise as the SA z/OS focal-point system to which it is related.

**requester**
A workstation from that user can log on to a domain from, that is, to the servers belonging to the domain, and use network resources. Users can access the shared resources and use the processing capability of the servers, thus reducing hardware investment.

**resource**
Any facility of the computing system or operating system required by a job or task, and including main storage, input/output devices, the processing unit, data sets, and control or processing programs.
In NetView, any hardware or software that provides function to the network.
In SA z/OS, any z/OS application, z/OS component, job, device, or target system capable of being monitored or automated through SA z/OS.

**Resource Access Control Facility (RACF)**
A program that can provide data security for all your resources. RACF protects data from accidental or deliberate unauthorized disclosure, modification, or destruction.
resource group
A physically partitionable portion of a processor. Also known as a side.

Resource Measurement Facility (RMF)
A feature of z/OS that measures selected areas of system activity and presents the data collected in the format of printed reports, System Management Facility (SMF) records, or display reports.

Resource Object Data Manager (RODM)
In NetView for z/OS, a component that provides an in-memory cache for maintaining real-time data in an address space that is accessible by multiple applications. RODM also allows an application to query an object and receive a rapid response and act on it.

resource token
A unique internal identifier of an ESCON resource or resource number of the object in the IODF.

restart automation
Automation provided by SA z/OS that monitors subsystems to ensure that they are running. If a subsystem fails, SA z/OS attempts to restart it according to the policy in the automation configuration file.

Restructured Extended Executor (REXX)
A general-purpose, high-level, programming language, particularly suitable for EXEC procedures or programs for personal computing, used to write command lists.

return code
A code returned from a program used to influence the issuing of subsequent instructions.

REXX See Restructured Extended Executor

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

RMF See Resource Measurement Facility

RODM See Resource Object Data Manager

S
SAF See Security Authorization Facility

SA IOM
See System Automation for Integrated Operations Management

SAplex
SAplex or "SA z/OS Subplex" is a term used in conjunction with a sysplex. In fact, a SAplex is a subset of a sysplex. However, it can also be a sysplex. For a detailed description, refer to "Using SA z/OS Subplexes" in IBM Tivoli System Automation for z/OS Planning and Installation.

SA z/OS
See System Automation for z/OS

SA z/OS customization dialogs
An ISPF application through which the SA z/OS policy administrator defines policy for individual z/OS systems and builds automation control data and RODM load function files.

SA z/OS customization focal point system
See focal point system

SA z/OS data model
The set of objects, classes and entity relationships necessary to support the function of SA z/OS and the NetView automation platform.

SA z/OS enterprise
The group of systems and resources defined in the customization dialogs under one enterprise name. An SA z/OS enterprise consists of connected z/OS systems running SA z/OS.

SA z/OS focal point system
See focal point system

SA z/OS policy
The description of the systems and resources that make up an SA z/OS enterprise, together with their monitoring and automation definitions.

SA z/OS policy administrator
The member of the operations staff who is responsible for defining SA z/OS policy.

SA z/OS satellite
If you are running two NetViews on a z/OS system to split the automation and networking functions of NetView, it is common to route alerts to the Networking NetView. For SA z/OS to process alerts properly on the Networking NetView, you
must install a subset of SA z/OS code, called an SA z/OS satellite on the Networking NetView.

**SA z/OS SDF focal point system**
See [focal point system](#).

**SCA**
In SA z/OS, system console A, the active system console for a target hardware. Contrast with [SCB](#).

**SCB**
In SA z/OS, system console B, the backup system console for a target hardware. Contrast with [SCA](#).

**screen**
Deprecated term for [panel](#).

**screen handler**
In SA z/OS, software that interprets all data to and from a full-screen image of a target system. The interpretation depends on the format of the data on the full-screen image. Every processor and operating system has its own format for the full-screen image. A screen handler controls one PS/2 connection to a target system.

**SDF**
See [status display facility](#).

**SDLC**
See [synchronous data link control](#).

**SDSF**
See [System Display and Search Facility](#).

**secondary system**
A system is a secondary system for an application if it is defined to automation on that system, but the application is not normally meant to be running there. Secondary systems are systems to which an application can be moved in the event that one or more of its primary systems are unavailable. SA z/OS does not start the application on its secondary systems.

**Security Authorization Facility (SAF)**
An MVS interface with which programs can communicate with an external security manager, such as RACF.

**server**
A server is a workstation that shares resources, which include directories, printers, serial devices, and computing powers.

**service language command (SLC)**
The line-oriented command language of processor controllers or service processors.

**service period**
Service periods allow the users to schedule the availability of applications. A service period is a set of time intervals (service windows), during which an application should be active.

**service processor (SVP)**
The name given to a processor controller on smaller System/370 processors.

**service threshold**
An SA z/OS policy setting that determines when to notify the operator of deteriorating service for a resource. See also [alert threshold](#) and [warning threshold](#).

**session**
In SNA, a logical connection between two network addressable units (NAUs) that can be activated, tailored to provide various protocols, and deactivated, as requested. Each session is uniquely identified in a transmission header by a pair of network addresses identifying the origin and destination NAUs of any transmissions exchanged during the session.

**session monitor**
The component of the NetView program that collects and correlates session-related data and provides online access to this information. The successor to NLDM.

**shutdown automation**
SA z/OS-provided automation that manages the shutdown process for subsystems by issuing shutdown commands and responding to prompts for additional information.

**side**
A part of a partitionable CPC that can run as a physical partition and is typically referred to as the A-side or the B-side.

**Simple Network Management Protocol (SNMP)**
A set of protocols for monitoring systems and devices in complex networks. Information about managed devices is defined and stored in a Management Information Base (MIB).

**single image**
A processor system capable of being physically partitioned that has not been physically partitioned. Single-image systems can be target hardware processors.
single-MVS environment
An environment that supports one MVS image. See also MVS image.

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 independent of, or asynchronously with, a controlling system.

In SA z/OS, an z/OS application or subsystem defined to SA z/OS.

subsystem interface (SSI)
The z/OS interface over which all messages sent to the z/OS console are broadcast.

support element
A hardware unit that provides communications, monitoring, and diagnostic functions to a central processor complex (CPC).

support processor
Another name given to a processor
controller on smaller System/370 processors. See service processor.

SVP
See service processor.

switch identifier
The switch device number (swchdevn), the logical switch number (LSN) and the switch name.

switches
ESCON directors are electronic units with ports that dynamically switch to route data to I/O devices. The switches are controlled by I/O operations commands that you enter on a workstation.

symbolic destination name (SDN)
Used locally at the workstation to relate to the VTAM application name.

synchronous data link control (SDLC)
A discipline for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or nonswitched links. The configuration of the link connection may be point-to-point, multipoint, or loop. SDLC conforms to subsets of the Advanced Data Communication Control Procedures (ADCCP) of the American National Standards Institute and High-Level Data Link Control (HDLC) of the International Standards Organization.

SYSINFO Report
An RMF report that presents an overview of the system, its workload, and the total number of jobs using resources or delayed for resources.

SysOps
See system operations.

sysplex
A set of z/OS systems communicating and cooperating with each other through certain multisystem hardware components (coupling devices and timers) and software services (couple data sets).

In a sysplex, z/OS provides the coupling services that handle the messages, data, and status for the parts of a multisystem application that has its workload spread across two or more of the connected processors, sysplex timers, coupling facilities, and couple data sets (which contains policy and states for automation).

A Parallel Sysplex is a sysplex that includes a coupling facility.

sysplex application group
A sysplex application group is a grouping of applications that can run on any system in a sysplex.

sysplex couple data set
A couple data set that contains sysplex-wide data about systems, groups, and members that use XCF services. All z/OS systems in a sysplex must have connectivity to the sysplex couple data set. See also couple data set.

Sysplex Timer
An IBM unit that synchronizes the time-of-day (TOD) clocks in multiple processors or processor sides. External Time Reference (ETR) is the z/OS generic name for the IBM Sysplex Timer (9037).

system
In SA z/OS, system means a focal point system (z/OS) or a target system (MVS, VM, VSE, LINUX, or CF).

System Automation for Integrated Operations Management
An outboard automation solution for secure remote access to mainframe/distributed systems. Tivoli System Automation for Integrated Operations Management, previously Tivoli AF/REMOTE, allows users to manage mainframe and distributed systems from any location.

The full name for SA IOM.

System Automation for OS/390
The full name for SA OS/390, the predecessor to System Automation for z/OS.

System Automation for z/OS
The full name for SA z/OS.

system console
A console, usually having a keyboard and a display screen, that is used by an operator to control and communicate with a system.

A logical device used for the operation and control of hardware functions (for
example, IPL, alter/display, and reconfiguration). The system console can be assigned to any of the physical displays attached to a processor controller or support processor.

In SA z/OS, the hardware system console for processor controllers or service processors of processors connected using SA z/OS. In the SA z/OS operator commands and configuration dialogs, SC is used to designate the system console for a target hardware processor.

**System Display and Search Facility (SDSF)**
An IBM licensed program that provides information about jobs, queues, and printers running under JES2 on a series of panels. Under SA z/OS you can select SDSF from a pull-down menu to see the resources' status, view the z/OS system log, see WTOR messages, and see active jobs on the system.

**System entry**
A construct, created with the customization dialogs, used to represent and contain policy for a system.

**System Modification Program/Extended (SMP/E)**
An IBM licensed program that facilitates the process of installing and servicing an z/OS system.

**system operations**
The part of SA z/OS that monitors and controls system operations applications and subsystems such as NetView, SDSF, JES, RMF, TSO, RODM, ACF/VTAM, CICS, IMS, and OPC. Also known as SysOps.

**system services control point (SSCP)**
In SNA, the focal point within an SNA network for managing the configuration, coordinating network operator and problem determination requests, and providing directory support and other session services for end users of the network. Multiple SSCPs, cooperating as peers, can divide the network into domains of control, with each SSCP having a hierarchical control relationship to the physical units and logical units within its domain.

**System/390 microprocessor cluster**
A configuration that consists of central processor complexes (CPCS) and may have one or more integrated coupling facilities.

**Systems Network Architecture (SNA)**
The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks.

**T**
**TAF**  See terminal access facility.

**target**
A processor or system monitored and controlled by a focal-point system.

**target control task**
In SA z/OS, target control tasks process commands and send data to target systems and workstations through communications tasks. A target control task (a NetView autotask) is assigned to a target system when the target system is initialized.

**target hardware**
In SA z/OS, the physical hardware on which a target system runs. It can be a single-image or physically partitioned processor. Contrast with target system.

**target system**
In a distributed system environment, a system that is monitored and controlled by the focal-point system. Multiple target systems can be controlled by a single focal-point system.

In SA z/OS, a computer system attached to the focal-point system for monitoring and control. The definition of a target system includes how remote sessions are established, what hardware is used, and what operating system is used.

**task**
A basic unit of work to be accomplished by a computer.

In the NetView environment, an operator station task (logged-on operator), automation operator (autotask), application task, or user task. A NetView task performs work in the NetView environment. All SA z/OS tasks are NetView tasks. See also message monitor task and target control task.
telecommunication line
Any physical medium, such as a wire or microwave beam, that is used to transmit data.

terminal access facility (TAF)
A NetView function that allows you to log onto multiple applications either on your system or other systems. You can define TAF sessions in the SA z/OS customization panels so you don't have to set them up each time you want to use them.

In NetView, a facility that allows a network operator to control a number of subsystems. In a full-screen or operator control session, operators can control any combination of subsystems simultaneously.

terminal emulation
The capability of a microcomputer or personal computer to operate as if it were a particular type of terminal linked to a processing unit to access data.

threshold
A value that determines the point at which SA z/OS automation performs a predefined action. See alert threshold, warning threshold, and error threshold.

time of day (TOD)
Typically refers to the time-of-day clock.

Time Sharing Option (TSO)
An optional configuration of the operating system that provides conversational timesharing from remote stations. It is an interactive service on z/OS, MVS/ESA, and MVS/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.

TSO/E
See Time-Sharing Option/Extended.

TWS
See Tivoli Workload Scheduler.

U
UCB  See unit control block

unit control block (UCB)
A control block in common storage that describes the characteristics of a particular I/O device on the operating system and that is used for allocating devices and controlling I/O operations.

unsolicited message
An SA z/OS message that is not a direct response to a command.

uniform resource identifier (URI)
A uniform resource identifier is a string of characters used to identify a name of a web resource. Such identification enables interaction with representations of the web resource over the internet, using specific protocols.

user task
An application of the NetView program defined in a NetView TASK definition statement.

Using An RMF Monitor III definition. Jobs getting service from hardware resources (processors or devices) are using these resources. The use of a resource by an address space can vary from 0% to 100% where 0% indicates no use during a Range period, and 100% indicates that the address space was found using the resource in every sample during that period.

V

view In the NetView Graphic Monitor Facility, a graphical picture of a network or part of a network. A view consists of nodes connected by links and may also include text and background lines. A view can be displayed, edited, and monitored for status information about network resources.

Virtual Server
A logical construct that appears to comprise processor, memory, and I/O resources conforming to a particular architecture. A virtual server can support an operating system, associated middleware, and applications. A hypervisor creates and manages virtual servers.

Virtual Server Collection
A set of virtual servers that supports a workload. This set is not necessarily static. The constituents of the collection at any given point are determined by virtual servers involved in supporting the workload at that time.

virtual Server Image
A package containing metadata that describes the system requirements, virtual storage drives, and any goals and constraints for the virtual machine (for example, isolation and availability). The Open Virtual Machine Format (OVF) is a Distributed Management Task Force (DMTF) standard that describes a packaging format for virtual server images.

Virtual Server Image Capture
The ability to store metadata and disk images of an existing virtual server. The metadata describes the virtual server storage, network needs, goals and constraints. The captured information is stored as a virtual server image that can be referenced and used to create and deploy other similar images.

Virtual Server Image Clone
The ability to create an identical copy (clone) of a virtual server image that can be used to create a new similar virtual server.

Virtual Storage Extended (VSE)
A system that consists of a basic operating system (VSE/Advanced Functions), and any IBM supplied and user-written programs required to meet the data processing needs of a user. VSE and the hardware that it controls form a complete computing system. Its current version is called VSE/ESA.

Virtual Telecommunications Access Method (VTAM)
An IBM licensed program that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability. Its full name is Advanced Communications Function for the Virtual Telecommunications Access Method. Synonymous with ACF/VTAM.

VM Second Level Systems Support
With this function, Processor Operations is able to control VM second level
systems (VM guest systems) in the same way that it controls systems running on real hardware.

VM/ESA
Virtual Machine/Enterprise Systems Architecture. Its current version is called z/VM.

volume
A direct access storage device (DASD) volume or a tape volume that serves a system in an SA z/OS enterprise.

VSE See Virtual Storage Extended

VTAM See Virtual Telecommunications Access Method

W
warning threshold
An application or volume service value that determines the level at which SA z/OS changes the associated icon in the graphical interface to the warning color. See alert threshold.

workstation
In SA z/OS workstation means the graphic workstation that an operator uses for day-to-day operations.

write-to-operator (WTO)
A request to send a message to an operator at the z/OS operator console. This request is made by an application and is handled by the WTO processor, which is part of the z/OS supervisor program.

write-to-operator-with-reply (WTOR)
A request to send a message to an operator at the z/OS operator console that requires a response from the operator. This request is made by an application and is handled by the WTO processor, which is part of the z/OS supervisor program.

WTO See write-to-operator

WTOR See write-to-operator-with-reply

WWV The US National Institute of Standards and Technology (NIST) radio station that provides standard time information. A second station, known as WWVB, provides standard time information at a different frequency.

X
XCF See cross-system coupling facility

XCF couple data set
The name for the sysplex couple data set prior to MVS/ESA System Product Version 5 Release 1. See also sysplex couple data set.

XCF group
A set of related members that a multisystem application defines to XCF. A member is a specific function, or instance, of the application. A member resides on one system and can communicate with other members of the same group across the sysplex.

XRF See extended recovery facility

Z
z/OS An IBM mainframe operating system that uses 64-bit real storage. See also Base Control Program.

z/OS component
A part of z/OS that performs a specific z/OS function. In SA z/OS, component refers to entities that are managed by SA z/OS automation.

z/OS subsystem
Software products that augment the z/OS operating system. JES and TSO/E are examples of z/OS subsystems. SA z/OS includes automation for some z/OS subsystems.

z/OS system
A z/OS image together with its associated hardware, which collectively are often referred to simply as a system, or z/OS system.

z196 See IBM Enterprise 196 (z196)

zAAP See IBM System z Application Assist Processor (zAAP)

zBX See IBM zEnterprise BladeCenter Extension (zBX)

zBX blade See IBM zEnterprise BladeCenter Extension (zBX) blade

zCPC The physical collection of main storage, central processors, timers, and channels within a zEnterprise mainframe. Although this collection of hardware resources is part of the larger zEnterprise central...
processor complex, you can apply energy
management policies to zCPC that are
different from those that you apply to any
attached IBM zEnterprise BladeCenter
Extension (zBX) or blades. See also central
processor complex.

zIIP  See IBM System z Integrated Information
Processor (zIIP).

zEnterprise  See IBM zEnterprise System (zEnterprise)
Index

A
ABENDING automation status 48
accessibility xi
accessing status display facility (SDF) 134
ACTIVE automation status 48
adaptive scheduling 24
additional BCP internal interfaces 253
advanced automation options
application monitor 61
alerts
monitoring 131
alternate CDS
allocating 153
spare volumes 153
switch to primary CDS 155
turn into primary CDS 153
anchors 206, 207
lost&found 209
AOCTRACE command 239, 248
AOF767I message and AOFEXI01 initialization exit 225
AOFARCAT 66
AOFCMD 66
AOFEXI01 initialization exit and AOF767I message 225
AOFKAH 66
AOFKMINORCHECK 43
AOFOPF* samples 66
AOFRDSAR 66
AOFRDSEV 66
AOFSESxx 66
AOFITWS 66
AOFWRKxx 66
AOFWRKxx operators 68
application thresholds for 177
application group
nature 110
working with 110
application group information 87
application group, automating 23
application move 26, 112
resource aware, example 38
resource aware, introduced 37
resource aware, overview 37
application move, resource aware 143
AUTBASE 66
AUTCICS 66
AUTCMON 66
AUTCON 66
AUTE2 66
AUTEVT1 66
AUTEVT2 66
AUTGSS 66
AUTHB 66
AUTHW1 66
AUTIMS 66
AUTJES 66
AUTKAH 66
AUTLOG 66
AUTMON 66
AUTMSG 66
AUTNETI, AUTNET2 66
AUTO1 66
AUTO2 66
AUTDOWN automation status 48
automated function name 66
Automatic Restart Manager statuses 56
Automatic Restart Manager status 56
automating application group 23
automation
goal driven 22
automation agent 11, 70
automation agent flags 40, 108
automation agent status 48
changing 116
automation agent view with DISPSTAT 88
automation configuration file identifying 76
automation control file 6
Automation flag 40
termination 39
Automation agent 40
automation agent flags 40
determining which are on 108
displaying 109
global automation flag 41
Initstart 40
minor resource flags 43
override settings 109
Recover 40
resource status 42
Restart 40
setting 42
Shutdown 40
Terminate 40
automation initialization options 225
automation manager 11
diagnostic information 71
interaction 115
making it primary 70
operation mode 115
retrieve information from 11
shutting down 71
starting 69
stopping 70
stopping with different options 70
trace services for 238
automation manager and automation agents cooperation 21
automation manager commands
INCMOVE 114
INGSTR 159
automation manager configuration file 6
identifying 76
automation manager messages
log 240
automation manager status 11
automation operators 65
automation operators policy
definition 66
automation policy 6
refreshing 75
automation status 43, 46
automation table 76
automation, setting on or off 109
AUTOPCE 66
AUTOPCP 66
AUTOPCR 66
AUTOTEC 66
AUTOTERM automation status 48
AUTPLEX 66
AUTPLEX2 66
AUTPLEX3 66
AUTPOST 66
AUTPOSTS 66
AUTRDSAR 66
AUTRDSEV 66
AUTREC 66
AUTRPC 66
AUTSESxx 66
AUTSHUT 66
AUTSYS 66
AUTTWS 66
AUTWRKxx 66
AUTXCF 66
AUTXCF2 66

B
backup
disruptive 33
non-disruptive 32
server battery 35
BASEOPER 66
basic mode 6
BCP internal interface
checking interlocks 254
recovery actions 253
terminating after recovery actions 254
BCP Internal Interface
First Failure Data Capture support 249
tracing and debugging 247
tracing specific connections 248
using AOCTRACE 248
TCP/IP internal interfaces
additional recovery information 253
Blade Extension(zBX)
monitoring and controlling 171
BREAKING automation status 48
BROKEN automation status 48

C
central processor 6
CFRM policy
  pending changes 151
  switching 155
changing automation agent status 116
changing policy for resources 112
channel
  types 8
  channel paths 196
checking BCP internal interface
  interlocks 254
checking resource status conditions with
  status display facility 136
CICSMSTR 66
clocks
  time of day 4
colors
in DISPSTAT 133
in status display facility (SDF) 133
command
  dialogs 82
  issuing 81
  Command Dialogs Main Menu 82
  Command Profile Editor 81
commands
  AOCTRACE 239
  DISPGW 178
  DISPSTAT 88
  INGLKUP 177
  INGMOVE 114
  INGNTFY 121
  INGPLEX SDUMP 157
  INGPLEX SLIP 159
  INGPLEX SVC DUMP 158
  INGPLEX SYTEM, WLM-related 143
  INGRIPT 91
  INGSEND 178
  INGSSTR 159
  INGTHRES 174
  ISQCMON 120
  ISQTRACE 165
  ISQVARS 129
  ISQXDS 120, 129
  ISQXMON 120
  issuing via NetView Management Console 203
  logging 82
  RMTCMD 178
  ROUTE 178
  SETSTATE 48, 116
  SETTIMER 115
status display facility (SDF) 131
commands and requests 22
  commands from NMC 212
  communicating with other systems 178
  communications path 6
  monitoring status 131
  compound status 45, 47
  condition relationship 18
  shutdown condition 24
  startup condition 24
conflicting requests
  introduced 20
  propagation of 20
  connections, tracing specific 248
connectivity changes 196
control units 6
coupled automation manager and
  automation agents 21
couple data set (CDS)
  alternate CDS
  allocating 153
  spare volumes 153
  turn into primary CDS 153
couple data sets 153
coupling facilities 6
  channels 8
  CPC activation profiles 169
  CPE 81
  creating a timer 115
  critical threshold 26, 174
  setting to 1 176
cross-domain session 179
  CTLDOWN automation status 48, 50
  customizing status display facility (SDF) 132
  displaying (continued) 132
  exceptional messages 119
  flag override settings 109
  interested operator list 120
  OMEGAMON sessions 125
  outstanding WTOs 119
  statistical information with
  INGRPT 91
  status with DISPSTAT 88
  status with status display facility (SDF) 131
work item statistics 241
  DIPSMTR 87
  DIPNTFY (Display Notification Operator Settings) 121
  DIPSS HD command dialog 109
  DIPSS FLT 89
  DISPSTAT 88, 108
  DSPTREE 90
  DSPTRG 107, 184
  disruptive restoration 33
  DLA 172
  DOWN automation status 48
  DSIOPF member 65
duplexing
  system-managed 151
  user-managed 151
dynamic switches 6
dynamic takeover file 72
D
DASD 6
debugging 237
  BCP Internal Interface 247
  First Failure Data Capture
  support 249
  ProcOps 247
  deleting a timer 115
  deleting status display facility records 137
dependency information 90
  dependency relationships 18, 20
dependent resource 18
desired status 43, 46
devices 6, 196
DIAGINFO 71
  diagnostic functions 240
  diagnostic information for the automation manager 71
dialog
  processor operations 161
disability xi
disabling recovery process 176
discovery Library Adapter
  configuring 174
  discovery scope 173
  IdML book, transferring 174
  INGCTRL command 174
  INGDLA command 173
  introduction 172
  overview 173
  process 173
discovery process 173
  DISPAFG 87
  DISPSAF 174
  DISPSERRS 174
  DISPEVT 107
  DISPEVTS 107
  DISPSFLGS 109
  DISPINFO 90
displaying
  additional OMEGAMON session information 127
  automation flags 109
  exception views 216, 220
  exception messages displaying 119
  EXTSTART automation status 48
E
E2EOPER 66
E2EOPR 66
enabling
  resource aware application move 39
ENDED automation status 48
ENDING automation status 48
terprise systems connection (ESCON) 6
error thresholds 174
  introducing 26
  set or change 175
errors
  display occurred 174
  ESCON
  convertor 6
  director 6
  switch 6
event
  concept 24
  defined 24
event information
  DISPEVT 107
  DISPEVTS 107
  INGEVENT 107
events (NMC) 207
events and conditions 24
  EVTOPER 66
exception views 216, 220
exceptional messages displaying 119
EXTSTART automation status 48
F
FALLBACK automation status 48, 51
fiber channel connection (FICON) 6
FICON
convertor 6
director 6
switch 6
filters
DISPSFLT 89
INGFILT 86
focal point monitoring 132
focal point processor 6
functional names for automation operators 66
functions of INNGROUP 111
G
GAT&DOMID 66
gateway sessions
determining status of 182
monitoring with status display facility (SDF) 131
overview 182
GATOPER 66
GDG 75
global automation flag 41
goal driven automation 11, 22
goal, defined 11
goals
creation of 11
group concepts 27
group relationships 28
group types 27
grouping
support 23
groups 27
groups and resources 23
GSSOPER 66
SHUTDOWN 68
STARTUP 68
guest machines
processor operations support 160
guest systems hierarchy 161
guests summary panel ISQETRGG 164
H
HALFDOWN automation status 48
HALTED automation status 48
Handling jobs that are not controlled by SA z/OS 177
hardware configuration overview 4
hardware monitoring 122
HAS PARENT relationship 18
HASPASSIVEPARENT relationship 18
HBOPER 66
HSBLV 66
health performance
monitoring 123
health status 43, 48
heartbeat 207
heartbeat function 212
hierarchy
guest systems 161
preferences 34
HSACTWR 238
HSAJPTRC 239
HWOPER 66
I/O operations 4
connectivity changes 196
introduction 195
starting and stopping 73
using trace services 249
I/O operations commands
using the z/OS operator console 198
tutorial 235
I/O operations ISPF dialog 198, 227
help
I/O resources 196
image 6
IMSMSTR 66
INGAMS 115, 241
diagnostic option 240
INGAMS REFRESH 75
INGAUTO 109
INGAUTO command and automation flags 42
INGDLA command 173
INGDLG 227
INGEO 73
INGEVENT 24, 107, 184
INGEXEC command 179
INGFILT 86
INGGROUP
application move 112
functions 111
introducing 111
options 112
rolling recycle 30
INGHIST 96
display a log 240
INGINFO 87, 184
INGLIST 85, 184
INGLIST command, using in problem solving 184
INGLKUP 177
INGMOVE command 114
INGNTFY 121
INGNTFY command 121
INGNTFY command dialog 121
INGPLEX 142
INGPLEX CDS 153
INGPLEX CF 146
INGPLEX command
SDUMP 157
SLIP 159
SVCDUMP 158
SYSTEM, WLM-related data 143
INGPLEX CONsole 144
INGPLEX Dump 157
INGPLEX IPL 156
INGPLEX SDUMP 157
INGPLEX SVCDUMP 158
INGPLEX SYSTEM 143
INGPOS T 206
INGRELS 87
INGREQ
command 22
INGREQ (continued)
request 22
INGREQ command, using 94
INGERPT 91
INGRCMD 215
INGRUN 99
INGRXQRY command 244
INGRYAMD command 246
INGSCHE D 105, 184
INGSEND command 179
INGSESS command 125
INCGSET 41
INGSET command 22
INGSTR command 159
INGTHRES 174
INGTHRES command 174
INGTOP0 211
INGTRIG 107
INGVOTE 95, 184
initialization exit, AOFEXi01, and AOF767I message 225
Initialize automation flag 40
initializing SA z/OS 69
integrated system channel 8
interest operator list 119
adding your ID 120
displaying 120
join or leave 120
interlocks, BCP internal interface, checking 254
ISPF dialog 198, 227
ISQ00I message 119
ISQ01I message 119
ISQCMON 120
ISQEHARP
target hardware summary panel 162
ISQEPNV C
PATH detail panel 165
ISQESUM
status summary panel 161
ISQETARG
target system summary panel 164
ISQETRGG
guests summary panel 164
ISQSTART 73
ISQSTOP 73
ISQTRACE command 165
ISQVARS command
change status
target system 129
ISQXDST 129
ISQXDST command 120
ISQXMON 120
issuing commands 81
J
JESOPER 66
K
keyboard xi

Index  289
lock request 22
locked votes and requests 22
logging commands 82
logical partition 6
LOGOPER 66
lost&found anchors 209
LPAR management functions
CPC activation profiles 169
CPC and LPAR information queries 169
enabling 169
for BCP internal interface 170
enabling for processor operations 169
functional overview 168
introduced 168
LPAR management 168
usage considerations 170
LPAR mode 6
LU 6.2
routing commands 178
Main Menu 82
major resources on the NMC 206
management functions, LPAR 168
mapping statuses
automation agent to automation manager, 57
marked and unmarked resources 221
message AOF767I and AOFEXI01 initialization exit 225
Message Capturing 116
message classes 120
messages
interested operator list 119
monitoring 119
notification operators 120
minor resource flags 43
minor resources 43
thresholds for 177
monitoring on the NMC 206
modifying a timer 115
monitor information 87
monitor resources
using the NCCF panels 124
using the NMC 124
monitoring
communications paths 131
exceptional messages 119
hardware 122
health performance 123
messages 119
outstanding WTORs 119
processors 128
with status display facility (SDF) 131
workitem queue 246
monitoring alerts 151
Monitoring for IPL Completion 98
MONOPER 66
move functions 29
move groups and recovery mode 35
MOVED automation status 48
moving resources 112
MSGOPER 66
MVSCONSi 66
names for processor operations
objects 218
names for system operations objects 217
nature of application groups
BASIC 110
MOVE 110
SERVER 110
NCCF panels
using with monitor resources 124
NETOPER 66
NetView commands
LIST ASSIGN=GROUP 120
RMTCMD 178
ROUTE 178
logging on 69
RMTCMD routing 178
starting 72
terminating after recovery actions 254
NetView automation table
identifying 76
NetView connection path 6
SNMP 131
NetView Management Console 4, 203
creating views 216
introduction 203
issuing commands 81, 212
marked and unmarked resources 221
minor resources 221
monitoring 205
names for processor operations objects 218
names for system operations objects 217
problem management 220
processor operations 219
properties of objects 209
starting 204
network views 216
NMC 4, 203
creating views 216
introduction 203
issuing commands 212
marked and unmarked resources 221
minor resources 221
monitoring 205
multiple focal point support 204
names for system operations objects 217, 218
problem management 220
processor operations 219
properties of objects 209
starting 204
using with monitor resources 124
NMC workstation 6
non-disruptive restoration 33
notification operators 120
adding 121
deleting 122
settings 121
notify messages 121
NVC (NetView connection) 6
observed status 43, 44
observed statuses 48
OCF (operations command facility) 6
OCF-based processor 6
OMEGAMON
INGSESS command 125
session management, operator interaction 125
sessions, displaying 125
sessions, displaying additional information 127
sessions, recovering 127
sessions, starting 126
sessions, stopping 126
OPACMRD 66
OPCAMSTR 66
OPCAOPR2 66
operation mode 115
operator console 198
OPERATORS policy item 66
order, defined 11
OSA (open system adapter) 6
outstanding WTORs
displaying 119
override resource 106
schedule 106
override settings, displaying 109
overrides 11
overriding automation flags
INGAUTO command 42
overriding schedules 184
overview of SA z/OS 3
PAM 11
panels
INGTHRES 174, 175
INGTHRES (Set Thresholds) 174
Target Hardware Summary 123
parallel sysplex 6
PARMLIB 39
partitioning
logical 6
physical 6
PATH detail panel ISQEPNVC 165
pending policy changes 151
physically partitioned configuration 6
PLEXOPER 66
PLEXOPR2 66
PLEXOPR3 66
policy 6
port
ESCON port 6
FICON port 6
POSTOPER 66
POSTSLV 66
PPIOPER 66
preference scenarios 32
preference values 30, 31
preference values (continued)
  effective 30
preferences
  real hierarchy 34
  primary automation manager 11, 70, 71
priority of requests
  FORCE 11
  HIGH 11
  LOW 11
problem determination 237
problem management on the NMC 220
problem solving scenario 183
processor 6
  monitoring status 128
  OCF-based 6
  target hardware summary 123
processor operations 4
  guest machines support 160
  starting 73
  stopping 73
processor operations commands
  ISQTRACE 165
processor operations dialog 161
processor operations on the NMC 219
processor operations status
  information 122
processors 10
ProcOps
  First Failure Data Capture
    support 249
  tracing and debugging 247
  tracing specific connections 248
  using AOCTRACE 248
propagation
  requests 17
requests, details 17
propagation of conflicting requests 20
propagation of requests 18
properties of objects on NMC 209
PSM console mode 165
PSM control program
  restarting 168
  starting 166
  stopping 167
R
rebuild 151
system-managed 146, 151
user-managed 146, 151
RECOVER 66
recovering OMEGAMON sessions 127
recovery actions
  BCP internal interface 253
  checking BCP internal interface
    interlocks 254
  terminating NetView 254
  terminating the BCP internal
    interface 254
Recovery automation flag 40
recovery mode and move groups 35
recovery process, disabling 176
refreshing automation policy 75
refreshing the automation manager
  configuration file 75
regular monitoring 61
regular resources 205
relationship
  HASPARENT 18
  HASPARENT/FAKE PARENT 18
  request propagation 18
  start 17
  stop 17
relationship information 87
relationships
  concept 18
  dependency 18
relationships concept 20
repair, takeover file 72
request 22
  calculation of schedules 24
  defined 11
  INGREQ 22
  priority 22
  propagation 17
  propagation, details 17
  using INGVOTE 95
request propagation 18
requests
  conflicting 20
  creation of 11
  hierarchy 11
  locked 22
requests and goals 17
resource
  dependent 18
  depending 17
  supporting 17, 18
resource aware application move 143
  enabling 39
  example 38
  introduced 37
  overview 37
resource information
  sysplexwide 85
resource override 105, 106
resource overrides 11
resource states
  automation flags 42
  description 42
  list 42
  status display facility 43
  resource status conditions, checking with
    status display facility 136
  resource subcategory 86
resources
  start and stop 94
  resources and groups 23
  resources, stopping 184
Restart automation flag 40
RESTART automation status 48, 52
restarting the PSM control program 168
restoration
  disruptive 33
  non-disruptive 33
resuming a timer 115
resynchronization 63
RMTCMD command 178
RODM (resource object data manager) 6
rolling recycle 30
ROUTE command 178
routing
  commands 178
RPCOPER 66
Runmodes
  runmodes, setting 104
  subsystems 99
RUNNING automation status 48
Runtokens 101
SA
z/OS
  overview 3
  starting 69
SA z/OS components
  I/O operations 4
    processor operations 4
    system operations 4
SA z/OS main menu 82
SA z/OS topology manager
  behavior 211
  initialization 211
SA z/OS topology manager
  overview 203
SAM 11
scenario, problem solving 183
schedule 11, 105
  adaptive 24
  automatic calculation of 24
  schedule override 105, 106, 184
  schedule overrides 11
SDF 131
second level systems
  VM support 160
secondary automation manager 11
server management 26
service period 105
  introduced 24
  service period overrides 11
service window 24
session management
  OMEGAMON, INGSESS 125
  OMEGAMON, operator
    interaction 125
sessions
  displaying OMEGAMON 125
  OMEGAMON, displaying additional
    information 127
  recovering OMEGAMON 127
  starting OMEGAMON 126
  stopping OMEGAMON 126
SETSTATE 116
SETSTATE command 48, 116
SETTIMER 115
setting automation flags
  INGAUTO command 42
  introduction 42
setting automation on or off 109
setting up thresholds 176
setting up, work item lifecycle
  recording 243
shortcut keys xi
SHUTDOWN
  GSSOPER 68
Shutdown automation flag 40
shut down condition 24
SHUTOPER 66
single point of control 178, 179
single system image 178, 179
SLIP TRAP 159

Index 291
specific connections, tracing 248
start resources 94
startability status 43, 47
STARTED automation status 48
STARTED2 automation status 48, 53
starting
automation manager 69
I/O operations 73
NetView 72
processor operations 73
PSM control program 166
SA z/OS 69
subsystems 95
system operations 72
starting OMEGAMON sessions 126
STARTUP
GSSOPER 68
startup condition 24
statistics, work item 241
status
ABENDING 48
ACTIVE 48
application monitor status 56
AUTODOWN 48
Automatic Restart Manager 56
automation agent status 48
AUTOTERM 48
BREAKING 48
BROKEN 48
CTLDOWN 48, 50
determination 61
displaying with DISPSTAT 88
displaying with status display facility (SDF) 131
DOWN 48
ENDED 48
ENDING 48
FALLBACK 48, 51
HALFDOWN 48
HALTED 48
MOVED 48
observed status 48
processor operations
attention status 129
path status 131
remote system attention status 129
remote system status 129
processors 128
RESTART 48, 52
RUNNING 48
STARTED 48
STARTED2 48, 53
STOPPED 48
STopping 48
STUCK 48
UP 48
ZOMBIE 48
status display facility
resource states 43
status display facility (SDF) 131
accessing 134
checking resource status conditions 136
colors 133
commands 131
customizing 132
status display facility (SDF) (continued)
deleting records 137
delayed point monitoring 132
how it works 133
monitoring with 131
using 134
status forwarding path failure 237
status information, processor operations 122
status of a resource
automation status 43
compound status 43
desired status 43
observed status 43
startability status 43
status of resources 43
status panels 88
status summary 128
status summary panel ISQESUM 161
statuses
mapping automation agent to observed 57
statuses supplied by the automation agent
application monitor status 48
Automatic Restart Manager status 48
automation agent status 48
stop resources 94
STOPPED automation status 48
stopping automation manager 70
I/O operations 73
processor operations 73
PSM control program 167
resources 184
STOPPING automation status 48
stopping OMEGAMON sessions 126
structure
duplexing
system-managed 151
user-managed 151
pending policy changes 151
rebuild 151
system-managed 146, 151
user-managed 146, 151
STUCK automation status 48
subtype 86
supporting resource 18
suspending a timer 115
switch 196
switch matrix 230
switches 6
switching
alternate CDS to primary CDS 155
CFRM policy 155
SYSOPER 66
sysplex 6
monitor and control 142
sysplex automation 142
sysplex timer 6
sysplexwide resource information 85
system operations 4
starting 72
system operations commands
INGMOVE 114
INGSTR 159
system operations control files
automation control file 6
automation manager configuration file 6
System Operations main menu 82
system-managed duplexing 151
system-managed rebuild 146, 151
systems
monitoring status 129
T
TAF (Terminal Access Facility)
fullscreen sessions 180
panels 181
TAF full screen display 181
takeover file
dynamic 72
repair 72
target hardware
status 129
target hardware summary 123
target hardware summary panel ISQEHARP 162
target processors 6
monitoring status 128
target system
status 129
status panels
interested operator list 120
overall status 128
status summary 128
summary status 129
target system summary 129
target system summary panel ISQETARG 164
target systems
monitoring messages 119
monitoring status 129
TECOPER 66
terminal access facility fullscreen sessions 180
Terminate automation flag 40
threshold
critical 26
critical, setting to 1 176
for applications 177
for minor resources 177
for z/OS components 26, 174
MVSCOMP 176
setting up 176
threshold settings 174
thresholds 174
set or change 175
timer
creating 115
deleting 115
modifying 115
NetView 115
resuming 115
suspending 115
TIMERS command 115
Tivoli Enterprise Portal 6
issuing commands 81
using 124
tokens 206, 209
trace records, printing 239