Defining Automation Policy

Version 2 Release 3
Defining Automation Policy

Version 2 Release 3
Note!
Before using this information and the product it supports, be sure to read the general information under “Notices” on page xiii.

Ninth Edition (July 2004)

This edition applies to Tivoli System Automation for z/OS (Program Number 5645-006) Version 2, Release 3, an IBM licensed program, and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters.

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

IBM welcomes your comments. A form for readers’ comments appears at the back of this publication. If the form has been removed, address your comments to:
IBM Deutschland Entwicklung GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of 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.

© Copyright International Business Machines Corporation 1996, 2004. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
## Contents

### Figures

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>vii</td>
</tr>
</tbody>
</table>

### Tables

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>xi</td>
</tr>
</tbody>
</table>

### Notices

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>xiii</td>
</tr>
</tbody>
</table>

- Web Site Disclaimer .................................................. xiii
- Programming Interface Information ................................. xiii
- Trademarks ......................................................................... xiv

### About This Book

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>xv</td>
</tr>
</tbody>
</table>

- Who Should Use This Book ............................................. xv
- Notes on Terminology .................................................... xv
- Where to Find More Information ....................................... xv
  - The System Automation for z/OS Library ............................ xv
  - Related Product Information ............................................ xvi
  - Using LookAt to look up message explanations ................... xviii
  - Accessing z/OS licensed documents on the Internet .............. xviii

### Chapter 1. Overview

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

- How to Define Automation Policy: An Outline .................. 2
  - The Customization Dialog Primary Menu ......................... 2
  - Entry Type Introduction ............................................. 3
  - Policy Objects and Policy Items ................................... 7
  - Considerations for the SA z/OS Graphical Interface ........... 8
  - Grouping Resources Logically ....................................... 8
  - Considerations for Automation ...................................... 9
  - Building and Distributing SA z/OS Automation Policy ....... 10

### Chapter 2. Using the Customization Dialog

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
</tr>
</tbody>
</table>

- How to Start the Customization Dialog ......................... 12
- How to Navigate in the Customization Dialog .................. 13
  - How to Use a Fast Path ............................................. 13
  - Concurrent User Access .............................................. 14
  - How to Cancel Input on a Panel ................................... 14
  - How to Create New Policy Objects ............................... 14
  - How to Rename a Policy Object .................................... 15
  - Delete Option on All Entry Name Selection Panels ......... 16
  - Common Policy Items ................................................ 18
  - Policy Item DESCRIPTION ............................................ 18
  - Policy Item COPY ....................................................... 19
  - Policy Item WHERE USED ............................................. 20

### Chapter 3. Automation Concepts

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
</tr>
</tbody>
</table>

- Goal Driven Automation .............................................. 23
  - Grouping Support .................................................... 23
  - Automatic Calculation of Schedules and Service Periods .... 24
  - Adaptive Scheduling ................................................ 28
  - Relationship Support ................................................. 28
  - How SA z/OS Uses Error Thresholds ............................... 31
  - Setting up a Threshold ............................................... 31
  - Automation Flags .................................................... 34
  - Types of Automation Flags ......................................... 34
  - Setting Automation Flags ........................................... 35
  - Automation Manager Flags ........................................... 37
  - Relationship Between Automation Flags and Resource Status .. 38
  - Automation Flags and Minor Resource Flags .................. 39
  - Event and Trigger Support ......................................... 39
  - Assigning System Automation Symbols (AOCCLONE) .......... 42
  - Using System Symbols and System Automation Symbols .... 43

### Chapter 4. Scenarios on How to Use the Customization Dialog

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>45</td>
</tr>
</tbody>
</table>

- How to Work with Resources ........................................ 45
  - Step 1: How to Create the Policy Objects that Make Up the Sysplex ................................. 45
  - Step 2: How to Create the Applications ........................................ 47
  - Step 3: How to Create the Application Groups .................. 48
  - How to Use Schedules (Service Periods) .......................... 69
  - Schedule Override .................................................... 69
  - Early Start ............................................................. 70
  - How to Use Triggers and Events .................................... 70
  - How to Use User E-T Pairs .......................................... 72
  - Step 1: Specifying User Data in the Customization Dialog ...... 72
  - Step 2: Access the User Data ....................................... 73
  - How to Implement Message Processing with User-Defined Data ........................................ 74
  - Step 1: How to Specify the User-Defined Data ................. 74
  - Step 2: How to Specify the Commands ............................. 75
  - Step 3: How to Update the NetView Automation Table ....... 76
  - Step 4: How to Write the REXX Exec .............................. 78

### Chapter 5. Defining Automation Policy

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>81</td>
</tr>
</tbody>
</table>

- Enterprise Policy Object ............................................ 82
  - Policy Items for an Enterprise .................................... 82
  - More about Policy Item SEND COMMAND OPERS ................. 83
  - More about Policy Item INQSEND PARMS ......................... 85
  - More about Policy Item PROCESSOR OPS INFO .................. 86
  - More about Policy Item AUTO MSG CLASSES .................... 87
  - More about Policy Item SYMBOL DESCRIPTION ................ 88
  - More about Policy Item MPF DEFINITIONS ....................... 89
  - Group Policy Object ................................................ 90
  - Creating a New Group ................................................. 90
  - Policy Items for Standard Groups ................................ 92
  - Policy Items for Sysplex Groups ................................ 92
  - More about Policy Item GROUP INFO ............................. 94
  - More about Policy Item SUBGROUPS ............................... 94
  - More about Policy Item SYSTEMS ................................. 95
  - More about Policy Item SYSplex ................................. 95

© Copyright IBM Corp. 1996, 2004
Chapter 6. Policy Database Maintenance

Policy Database Selection List

Policy Items for JES3 Subsystems
Policy Items for CICS Subsystems
Policy Items for DB2 Subsystems
Policy Items for IMS Subsystems
Policy Items for OPC Subsystems
Policy Items for USS Subsystems
Defining VTAM Application Subsystems
Events Policy Object
Creating a New Event
Policy Items for Events
More about Policy Item CONDITION
Service Periods Policy Object
More about Policy Item SERVICE WINDOW
Triggers Policy Object
More about Policy Item CONDITION
Processor Policy Object
Creating a New Processor
Policy Items for Processors
Policy Items for PSM Processors
Monitor Resource Policy Object
More about Policy Item MONITOR INFO
More about Policy Item HEALTHSTATE
More about Policy Item SERVICE PERIOD
Product Automation Policy Object
Defining Automation for DB2 Components
Defining Automation for CICS Components
Defining Automation for IMS Components
Defining Automation for OPC Components
Timers Policy Object
Timeout Settings Policy Object
Tape Attendance Policy Object
MVS Component Policy Object
More about the THRESHOLD Policy Items
More about the WTOBUF RECOVERY Policy Item
More about Policy Item MESSAGES/USER DATA
More about Policy Item MINOR RESOURCE FLAGS
MVSCOMP Defaults Policy Object
System Defaults Policy Object
More about Policy Item AUTOMATION OPTIONS
More about Policy Item ENVIRONMENT
Application Defaults Policy Object
Auto Operators Policy Object
More about the OPERATORS Policy Item
Structure of AOFWRKxx Operators
Notify Operators Policy Object
Network Policy Object
NNT Sessions Policy Object
Resident CLISTS Policy Object
Status Details Policy Object
More about Policy Item SDF DETAILS
Includes Policy Object
User E-T Pairs Policy Object
More about Policy Item E-T DATA

More about Policy Item NMC DEFINITIONS
More about Policy Item APPLICATION GROUPS
More about Policy Item LOCAL PAGE DATA
More about Policy Item JOB/ASID DEFINITIONS
More about Policy Item RESOURCE DEFINITIONS
More about Policy Item IEMACXX SYMBOLS
SubGroup Policy Object
Creating a New SubGroup
Policy Items for SubGroups
System Policy Object
Creating a New System
Policy Items for Systems
More about Policy Item SYSTEM INFO
More about Policy Item AUTOMATION SYMBOLS
More about Policy Item NETVIEW CONSOLE
More about Policy Item AUTOMATION SETUP
More about Policy Item PROCESSOR
More about Policy Item TARGET SYSTEM INFO
More about Policy Item IPL INFO
ApplicationGroup Policy Object
Creating a New ApplicationGroup
Policy Items for ApplicationGroups
More about Policy Group APPLGROUP INFO
More about Policy Item APPLICATIONS
More about Policy Item TRIGGER
More about Policy Item SERVICE PERIOD
More about Policy Item RELATIONSHIPS
More about Policy Item RESOURCES
More about Policy Item GENERATED RESOURCES
More about Policy Item MEMBER OF
Application Policy Object
Creating a New Application
Policy Items for Applications
More about Policy Item APPLICATION INFO
More about Policy Item AUTOMATION INFO
More about Policy Item AUTOMATION FLAGS
More about Policy Item SERVICE PERIOD
More about Policy Item RELATIONSHIPS
More about Policy Item MESSAGES/USER DATA
More about Policy Item STARTUP
More about Policy Item SHUTDOWN
More about Policy Item THRESHOLDS
More about Policy Item MINOR RESOURCE
FLAGS
More about Policy Item SYSTEM ASSOCIATION
More about Policy Item GENERATED RESOURCES
More about Policy Item MEMBER OF for Applications
Policy Items for JES2 Subsystems
Figures

1. Customization Dialog Primary Menu ................................................. 2
2. Entry Type Selection Panel ............................................................. 7
3. Entry Name Selection Panel for Groups ........................................... 7
4. Policy Selection Panel for Sysplex Groups ....................................... 8
5. Customization Dialog Primary Menu ............................................... 12
6. Define New Entry Panel for an Application ....................................... 15
7. Sample Generic Define New Entry Panel .......................................... 15
8. Entry Name Selection Panel: Rename Action ..................................... 16
9. Entry Name Rename Popup Panel ..................................................... 16
10. Delete Error Panel ........................................................................ 17
11. Where Used Panel .......................................................................... 17
12. Requesting a List of Instances ......................................................... 18
13. Links of Application Class to Instances ........................................... 18
14. Sample Description Policy Item Panel ............................................. 19
15. Where Used Panel .......................................................................... 21
16. Example of Aggregating Applications into Application Groups .......... 24
17. Application Groups ....................................................................... 26
18. Grouping of Applications into Nested Application Groups .................. 27
19. Relationships and Conditions ............................................................ 29
20. The HASPARENT Relationship ........................................................ 29
21. Relationship in a Sysplex ................................................................. 30
22. Relationship between Applications and Application Groups ............. 31
23. Thresholds Specification Panel ........................................................ 32
24. Thresholds Definition Panel ............................................................ 33
25. Component Definition: MVS DUMP Panel ....................................... 34
26. Flag Automation Specification Panel ................................................. 35
27. How a Trigger Uses Events .............................................................. 40
28. Triggers, Events and Conditions ...................................................... 41
29. Current Configuration with Sysplex Application Group CICSAOR ........ 51
30. Current Configuration with Applications CICSAOR1 through CICSAOR3 52
31. Resources for the CICSAOR Sysplex Application Group ..................... 53
32. Current Configuration with Application Group DB2 .......................... 55
33. Current Configuration with Applications Linked to DB2 ..................... 56
34. Select System Panel for DB2 Application group ................................ 56
35. Select Resources Panel for DB2 Application Group ........................... 57
36. Grouping Resources ..................................................................... 58
37. Resources for the CICS Sysplex Application Group .......................... 58
38. Current Configuration with Applications Linked to DB2 ..................... 59
39. Select Resources Panel for Application Group NET ........................... 59
40. Select Resources Panel for DWHAPG with valid APG-Resources ........ 61
41. Select Resources Panel for DWHAPG2 ........................................... 62
42. Select Resources Panel for DWHAPG3 ........................................... 63
43. Select Resources Panel for DWHAPG with Valid APG-Resources ........ 64
44. Select Resources Panel for DWHAPG with Selected Preferences ........ 66
45. Select Resources Panel for DWHAPG with Converted Preferences ........ 66
46. Define New Entry Panel for WBSRVAPG ........................................ 68
47. Final Select Resources Panel for WBSRVAPG .................................. 69
48. Scenario with Triggers, Events and Service Period .......................... 72
49. Operator Profile for Sending Commands Panel .................................. 84
50. UET Entry-Type Selection Panel ..................................................... 72
51. UET Keyword-Data Specification Panel .......................................... 73
52. Message Processing Panel ............................................................... 74
53. User Defined Data Panel ................................................................. 75
54. CMD Processing Panel ................................................................. 76
55. Message Processing Panel ............................................................... 76
56. Automation Processing Panel ......................................................... 77
57. Automation Processing Panel ......................................................... 77
58. Entry Type Selection Panel ............................................................. 81
59. Policy Selection Panel for Enterprises .............................................. 82
60. SMM Header and Trailer Definition Panel ....................................... 89
61. Define New Entry Panel for a Sysplex Group ................................. 91
63. Policy Selection Panel for Sysplex Groups ....................................... 93
64. Group Information Panel for Standard Groups ................................. 94
65. Subgroups Panel for Standard Groups ............................................. 95
66. Systems Panel for Standard Groups ................................................ 95
67. Sysplex Policy Definition Panel for Sysplex Groups .......................... 96
68. NMC Definitions Panel for Sysplex Groups .................................... 98
69. Group Information Panel for Standard Groups .................................. 99
70. Local Page Data Set Recovery ....................................................... 100
71. Job Definitions for Local Page Data Set Recovery ............................ 101
72. JOB/ASID Definitions ............................................................... 101
73. Long Running ENQ Resource Definitions ....................................... 102
74. Long Running ENQ IEADMCxx Symbols ....................................... 103
75. Define New Entry Panel for a Subgroup ......................................... 104
76. Policy Selection Panel for Subgroups .............................................. 104
77. Define New Entry Panel for a System ............................................. 105
78. Policy Selection Panel for Systems ............................................... 109
79. SYSTEM INFO Panel ................................................................. 113
80. Automation Symbols Panel ........................................................... 114
81. NetView Information Panel ........................................................... 115
82. Environment Definition: CONSOLE (Route Codes) Panel ............... 116
83. Environment Setup Panel ............................................................. 117
84. Select Target Hardware for System Panel ....................................... 120
85. Target System Information Panel ................................................... 121
86. MVS Target System IPL Information Panel ..................................... 123

© Copyright IBM Corp. 1996, 2004
197. OPC Special Resource Definitions Panel 255
198. Policy Selection Panel for OPC Workstation Domain 256
199. Policy Selection Panel for Timers 257
200. Timer Specification Panel 257
201. Policy Selection Panel for Timeout Settings 258
202. Environment Definition: TIMEOUT Panel 259
203. Policy Selection Panel for Tape Attendance 261
204. Environment Definition: TAPES Panel 261
205. Policy Selection Panel for MVS Components 263
206. Environment Definition: WTOBUF Overflow Panel 265
207. WTOBUF Recovery Passes Panel for MVS Component 266
208. Console Mode After WTOBUF Recovery Panel 267
209. Message Processing Panel 268
210. Minor Resource Selection Panel 269
211. Policy Selection Panel for MVSCOMP Defaults 270
212. Policy Selection Panel for System Defaults 271
213. System Automation Options Panel 272
214. Subsystem Defaults Panel 272
215. Policy Selection Panel for Application Defaults 273
216. Policy Selection Panel for Auto Operators 274
217. Automation Operator Definition Panel 276
218. Automation Operator: MVSCONS Panel 277
219. Automation Operator: MVSCONS Panel 278
220. Policy Selection Panel for Notify Operators 279
221. Operator Notification Panel 280
222. Operator Notification Message Classes Panel 281
223. Policy Selection Panel for Network 283
224. Policy Selection Panel for NNT Sessions 287
225. NNT Sessions Definition Panel 287
226. NNT Sessions Definition Panel 287
227. Policy Selection Panel for Resident CLISTS 289
228. Policy Selection Panel for Status Details 290
229. Status Display Facility Details Panel 290
230. Policy Selection Panel for User E-T Pairs 293
231. UET Entry-Type Selection Panel 293
232. UET Keyword-Data Specification Panel 294
233. Policy Data Base Selection Panel 295
234. Policy Data Base Selection Panel 297
235. Create a New Policy Database Panel 298
### Tables

1. System Automation for z/OS Library  
2. Related Products Books  
3. System Symbols and System Automation Symbols  
4. System Symbols  
5. System Automation Symbols  
6. Relationship Conditions  
7. Preference Value Scenarios  
8. SA z/OS Mapping of NetView Automation Operator to SA z/OS Automated Function  
9. SA z/OS Message Classes  
10. Creation of Entry Names  
11. Cross Reference of SA OS/390 1.3 Feature Keywords to SA z/OS 2.3 Dialog Entries  
12. Policy Selection Filter Attributes

© Copyright IBM Corp. 1996, 2004
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 Entwicklung GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany

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

Web Site Disclaimer

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

Programming Interface Information

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

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

ACF/VTAM       APPN
CICS            DB2
DFSM/MVS        ESCON
FICON           IBM
IMS             Library Reader
Micro Channel   WebSphere MQ
Multiprise      MVS
MVS/ESA         Netfinity
Netfinity Manager NetView
OpenEdition     OS/2
OS/390          Parallel Sysplex
RACF            RMF
Sysplex Timer   System/390
Tivoli          Tivoli Enterprise Console
VM/ESA          VSE/ESA
VTAM            z/OS
zSeries

The following terms are trademarks of other companies:

HealthKit
    Heath Co., St. Joseph, MO
Spectracom
    Spectracom Corp., Rochester, NY
Windows
    Microsoft Corporation
Windows NT
    Microsoft Corporation
Java
    Sun Microsystems, Inc.
UNIX
    UNIX is a registered trademark of The Open Group in the United States and other countries.
LINUX
    LINUX is a registered trademark of Linus Torvalds and others.
About This Book

This book describes Tivoli® System Automation for z/OS® (SA z/OS). This product plays a key role in supplying high-end automation solutions whereas Tivoli Systems (Tivoli Management Environment, Tivoli) delivers integrated cross-platform management functions. This book contains information about how to enable SA z/OS’s automation.

Who Should Use This Book

This book is primarily intended for automation programmers responsible for defining and maintaining the automation policy.

Notes on Terminology

**MVS™**

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

**NetView®**

The term NetView used in this documentation stands for IBM® Tivoli NetView for OS/390®.

**OPC**

References in this book to OPC refer to Tivoli Workload Scheduler (TWS).

**Resources, Applications, and Subsystems**

These three terms are used in the SA z/OS library to designate automated entities. The correct term to be used depends on the context. Refer to the Introduction chapter in `System Automation for z/OS Operator’s Commands` for the definition of these terms.

Where to Find More Information

**The System Automation for z/OS Library**

The following table shows the information units in the System Automation for z/OS library:
Table 1. System Automation for z/OS Library

<table>
<thead>
<tr>
<th>Title</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Automation for z/OS Planning and Installation</td>
<td>SC33-7038</td>
</tr>
<tr>
<td>System Automation for z/OS Customizing and Programming</td>
<td>SC33-7035</td>
</tr>
<tr>
<td>System Automation for z/OS Defining Automation Policy</td>
<td>SC33-7039</td>
</tr>
<tr>
<td>System Automation for z/OS User’s Guide</td>
<td>SC33-7040</td>
</tr>
<tr>
<td>System Automation for z/OS Messages and Codes</td>
<td>SC33-7041</td>
</tr>
<tr>
<td>System Automation for z/OS Operator’s Commands</td>
<td>SC33-7042</td>
</tr>
<tr>
<td>System Automation for z/OS Licensed Program Specifications</td>
<td>SC33-7037</td>
</tr>
</tbody>
</table>

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

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

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

Related Product Information
The following table shows the books in the related product libraries that you may find useful for support of the SA z/OS base program.

Table 2. Related Products Books

<table>
<thead>
<tr>
<th>Title</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISPF User’s Guide</td>
<td>SC34-4484</td>
</tr>
<tr>
<td>ISPF Dialog Management Guide and Reference</td>
<td>SC34-4266</td>
</tr>
<tr>
<td>MVS/ESA MVS Configuration Program Guide and Reference</td>
<td>GC28-1817</td>
</tr>
<tr>
<td>MVS/ESA Planning: Dynamic I/O Configuration</td>
<td>GC28-1674</td>
</tr>
<tr>
<td>MVS/ESA Support for the Enterprise Systems Connection</td>
<td>GC28-1140</td>
</tr>
<tr>
<td>MVS/ESA Planning: APPC Management</td>
<td>GC28-1110</td>
</tr>
<tr>
<td>MVS/ESA Application Development Macro Reference</td>
<td>GC28-1822</td>
</tr>
<tr>
<td>OS/390: MVS System Commands</td>
<td>GC28-1781</td>
</tr>
<tr>
<td>MVS/ESA SPL Application Development Macro Reference</td>
<td>GC28-1857</td>
</tr>
<tr>
<td>OS/390 Hardware Configuration Definition: User’s Guide</td>
<td>SC28-1848</td>
</tr>
<tr>
<td>OS/390 Information Roadmap</td>
<td>GC28-1727</td>
</tr>
<tr>
<td>OS/390 Information Transformation</td>
<td>GC28-1985</td>
</tr>
<tr>
<td>OS/390 Introduction and Release Guide</td>
<td>GC28-1725</td>
</tr>
</tbody>
</table>
Table 2. Related Products Books (continued)

<table>
<thead>
<tr>
<th>Title</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS/390 JES Commands Summary</td>
<td>GX22-0041</td>
</tr>
<tr>
<td>OS/390 Licensed Program Specifications</td>
<td>GC28-1728</td>
</tr>
<tr>
<td>OS/390 Printing Softcopy Books</td>
<td>S544-5354</td>
</tr>
<tr>
<td>OS/390 Starting Up a Sysplex</td>
<td>GC28-1779</td>
</tr>
<tr>
<td>OS/390 Up and Running!</td>
<td>GC28-1726</td>
</tr>
<tr>
<td>Planning for the 9032 Model 3 and 9033 Enterprise Systems Connection</td>
<td>SA26-6100</td>
</tr>
<tr>
<td>S/390 MVS Sysplex Overview -- An Introduction to Data Sharing and</td>
<td>GC23-1208</td>
</tr>
<tr>
<td>Parallelism</td>
<td></td>
</tr>
<tr>
<td>S/390 MVS Sysplex Systems Management</td>
<td>GC23-1209</td>
</tr>
<tr>
<td>S/390 Sysplex Hardware and Software Migration</td>
<td>GC23-1210</td>
</tr>
<tr>
<td>S/390 MVS Sysplex Application Migration</td>
<td>GC23-1211</td>
</tr>
<tr>
<td>S/390 Managing Your Processors</td>
<td>GC38-0452</td>
</tr>
<tr>
<td>Tivoli/Enterprise Console User's Guide Volume I</td>
<td>GC31-8334</td>
</tr>
<tr>
<td>Tivoli/Enterprise Console User's Guide Volume II</td>
<td>GC31-8335</td>
</tr>
<tr>
<td>Tivoli/Enterprise Console Event Integration Facility Guide</td>
<td>GC31-8337</td>
</tr>
<tr>
<td>Tivoli NetView for OS/390 Administration Reference</td>
<td>SC31-8222</td>
</tr>
<tr>
<td>Tivoli NetView for OS/390 Application Programming Guide</td>
<td>SC31-8223</td>
</tr>
<tr>
<td>Tivoli NetView for OS/390 APPN Topology and Accounting Agent</td>
<td>SC31-8224</td>
</tr>
<tr>
<td>Tivoli NetView for OS/390 Automation Guide</td>
<td>SC31-8225</td>
</tr>
<tr>
<td>Tivoli NetView for OS/390 AON Customization Guide</td>
<td>SC31-8662</td>
</tr>
<tr>
<td>Tivoli NetView for OS/390 AON User's Guide</td>
<td>GC31-8661</td>
</tr>
<tr>
<td>Tivoli NetView for OS/390 Bridge Implementation</td>
<td>SC31-8238</td>
</tr>
<tr>
<td>Tivoli NetView for OS/390 Command Reference Vol. 1</td>
<td>SC31-8227</td>
</tr>
<tr>
<td>Tivoli NetView for OS/390 Command Reference Vol. 2</td>
<td>SC31-8735</td>
</tr>
<tr>
<td>Tivoli NetView for OS/390 Customization Guide</td>
<td>SC31-8228</td>
</tr>
<tr>
<td>Tivoli NetView for OS/390 Customization: Using Assembler</td>
<td>SC31-8229</td>
</tr>
<tr>
<td>Tivoli NetView for OS/390 Customization: Using Pipes</td>
<td>SC31-8248</td>
</tr>
<tr>
<td>Tivoli NetView for OS/390 Customization: Using PL/I and C</td>
<td>SC31-8230</td>
</tr>
<tr>
<td>Tivoli NetView for OS/390 Customization: Using REXX and CLIST Language</td>
<td>SC31-8231</td>
</tr>
<tr>
<td>Tivoli NetView for OS/390 Data Mode Reference</td>
<td>SC31-8232</td>
</tr>
<tr>
<td>Tivoli NetView for OS/390 Installation: Getting Started</td>
<td>SC31-8767</td>
</tr>
<tr>
<td>Tivoli NetView for OS/390 Installation: Migration Guide</td>
<td>SC31-8768</td>
</tr>
<tr>
<td>Tivoli NetView for OS/390 Installation: Configuring Graphical</td>
<td>SC31-8770</td>
</tr>
<tr>
<td>Components</td>
<td></td>
</tr>
<tr>
<td>Tivoli NetView for OS/390 Installation: Configuring Additional</td>
<td>SC31-8769</td>
</tr>
<tr>
<td>Components</td>
<td></td>
</tr>
<tr>
<td>Tivoli NetView for OS/390 Messages and Codes</td>
<td>SC31-8237</td>
</tr>
<tr>
<td>Tivoli NetView for OS/390 MultiSystem Manager User’s Guide</td>
<td>SC31-8607</td>
</tr>
<tr>
<td>Title</td>
<td>Order Number</td>
</tr>
<tr>
<td>--------------------------------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Tivoli NetView for OS/390 NetView Management Console User’s Guide</td>
<td>GC31-8665</td>
</tr>
<tr>
<td>Tivoli NetView for OS/390 User’s Guide</td>
<td>SC31-8241</td>
</tr>
<tr>
<td>Tivoli NetView for OS/390 RODM and GMFHS Programming Guide</td>
<td>SC31-8233</td>
</tr>
<tr>
<td>Tivoli NetView for OS/390 Security Reference</td>
<td>SC31-8606</td>
</tr>
<tr>
<td>Tivoli NetView for OS/390 SNA Topology Manager and APPN Accounting Manager Implementation Guide</td>
<td>SC31-8239</td>
</tr>
<tr>
<td>Tivoli Management Platform Reference Guide</td>
<td>GC31-8324</td>
</tr>
<tr>
<td>TSO/E REXX/MVS Reference</td>
<td>SC28-1883</td>
</tr>
<tr>
<td>VM/XA SP GCS Command and Macro Reference</td>
<td>SC23-0433</td>
</tr>
<tr>
<td>VSE/SP Unattended Node Support</td>
<td>SC33-6412</td>
</tr>
<tr>
<td>VTAM Messages and Codes</td>
<td>SC31-6493</td>
</tr>
<tr>
<td>VTAM Network Implementation Guide</td>
<td>SC31-6404</td>
</tr>
<tr>
<td>VTAM Network Implementation Guide</td>
<td>SC31-6434</td>
</tr>
</tbody>
</table>

**Using LookAt to look up message explanations**

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

You can access LookAt from the Internet at: [http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/](http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/) or from anywhere in z/OS or z/OS.e where you can access a TSO/E command line (for example, TSO/E prompt, ISPF, z/OS UNIX System Services running OMVS).

The LookAt Web site also features a mobile edition of LookAt for devices such as Pocket PCs, Palm OS, or Linux-based handhelds. So, if you have a handheld device with wireless access and an Internet browser, you can now access LookAt message information from almost anywhere.

To use LookAt as a TSO/E command, you must have LookAt installed on your host system. You can obtain the LookAt code for TSO/E from a disk on your z/OS Collection (SK3T-4269) or from the LookAt Web site’s **Download** link.

**Accessing z/OS licensed documents on the Internet**

z/OS licensed documentation is available on the Internet in PDF format at the IBM Resource Link™ Web site at: [http://www.ibm.com/servers/resourcelink](http://www.ibm.com/servers/resourcelink)

Licensed documents are available only to customers with a z/OS license. Access to these documents requires an IBM Resource Link user ID and password, and a key code. With your z/OS order you received a Memo to Licensees, (GI10-0671), that includes this key code.  

---

1. **z/OS.e™** customers received a Memo to Licensees, (GI10-0684) that includes this key code.
To obtain your IBM Resource Link user ID and password, log on to:
http://www.ibm.com/servers/resourcelink

To register for access to the z/OS licensed documents:
1. Sign in to Resource Link using your Resource Link user ID and password.
2. Select User Profiles located on the left-hand navigation bar.

Note: You cannot access the z/OS licensed documents unless you have registered for access to them and received an e-mail confirmation informing you that your request has been processed.

Printed licensed documents are not available from IBM.

You can use the PDF format on either z/OS Licensed Product Library CD-ROM or IBM Resource Link to print licensed documents.
Chapter 1. Overview

Before you can use SA z/OS to automate, monitor, and control the resources in your enterprise, you must define your enterprise’s automation policy. For this purpose you use the SA z/OS customization dialog to provide information to SA z/OS such as:

- Which resources (such as systems or applications) you want to automate, monitor and control
- How resources are to be associated (grouped) with each other for automation
- The dependencies between resources and groups of resources
- How and what automation, such as automatic startup or shutdown, is to be applied to these resources
- Conditions under which automated actions should occur, and what actions should be performed
- When automation is active, and how it handles certain tasks and events during system operation

The customization dialog leads the SA z/OS policy administrator through the process of defining resources and automation for the enterprise. SA z/OS automatically extracts the required information from this to display resources graphically on the NMC workstation.

Before you begin to use the customization dialog to define your enterprise automation policy, you need to determine how you want SA z/OS to work within your enterprise. Making this determination is part of the process of planning to use SA z/OS. There are many considerations for planning to use SA z/OS of which you should be aware. For this purpose, also refer to the planning instructions given in System Automation for z/OS Planning and Installation

Note: Do not use the customization dialogs on a Kanji terminal as they will not work. If you have your terminal defined as a 3278 KN terminal, you can circumvent the Kanji restriction by going into ISPF option 0.1 (panel ID ISPOPT1) and redefining the TERMINAL TYPE to 3278.

The remainder of this chapter deals with the following topics:

- “How to Define Automation Policy: An Outline” on page 2 gives an outline of the process of creating enterprise automation policy.
- “The Customization Dialog Primary Menu” on page 2 introduces the Customization Dialog Primary Menu.
- “Entry Type Introduction” on page 3 introduces all entry types.
- “Considerations for the SA z/OS Graphical Interface” on page 8 provides some information on how to customize the monitoring aspect of SA z/OS.
- “Considerations for Automation” on page 9 provides some information on how to customize the automation aspect of SA z/OS.
- “Building and Distributing SA z/OS Automation Policy” on page 10 introduces what you need to do once you have entered your policy information (both for monitoring and automation) into the policy database using the customization dialog.
How to Define Automation Policy: An Outline

The process of defining automation policy comprises the following steps:

1. At the configuration focal point, where you have the customization dialog installed, create the information in the policy database using the ISPF input panels of the customization dialog.

2. Then use the BUILD function of the customization dialog to produce the required output, the system operations configuration files: Automation Control File (ACF), Automation Manager Configuration File (AMC), NetView Automation Table (AT), and the MPFLSTSA file.

3. Then you distribute the generated output to the sysplexes or standalone systems that you want to automate.

Refer to Chapter 8, “Building and Distributing Control Files,” on page 313 and Figure 248 on page 329 for more information on building and distributing the generated output files.

The Customization Dialog Primary Menu

Once you have started the customization dialog, you are presented with the Customization Dialog Primary Menu, as shown in Figure 1.

```
MENU OPTIONS HELP
-----------------------------------------------
AOFGRM SA z/OS 2.3 Customization Dialog Primary Menu
Option ===>

0 Settings User parameters
1 Open Work with the Policy Data Base
2 Build Build functions for Policy Data Base
3 Report Generate reports from Policy Data Base
4 Policies Maintain Policy Data Base list
5 Data Management Import policies/Migrate files into a Policy Data Base
U User User-defined selections
X Exit Terminate Customization Dialog

To switch to another Policy Data Base, specify the Policy Data Base name in the following field, or specify a ? to get a selection list.
Current Policy Data Base .. RESOURCE_PDB

Licensed Materials - Property of IBM
5645-006 (C) Copyright IBM Corp. 1990, 2004 All Rights Reserved.
```

Figure 1. Customization Dialog Primary Menu

You can select the following options:

0 Settings
  This option displays the Customization Dialog Settings Menu, see ‘Changing Your Customization Dialog Environment’ on page 369.

1 Open
  With this option, you open the policy database of your choice, see ‘How to Start the Customization Dialog’ on page 12.

2 Build
  This option displays the Build Functions Menu, see Chapter 8, “Building and Distributing Control Files,” on page 313.
Starting the Customization Dialog

3 Report
This option displays the Report Selection Menu, see Chapter 7, “Creating Reports about Policy Databases,” on page 307.

4 Policies
This option displays the Policy Data Base Selection panel with the list of existing policy databases, see “Creating a New Policy Database” on page 297.

5 Data Management
This option displays the Data Management Select panel, see “Migrating the Automation Control File” on page 329.

U User
This option invokes the User-written Functions Selection Menu. This menu allows you to add installation-specific functions to the customization dialog. For more information refer to “User-written Functions” on page 377.

X Exit Use this option to exit the customization dialog

Entry Type Introduction

All resources in your enterprise, such as systems or applications, or other objects that you use to automate your enterprise data processing, such as components of your z/OS operating system, are called entry types. A complete list of SA z/OS entry types is listed on the Entry Type Selection panel (see Figure 2 on page 7), which is the first panel you see after you have decided which enterprise or policy database you are going to work with.

For each entry type you can invoke a series of panels to create, modify, or delete entries of a certain entry type. A single instance of a certain entry type is called a policy object. These entries are referred to by their entry name.

When you select an object via its name, you get a list of policy items specific for the entry type you are working with. For example, the Timers entry type may consist of several named sets of timer definitions, each containing a number of timers. These named sets are selected and linked to a particular system as required. Each set can be linked to an arbitrary number of systems and any number of sets can be linked to any particular system.

The numbers and letters in parentheses in the following list indicate the entry type’s index on the Entry Type Selection panel. You can use either numbers or letters to select the appropriate entry type. The alphanumeric index is used in other panels of the customization dialog to identify a certain entry type.

Detailed information and scenarios on how to automate the resources (applications and application groups) and make them available on certain systems are described in Chapter 4, “Scenarios on How to Use the Customization Dialog,” on page 45.

(1/ENT) Enterprise
An Enterprise is the top level of your SA z/OS structure. It comprises enterprise wide data for the complete policy database, including:
- the definition of an operator profile for sending commands
- the definition of INGSEND command parameters
- Processor operations focal point information
Entry Type Introduction

(2/GRP) Group
A group is the largest subdivision of an SA z/OS enterprise. An SA z/OS enterprise may have several groups, each consisting of systems or subgroups, or both.

A sysplex group can be used to represent a physical sysplex (systems within the same XCF group ID). A standard group can be used to represent a logical sysplex (systems with different XCF group IDs). The definition of groups is a prerequisite for sysplex-wide automation.

(3/SBG) Subgroup
Like groups, subgroups are collections of systems. You define a subgroup by selecting the member systems and assigning a name to the subgroup. Subgroups can be collected into groups. They offer a finer level of structuring groups.

(4/SYS) System
A system is an image of an operating system running on a host processor monitored by SA z/OS. The system may be MVS, VM, VSE, LINUX, or CF (coupling facility). A system definition includes for example:

- links to other objects associated with the system, for example, links to application groups
- Application defaults
- System defaults
- Clone specifications

(5/APG) Application Group
An application group is a set of applications to which you assign a name. You can define two types of application groups: System Application Groups and Sysplex Application Groups. You assign System Application Groups to one or more systems and Sysplex Application Groups to one sysplex group.

(6/APL) Application
An application is a z/OS subsystem or job that runs on a system in the SA z/OS enterprise. SA z/OS monitors applications through the messages they issue to the system, MVS commands, and SA z/OS-provided services. An application can belong to more than one application group. An application definition includes:

- Subsystem name and the name of the procedure that runs the job
- Warning and alert thresholds
- Start and Stop specifications
- individual automation flags

(7/EVT) Events
Events are used for external dependencies where they represent a process that the automation cannot handle. For example, an operator needs to manually edit a file before the application can continue. SA z/OS remembers which events are SET (have occurred) and can be told when to forget them (by defining an UNSET condition).

(8/SVP) Service Periods
Define the availability of applications. A service period is a set of time intervals (service windows) during which an application should be active or inactive.

(9/TRG) Triggers
In combination with events and service periods, triggers are used to control the starting and stopping of applications.
(10/PRO) Processor
A processor is the physical or physically partitioned hardware on which a system runs. To monitor VM guest systems, a "logical" processor, called the ProcOps Service Machine (PSM), has been introduced.

The OCF-based processors supported by SA z/OS are zSeries® and 390-CMOS processors, excluding the 9221 processor. An OCF-based processor is a processor that uses an operations command facility (OCF) to interact with human operators or external programs to perform operations management functions on the processor. The OCF accepts and processes operations management commands, which SA z/OS sends from the focal point system. The processor operations commands are described in System Automation for z/OS Operator's Commands.

(11/MTR) Monitor Resource
The Monitor Resource Policy Object enables you to monitor the performance and health of an application. That is, a separate status informs you about the application's health. This health status can be used by the automation manager to make decisions and, if necessary, trigger automation for the subject application. This monitoring can be used to make the currently available JES3 monitoring obsolete.

(20/PRD) Product Automation
With this policy object, you can define automation policy for the following products:
- CICS
- DB2
- IMS
- OPC

(30/TMR) Timers
Automation timer definitions. This object is used to schedule automation tasks which are periodic, or which must run at a particular time.

(31/TMO) Timeout Settings
The timeout values for responses to MVS commands and cross-domain commands. This entry type enables you to create sets of timeout definitions. Each timeout definition can specify timeout values for:
- System, NetView, or SA z/OS commands
- Cross-domain logon requests
- Your own timeout variables

If the time limit is exceeded, then it is assumed that an error condition exists and an appropriate action is taken.

(32/TPA) Tape Attendance
Specifies the time during which tape drives are either attended or unattended. This entry type also allows you to specify whether or not you want to monitor tape mounts on this system. If you do want to monitor tape mounts, you must enter warning and alert thresholds (in minutes and seconds) indicating how long a wait is allowed in mounting a tape before the response is considered unsatisfactory.

(33/MVC) MVS Component
This entry type allows you to apply automation to various z/OS data sets, resources and facilities.
(34/MDF) MVSCOMP Defaults
Sets defaults for z/OS component (entry type MVS Component) automation.

(35/SDF) System Defaults
This entry type allows you to set various defaults that are used throughout the system. The values set here may be overridden by other entry types, such as Application, Application Defaults, or MVSCOMP Defaults, or by setting appropriate values in individual applications or MVS Component objects.

(36/ADF) Application Defaults
This entry type specifies default values for automation flags and error threshold settings. These values will be used as the default for any Application object on the system to which the Application Defaults set is attached.

(37/AOP) Auto Operators
Maps automation functions to automation operators. Automation operators are automated operator tasks, or autotasks, that can respond to operating system, application, and network messages without requiring a human operator. Sets of automation operators can be linked to a system. Each automation operator takes actions similar to those of a human operator, such as monitoring the system and handling and coordinating recovery and shutdown operations.

Automation operators are assigned specific messages on which to act.

(38/NFY) Notify Operators
Defines NetView operators who receive notification messages.

(39/NTW) Network
Defines Gateway, adjacent NetView domains (downlevel only), and SDF focal points.

(40/NNT) NNT Sessions
Defines NetView to NetView sessions (user IDs, passwords, NetView domains).

Note: From SA z/OS 2.3 onwards, this entry type is no longer being supported. Use this function only to maintain systems with SA OS/390 releases up to 2.2 or for cleanup.

(41/RES) Resident CLISTs
Automation CLISTs that must be preloaded to improve performance.

(42/SCR) Status Details
Defines NetView Status Display facility (SDF) settings.

(98/ICL) Includes
Entry type ICL is no longer supported by SA z/OS Version 2. Use this function only to maintain systems with releases up to SA OS/390 1.3 and for cleanup.

(99/UET) User E-T Pairs
User-defined entry/type data for the automation control file. Used for user-defined extensions to automation policy.
Policy Objects and Policy Items

Each policy object has a number of policy items associated with it. To define policy, you enter data for the policy items of each policy object in the SA z/OS enterprise.

For example, if you choose option 2 on the Entry Type Selection panel (Figure 2) to specify Group policy, Figure 3 is displayed. (The labels, for example, AOFGETYP, AOFGENAM, are the panel IDs that are displayed after you enter the command PANELID on the command line.)

Figure 2. Entry Type Selection Panel

Figure 3 shows the Group policy objects SYSPLEX1 and SYSPLEX2.

Figure 3. Entry Name Selection Panel for Groups

If you select an existing group, the Policy Selection panel is displayed, as shown in Figure 4 on page 8. This panel gives you a list of the policy items for a group. You can enter S for SELECT next to any policy item, to display the appropriate policy input panel. Figure 4 on page 8 shows the Policy Selection panel for groups, listing the policy items for the selected policy object SYSPLEX1.
Considerations for the SA z/OS Graphical Interface

The customization dialog leads the SA z/OS policy administrator through several panels to define the systems and other automated resources of the enterprise. The resulting enterprise policy provides the information that SA z/OS uses to provide objects for the displays on the NMC workstation.

The objects visible on the NMC workstation are based on the resources that are defined in the customization dialog and therefore known to SA z/OS. The automation manager component of SA z/OS monitors the resources and their status. SA z/OS permanently puts information about the resources and their status into the focal point RODM from where it is retrieved by the NMC workstation.

Grouping Resources Logically

To provide the hierarchy required for graphical representation on the NMC workstation, you need to define your enterprise in terms of the relationships between the resources. The reason for grouping your enterprise resources within SA z/OS is to simplify the representation of the status information that can be generated by your enterprise. The decisions you make on how to group your resources have a great impact on how powerful the graphic interface on the NMC workstation is for you. Each group (of systems or applications) can be represented by a single icon in the NMC workstation displays. SA z/OS monitoring acts as a filtering mechanism for the status data generated by the components of these groups. The status of the group is determined by the statuses of its members using an algorithm defined independently for each grouping of resources in your SA z/OS enterprise.

The grouping of resources allowed by SA z/OS is flexible. For example, systems may be grouped by geographical region or use or type. You may decide to place all high priority jobs in one application group and specify that the status of the group will be “degraded” if only one of its subsidiary applications has an unsatisfactory status.
Considerations for the SA z/OS Graphical Interface

For information on how to set up the graphical interface on the NMC workstation refer to System Automation for z/OS Planning and Installation. For information on how to use this graphical interface refer to System Automation for z/OS User’s Guide.

Considerations for Automation

While information for monitoring of resources is defined at the enterprise level, automation of resources and applications is defined at the system level. Each system in an SA z/OS enterprise uses SA z/OS routines to automate operator functions for its own applications and z/OS components. The programs that automate these functions are called SA z/OS automation procedures. SA z/OS automation procedures invoke SA z/OS provided common routines and generic routines. You can customize your automation by writing your own automation procedures which can also invoke the SA z/OS provided routines.

SA z/OS provides automation procedures that are internally invoked according to the NetView automation table and that manage, for example, the following:

- WebSphere® Application Server
- SAP
- TCP/IP
- HTTP Server
- Job Entry Subsystem/2 (JES2)
- Job Entry Subsystem/3 (JES3)
- Time Sharing Option/Extensions (TSO/E)
- APPC
- ASCH
- Resource Measurement Facility (RMF)
- Virtual Telecommunication Access Method (VTAM®)
- NetView
- Transient jobs
- Non-z/OS resources
- Any other applications
- Special automation functions are also provided to manage CICS®, IMS™, DB2™, and to integrate with OPC

SA z/OS provides automation procedures that enable recovery of the following z/OS components and data sets:

- SYSLOG data sets
- LOGREC data sets
- System Management Facility (SMF) data sets
- Write-to-operator (WTO) buffers
- JES spools
- z/OS dump data sets
- Action Message Retention Facility (AMRF) buffers
- sysplex resources, for example, coupling facility

With SA z/OS you can also define policies for the automated startup and shutdown of processors and for the automated IPL and shutdown of operating systems.

The above lists represent applications and z/OS components for which SA z/OS provides automation, but you are not limited to these. You can extend automation by adding new applications, and you can create your own automation procedures.

The SA z/OS policy administrator uses the customization dialog to set the way in which enterprise resources are to be handled by SA z/OS when predefined
Considerations for Automation

conditions are met. In addition to invoking provided automation procedures, SA z/OS can be used to trigger user-written automation procedures.

Building and Distributing SA z/OS Automation Policy

After you have entered all data necessary to define your enterprise monitoring and automation policy in the policy database, you use the BUILD function of the customization dialog to build the files needed by SA z/OS. These include:

- The automation control files for the automation agent (ACF) and automation manager (AMC).
- The processor operations control file and a sample NetView operation definition file.
- NetView automation tables and the MPFLSTSA member.

The automation configuration definitions are needed if you want to use system operations automation functions. If you want to use processor operations automation functions (for example, monitoring processors), the processor operations control file must be built, and the corresponding NetView operator definitions must be provided.

Refer to Chapter 8, “Building and Distributing Control Files,” on page 313 for more information on the SA z/OS build function.

The automation configuration files must be distributed to each system in the SA z/OS enterprise where SA z/OS system operations functions are to run. In general, this is all of the z/OS systems in the SA z/OS enterprise. In a sysplex, the data set name of the automation control file must be equal on all systems in the sysplex.

However, it is highly recommended to use a shared data set in a sysplex, and if possible, to use a shared data set for the whole enterprise.
Chapter 2. Using the Customization Dialog

The SA z/OS customization dialog runs under Interactive System Productivity Facility (ISPF) Dialog Manager, which operates under Time Sharing Option/Extended (TSO/E). For information on installing the customization dialog and sample policy databases on the focal point system, refer to System Automation for z/OS Planning and Installation.

The customization dialog provides panels on which you create and work with SA z/OS automation policy definitions. The task of defining and building automation policy is performed on the SA z/OS configuration focal point system from where it is distributed to the systems and sysplexes that you want to automate (see also Figure 248 on page 325).

The combined definition of all the resources is called an enterprise definition or also an automation configuration. SA z/OS enterprise definition data entered in the customization dialog is stored in an z/OS data set called a policy database. You can define more than one enterprise, each with its own policy database, or you can store the automation policy of an enterprise in more than one policy database, however, the policy for all systems of a sysplex must be in the same policy database.

Your first step in defining automation policy will be to create a new policy database, using one of the sample databases as a model. SA z/OS provides sample policy databases containing sample enterprise definitions. These are described in "Using Sample SA z/OS Policy Databases” on page 300.

To do this, you should create an empty policy database using the *EMPTY sample database, and then use the IMPORT function to import policy definitions from either the other sample databases that are supplied with SA z/OS, or from your own policy databases. For more details about importing policy databases, see "Importing Policy Database Data” on page 327. You should note that this is now the preferred method for creating new policy databases.

You can create many different SA z/OS enterprise definitions. However, SA z/OS uses only one enterprise definition at a time. Each SA z/OS enterprise definition is stored in a different policy database. The customization dialog enables you to edit, copy, delete, and build SA z/OS enterprise definitions after you create them.

You select entry types from the list shown on the Entry Type Selection panel (Figure 2 on page 7) and certain objects from these entry types to specify the way enterprise resources are to be handled by SA z/OS when predefined conditions are met. You can link automation policies to individual systems as required to meet your automation needs.

You can monitor systems and resources using the NMC workstation that are defined to SA z/OS even if you have not defined special automation policy for these resources. Conversely, you may automate your enterprise without monitoring it using the NMC workstation. If you want to use SA z/OS for automation only you must still use the customization dialog to create policy for the resources you want to automate.

© Copyright IBM Corp. 1996, 2004
How to Start the Customization Dialog

To start the customization dialog:

1. Log on to TSO/E using a logon procedure that will make the customization dialog available to you. Refer to **System Automation for z/OS Planning and Installation** for instructions on how to create an appropriate logon procedure.

2. Select the customization dialog from the ISPF menu on which they have been installed. SA z/OS displays the **Customization Dialog Primary Menu** as shown in Figure 5.

   **Note:**
   The installation can use the keyword INITSEL for the startup exec INGDLG to automatically provide an initial selection on the **Customization Dialog Primary Menu**. If this keyword is specified, you will not see the **Customization Dialog Primary Menu** as the first panel when invoking the customization dialog, but for example, the **Entry Name Selection** panel for a frequently used entry type.

---

**Figure 5. Customization Dialog Primary Menu**

To start the customization dialog for a policy database of your choice, you can:

- Select option 1 **Open** and type the name of your desired policy database into the **Current Policy Data Base** field (if not filled in from a previous invocation). This will lead you directly to the **Entry Type Selection** panel shown in Figure 2 on page 7

- Type a ‘?’ in the **Current Policy Data Base** field to get a selection of available policy databases in the **Policy Data Base Selection** panel as shown in Figure 23 on page 297 From this panel then you select your desired policy database.

- You can also reach the **Policy Data Base Selection** panel by selecting option 4 **Policies**.

If there are no policy databases displayed when you start the customization dialog, you must either create a policy database (see “Creating a New Policy Database” on page 297) or add an existing policy database to your display list.
How to Navigate in the Customization Dialog

Many characteristics of the customization dialog are ISPF facilities. See ISPF User’s Guide for details. You need to know the following characteristics of the customization dialog:

- For panels that display a list from which you are required to select one or more entries and specify an action against that entry, you can:
  - Enter “action entry name” on the command line, for example, OPEN entry name
  - Enter the action in the Action column next to the appropriate entry, for example, enter OPEN in the Action column next to entry name.
  - Select the entry by entering S or SELECT in the Action column next to the entry. Move the cursor to the ACTIONS menu bar entry, press ENTER, and enter the number corresponding to the action you want to take. For example, place an S in the Action column next to “entry name” and select 1 (OPEN) from the Action menu.

The action most likely to be used on a certain panel is always the first action listed in the ACTIONS pull-down menu. When you enter S or SELECT in the Action column and press the ENTER key, this default action is used. For example, entering S under Action for “MY_ENTERPRISE” in the Policy Data Base Selection panel opens the MY_ENTERPRISE database.

Some panels display a list from which you can select only one entry. If you attempt to select more than one entry from such a list, only one selection is accepted. If one entry is already selected and you select another one, this new selection becomes active and the previous selection is removed.

From a list of policy items, you can select multiple items. Then they are processed sequentially. Pressing the END (PF3) key takes you to the next selected item to be processed without returning to the selection panel.

- The COMMANDS pull-down menu lists all commands that can be entered at the command line.
- The ACTIONS pull-down menu lists all actions that can be entered in the Action column for panels displaying entry lists.
- The VIEW pull-down menu lists the different levels of detail that can be displayed for the entries in an entry list. For example, in many entry lists you can display just the entry name and short description, or you can also display a detailed description for each entry.

Sample panels shown in this documentation may look different on your system depending on the selected view.

- On data entry panels, the END (PF3) key verifies input, saves the data, and exits the current panel.
- Pressing the ENTER key verifies input and redisplays the current panel.

How to Use a Fast Path

If the policy database that you want to work with, is already displayed in the field Current Policy Data Base of the Customization Dialog Primary Menu, you can use the entry type abbreviation as a fast path:

- You can directly type the code for the entry type that you want to work with. For example, to reach the Entry Name Selection panel for Applications, just type: APL
How to Navigate in the Customization Dialog

- If you want to start with defining a new application group for example, just type:
  
  APL; NEW application group_name

- From every panel within the customization dialog, if you want to reach a certain entry type and you already know the name of the object, you want to edit, you can type, for example, to reach the APPC application:
  
  =APL; S APPC

  or to reach application group CICS_APG:
  
  =APL; S CICS_APG

  or to just reach the Entry Name Selection panel for Applications:
  
  =APL

Concurrent User Access

As you may have separate administrators for different areas, for example, JES or VTAM, who should be able to do their updates independently, the customization dialog allows concurrent update for entry types APL, APG, and MTR. For other entry types, only one user at a time can access a policy database. See also Appendix A, “Concurrent Customization by Multiple Users,” on page 365 for further information.

How to Cancel Input on a Panel

To cancel your input on the current panel, type the CANCEL (or C) at the Command prompt. CANCEL can be used to disregard data entered on the panel, whether or not you have pressed the ENTER key. However, once the message POLICY SAVED

is displayed when leaving a panel, the data entered has been saved and cannot be canceled. To delete input after this message is displayed, you must go back to that panel and type over existing values.

How to Create New Policy Objects

The process of creating a new policy object is the same for all types of policy objects, as follows:

1. From the Entry Type Selection panel, you select the desired policy object’s entry type. For example, if you want to create a new Application object, enter option 6 or APL at the command line.

2. SA z/OS displays the Entry Name Selection panel for the selected entry type. You can either select action NEW from the COMMANDS pull-down or enter the command new entry_type_name at the panel’s command line. The entry type name is optional for this command.

3. SA z/OS displays the Define New Entry panel (an example for defining a new Application is shown in Figure 6 on page 15), for the selected entry type, where you now specify the required information. See Chapter 5, “Defining Automation Policy,” on page 81 for more information about how to create new policy objects.
4. If you specify an entry name and press Enter, the subsystem name and job name are defaulted to the entry name, if possible. This is also true for all other entry types where the entry name could be used as a default for further qualifications.

Throughout Chapter 5, “Defining Automation Policy,” on page 81, if there is no extra information given about creating a particular policy object, then it is created using a generic panel, as shown in Figure 7.

How to Rename a Policy Object

The process of renaming an existing policy object is the same for all types of policy object. From the Entry Name Selection panel, type r or rename in the Action field in front of the entry you want to rename, as shown in Figure 8 on page 16.
The sample screen shows a list of applications with command r in the action field of entry name ASCH. Pressing the ENTER key will display a rename popup panel where you can enter the new entry name, as shown in Figure 9.

Pressing ENTER will start the renaming process; if the name you entered already exists or is invalid you will be required to enter a different name. After completion, the newly sorted Entry Name Selection panel will be displayed with the renamed entry as the top row. If there are multiple rename requests, the last renamed entry is shown as top row.

You can leave the rename popup panel using END or CANCEL without renaming the policy object.

Notes:
- The rename function is implemented for all entry types except for Includes (ICL).
- To modify the Enterprise name, edit the policy database and then simply overtype the name, see “Specifying Settings for Policy Databases” on page 301.

Delete Option on All Entry Name Selection Panels

To delete a policy object select it from the Entry Name Selection panel, and either select Delete from the ACTIONS pull-down menu or type d in the Action column. A popup panel will ask you to confirm your deletion request.
Delete Option on All Entry Name Selection Panels

If you want to delete an object that is selected by other objects, a panel is displayed that informs you of this fact and offers you to display a list of the referencing objects and, optionally, remove the links.

**Examples:**
- An application is linked to one or more application groups.
- An application class has links to instances.
- An application group is linked to a system.

**Example 1:**

For applications that are members of application groups, the *Delete Error* panel offers the following possibility:

![Delete Error Panel](image)

**Figure 10. Delete Error Panel**

Entering YES on this panel displays the panel shown in **Figure 11** where you can remove the application to be deleted from the application groups where it is contained.

![Where Used Panel](image)

**Figure 11. Where Used Panel**

**Example 2:**
Delete Option on All Entry Name Selection Panels

For application classes that have links to applications, the Delete Error panel offers the following possibility:

```
COMMANDS       HELP
------------------------
AOFGM0C5      Delete Error

Entry Type : Application  PolicyDB Name : DATABASE_NAME
Entry Name   : APL_CLASS      Enterprise Name : YOUR_ENTERPRISE

The delete request cannot be completed because the Class selected is currently referenced by one or more Instances.

You can use the LINK TO INSTANCES function to list all the referencing Instances and, optionally, remove them.

Display list of INSTANCES. . . YES 2 YES  NO
```

Figure 12. Requesting a List of Instances

Entering YES on this panel, (2), displays the panel shown in Figure 13.

```
COMMANDS ACTIONS VIEW HELP
------------------------
AOFGX2I     Link Class to Instances

Entry Type : Application  PolicyDB Name : DATABASE_NAME
Entry Name   : APL_CLASS      Enterprise Name : YOUR_ENTERPRISE

Action     Status  Entry Name
--------     ------  -------
          SELECTED ABC_INST
          SELECTED DEF_INST
          SELECTED GHI_INST
```

Figure 13. Links of Application Class to Instances

This panel allows you to scroll through the list of all links and to remove them, if appropriate. If all links are removed, then the deletion of the application class can be continued by confirming your deletion request.

Common Policy Items

The following policy items are found on many policy definitions:

- DESCRIPTION
- COPY
- WHERE USED

Policy Item DESCRIPTION

Figure 14 on page 19 shows an example of a Description policy item panel (in this case for the SYSTEM entry type)
Figure 14. Sample Description Policy Item Panel

Both types of description (short and long) can be displayed in the customization dialog, depending on which VIEW option you choose.

Short Description
This is optional and can be up to forty characters in length. The short description is also used by the NMC workstation, DISPSTAT, and INGLIST.

Long Description
This is optional and can consist of up to five lines of up to fifty characters in length. The extended description is part of the DISPINFO display for applications.

Policy Item COPY
While creating a new policy object, you can use this policy item to copy policy information from an existing object to the created object. This function displays a list of all other entries of the same entry type in the SA z/OS enterprise that are eligible to be used as the source of a copy function.

The following list outlines the process to use an object’s Copy policy item while creating a new object.

1. Select an entry type from the Entry Type Selection panel. The Entry Name Selection panel is displayed.
2. Enter new (N) on the command line to create a new object. SA z/OS will display the Define New Entry panel for the policy object that you want create.
3. Enter information for the new object as prompted by its Define New Entry panel. Exit the panel (END command).
4. The Policy Selection panel for the new object is displayed. Select its Copy policy item.
5. A panel listing all currently defined objects from the same entry type is displayed. Select the concrete policy object from which you want to copy policy information.
6. When the copy is complete, you return to the Policy Selection panel for the new object. The Copy complete message shows that the copy has been successful. If you copy groups or processors, you may receive message partial copy complete, which means that not all entries could have been copied. The help function shows details about the data that was not copied.
Common Policy Items

7. Complete the new application definition by performing any necessary modifications to the information that has been copied.

The following list shows which information items are or are not copied by the Copy policy item:

1. All data of the source object is copied to the target object for all entry types with two exceptions:
   - Data which has to be entered during the 'NEW' processing on the related Define New Entry panel for the target object is not copied from the source object.
   - The short and extended description which is also offered on the Define New Entry panel is copied from the source object whenever there is not already data specified in the target object.

2. All links to other objects down the hierarchy of objects are copied from the source to the target. For example, if you copy an application group with linked applications, then the links of the source application group are copied to the target application group.

3. All links to other objects up the hierarchy of objects are NOT copied from the source to the target. For example, if you copy an application that is linked to an application group, then the copied target application is NOT linked to that application group.

4. Groups of type SYSPLEX can only be copied from groups of type SYSPLEX and groups of type STANDARD can only be copied from groups of type STANDARD (see "Creating a New Group" on page 90).

5. Sysplex Application Groups can only be copied from Sysplex Application Groups and System Application Groups can only be copied from System Application Groups (see "Application Group Policy Object" on page 125).

6. Links to members of sysplex groups (that is, links to subgroups, systems and Sysplex Application Groups) are NOT copied, to avoid conflicts like having the same system defined in more than one sysplex.

7. Links from systems to processors are not copied to avoid conflicts having the same system defined in more than one processor.

8. Resource links of Application Groups are not copied.

Policy Item WHERE USED

When you select this option for a certain resource, SA z/OS displays a list of all resources from a higher level which currently do include or possibly may include this resource into their hierarchy. You can add or remove the lower level resource to or from the hierarchy of the higher level resource.

For example, if you select this policy item from the Policy Selection panel for a system (entry type System), SA z/OS displays a list of all groups and subgroups in the SA z/OS enterprise, indicating which groups and subgroups include this system. You can add the system to a group or subgroup, or remove the system from a group or subgroup.

Or, if you select this policy item for an application, you get a list of existing application groups. The word SELECTED in the Status column indicates in which application group the application currently is included (see Figure 15 on page 21).

Other examples for the usage of this policy item are:
   - for Service Periods: to which applications are they linked?
**Common Policy Items**

- for *System Defaults*: which z/OS systems in the enterprise make use of this system defaults set.

```
<table>
<thead>
<tr>
<th>COMMANDS</th>
<th>ACTIONS</th>
<th>VIEW</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFGXWHU</td>
<td>Where Used</td>
<td>Row 1 to 9 of 9</td>
<td></td>
</tr>
<tr>
<td>Command ===</td>
<td>SCROLL===</td>
<td>PAGE</td>
<td></td>
</tr>
</tbody>
</table>

| Entry Type : Application PolicyDB Name : DATABASE_NAME |
| Entry Name : CICS Enterprise Name : YOUR_ENTERPRISE |

<table>
<thead>
<tr>
<th>Action</th>
<th>Status</th>
<th>Entry Name</th>
<th>Entry Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SELECTED</td>
<td>CICS_APG1</td>
<td>APG</td>
</tr>
<tr>
<td></td>
<td>SELECTED</td>
<td>CICS_APG2</td>
<td>APG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COPY_APG</td>
<td>APG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DEFAULTS</td>
<td>APG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OPERATIONS</td>
<td>APG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SAG_APPL</td>
<td>APG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYSTAPG</td>
<td>APG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYSTEM</td>
<td>APG</td>
</tr>
</tbody>
</table>

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

**Figure 15. Where Used Panel**
Common Policy Items
Chapter 3. Automation Concepts

This chapter describes various SA z/OS concepts that you should read before using the customization dialog to define automation policy for your enterprise. The following topics are discussed:

- “Goal Driven Automation”
- “Relationship Support” on page 28
- “How SA z/OS Uses Error Thresholds” on page 31
- “Automation Flags” on page 34
- “Event and Trigger Support” on page 39

Goal Driven Automation

SA z/OS implements goal driven automation.

One flavor of goal driven automation in SA z/OS are events and triggers on the one hand and service periods on the other hand (see “Event and Trigger Support” on page 39 and “Automatic Calculation of Schedules (Service Periods)” on page 28). They basically allow you to define goals for applications, application groups, or monitor resources (to be available or unavailable) 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 really 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, as 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.

You can find more details on the concepts of goal driven automation later in this chapter or you can also refer to System Automation for z/OS User’s Guide.

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 16 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 An and Bn. For example, the complex application group G2 comprises application group G1 and the single applications A4, A5. Application group G1 in turn consists of three applications A1, A2, A3. Also, application groups may comprise applications that may or may not be contained in other application groups. G3 is such an example.

A Grouping Scenario

The following scenario shows how grouping is performed on the conceptual level. For information on how to enter such a grouping policy into the customization dialog, refer to “How to Work with Resources” on page 45.

Grouping applications and nesting application groups is a process that can be logically divided into multiple steps:

- “Step 1: Identify Resources”
- “Step 2: Group the Resources” on page 25
- “Step 3: Nest the Groups” on page 26
- “Step 4: Match Resources to Systems” on page 27

Note:
Applications and application groups are also referred to with the general term resources.

Step 1: Identify Resources
Imagine a scenario with two application groups representing a data warehouse and a web server.
• The data warehouse requires the following resources: VTAM, TCP/IP, DB2 (consisting of components DB2MAIN, DB2IRLM), CICS (consisting of components CICSMAIN, CICSAOR1, CICSAOR2, CICSAOR3, CICSTOR, CICSFOR), and DWH.

• The web server requires the following resources: VTAM, TCP/IP, DB2 (DB2MAIN, DB2IRLM), and WBSRV.

**Step 2: Group the Resources**

Some of the needed resources identified in the previous step are candidates for aggregation:

• Resources that are made of multiple components; in our example, these are:
  – CICSAOR1, CICSAOR2 and CICSAOR3 which could be aggregated into a group called CICSAOR, so that they can be automated as a unit.
  – group CICS, which will contain resources CICSMAIN, CICSTOR, CICSFOR and the aggregated CICSAOR group.
  – group DB2, which will contain resources DB2MAIN and DB2IRLM.

• Resources that are used by multiple exploiters; in our example, these are:
  – VTAM and TCP/IP, which are used by the data warehouse and the web server; so we will aggregate them into a group called NET.
  – DB2MAIN and DB2IRLM, which are used by the data warehouse and the web server; so we will aggregate them into a group called DB2.

• Finally we conceive the data warehouse as group DWHAPG and the web server as group WBSRVAPG for later nesting the other groups into them as required (see "Step 3: Nest the Groups" on page 26).

Figure 17 on page 26 shows a possible aggregation of all components into groups.
In Figure 17, oval labels represent application groups, rectangle labels represent applications. For example, application group CICS contains applications CICSMAIN, CICSTOR, CICSFOR and the nested application group CICSAOR.

**Step 3: Nest the Groups**

Figure 18 on page 27 shows how the application groups are nested in this environment. From an operator’s point of view, a start request against WBSRVAPG should start all imbedded components in a reasonable order. The same is true for requests against all other groups such as, for example, NET and DB2.
Step 4: Match Resources to Systems

In a next step, system administrators must decide on which systems and in which sysplexes these groups and applications should run. Assume that in our scenario we have an environment of three z/OS systems SYSTEM1, SYSTEM2, SYSTEM3 that are members in sysplex SYSPLEX1.

In our example, a possible decision could be:

- The web server group WBSRVAPG should run on SYSTEM1.
- The data warehouse group DWHAPG should run on SYSTEM2 and - if SYSTEM2 is not available - on SYSTEM3.
- The DB2 group, containing applications DB2MAIN, DB2IRLM should run on SYSTEM2 and on SYSTEM3.
- The NET group, containing applications VTAM, TCPIP should run in the entire sysplex.
- The CICS group, containing applications CICSMAIN, CICSFOR, CICSTOR and the application group CICSAOR should run on SYSTEM2 and on SYSTEM3.

Refer to “How to Work with Resources” on page 45 for detailed information on how to implement this scenario using the customization dialog.
Automatic Calculation of Schedules (Service Periods)

Service periods allow users 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 7am to 7pm, and application B should be up from 9am to 3pm and 5pm to 9pm, then 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 7am until 9pm. Note that this inference is done dynamically at runtime and will also take into account operator requests and schedule overrides.

Adaptive Scheduling

When defining the sysplex policy, the automation programmer can specify that certain resources are backups for other resources. In order to meet and/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 which 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, as it is the selective request propagation by the group resources that achieves the effect.

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, \texttt{RES\_DEP}, would be: \texttt{MAKEAVAILABLE/WhenAvailable}, where the condition \texttt{WhenAvailable} applies to a supporting resource \texttt{RES\_SUP} (see Figure 19).

So the relationship defined in Figure 19 would read: Only if the desired state for the dependent resource \texttt{RES\_DEP} is available, make it available as soon as the supporting resource \texttt{RES\_SUP} is available.

![Figure 19. Relationships and Conditions](image)

In the example shown in Figure 19 a request to become available is issued for \texttt{RES\_DEP}. \texttt{RES\_DEP} cannot be made available, unless \texttt{RES\_SUP} is available. SA z/OS implements the concept of a request propagation. A request will be propagated along the dependency graph, so that finally the original request can be fulfilled. In this example, the request to \texttt{RES\_DEP} will be propagated to \texttt{RES\_SUP}, which can make itself available, because it is not depending on anything else. Then, after \texttt{RES\_SUP}, is available, \texttt{RES\_DEP} will become available. Thus the original request is persistent until it is withdrawn.

Another example: assume that \texttt{RES\_DEP} has its relationship with the supporting resource \texttt{RES\_SUP} defined as \texttt{HASPASSIVEPARENT} and a request to become available is issued for \texttt{RES\_DEP}. This request will not be propagated along the dependency graph. A separate request must be issued for \texttt{RES\_SUP} to make the resource \texttt{RES\_DEP} available.

There is a special relationship, \texttt{HASPARENT}, that has been introduced to simplify relationship specifications. It is equivalent to a \texttt{MAKEAVAILABLE/WhenAvailable} dependency between the dependent and supporting resource and a \texttt{MAKEUNAVAILABLE/WhenDown} dependency from the supporting resource to the dependent resource, as shown in Figure 20.

![Figure 20. The HASPARENT Relationship](image)

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 21 on page 30 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 21.

Be careful not to define loops.

Figure 21. Relationship in a Sysplex

Figure 21 shows an example of grouping and relationships defined across a sysplex consisting of the systems SYS1, SYS2 and SYS3.

In this figure, the arrows denote \texttt{HASPARENT} relationships. For example, \texttt{CICS\_TOR} will be started as soon as one member of application group \texttt{CICS\_AOR/APG} is available and VTAM is available.

If a group is a dependent resource, then all \texttt{MAKEAVAILABLE} relationships apply to the members of the group as well as to the group as a whole, for example, in Figure 22 on page 31 resource APL will not start before the resources CICS, VTAM, and JES2 are available.
For information on how to define relationships refer to “More about Policy Item RELATIONSHIPS” on page 132. During automation in your enterprise, conflicting relationships and requests could possibly occur. System Automation for z/OS User's Guide describes such a scenario and informs the operator how to handle such a situation.

How SA z/OS Uses Error Thresholds

Error thresholds influence whether SA z/OS recovers from an error situation or not. These are a number of error conditions within a certain time interval, such as 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 restarting a resource. For applications, you can define a critical threshold for restarting these applications.

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 ran 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 they may also be customized for other uses.

The following sections inform you how to define error thresholds. Information on how to display threshold settings and occurrences, and to interactively set thresholds (by using the command INGTHRES) is provided in System Automation for z/OS User’s Guide.

Setting up a Threshold

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 by selecting policy item THRESHOLDS for the following policy objects:

- System Defaults, see Figure 23 on page 32
- Application Defaults, you get a panel similar to Figure 23 on page 32
- MVSCOMP Defaults, you get a panel similar to Figure 23 on page 32
- Application, see Figure 24 on page 33
How SA z/OS Uses Error Thresholds

- MVS Component, see “Thresholds for MVS Components” on page 34

Thresholds can be displayed and modified using the INGTHRES command (see System Automation for z/OS Operator’s Commands).

To set error thresholds for all MVSCOMP resources, choose MVSCOMP Defaults from the Entry Type Selection panel. To set error thresholds for all applications, choose Application Defaults from the Entry Type Selection panel. If you do not specify error threshold defaults for MVS components or applications, SA z/OS uses the error thresholds for all monitored resources, which are defined in the System Defaults objects.

Thresholds are defined by specifying the number of errors in a particular time interval, for example three times in one hour. Figure 23 shows the Thresholds Specification panel for the System Defaults object.

Note:
Setting a critical threshold to 1 will disable the recovery process already with the first occurrence of the error. This is valid independently from the specified time interval.

<table>
<thead>
<tr>
<th>COMMANDS HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFPTH1</td>
</tr>
</tbody>
</table>
| Command ===>
| Entry Type : System Defaults  | PolicyDB Name : DATABASE_NAME |
| Entry Name : SYSTEM_DEFAULTS  | Enterprise Name : YOUR_ENTERPRISE |

Resource: System wide defaults
Specify the number of times an event must occur to define a particular level.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Critical Number</th>
<th>Critical Interval (hh:mm)</th>
<th>Frequent Number</th>
<th>Frequent Interval (hh:mm)</th>
<th>Infrequent Number</th>
<th>Infrequent Interval (hh:mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>System wide defaults</td>
<td>4</td>
<td>01:00</td>
<td>3</td>
<td>02:00</td>
<td>2</td>
<td>12:00</td>
</tr>
</tbody>
</table>

Figure 23. Thresholds Specification Panel

There are three levels of threshold defined:
- Infrequent: Issues a warning message and restart processing is done
- Frequent: Issues a warning message, but restart processing is still done
- Critical: Sets the resource to a status of BROKEN, stops the restart automation and issues a message

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 to recover 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, an application has a critical error threshold of two per hour defined.
• 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.
• 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 will display a panel as shown in Figure 24.

If no error thresholds are set here, the application uses the thresholds 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.

<table>
<thead>
<tr>
<th>COMMANDS HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFPIZTH Thresholds Definition</td>
</tr>
</tbody>
</table>
| Command ===>

| Entry Type : Application | PolicyDB Name : DATABASE_NAME |
| Entry Name : APPC | Enterprise Name : YOUR_ENTERPRISE |

| Resource: APPC |
| Description: APPC/MVS automation policy |

Specify the number of times an event must occur to define a particular level.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Critical Number</th>
<th>Frequent Number</th>
<th>Infrequent Number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Interval (hh:mm)</td>
<td>Interval (hh:mm)</td>
<td>Interval (hh:mm)</td>
</tr>
<tr>
<td>APPC</td>
<td>4 01:00</td>
<td>3 02:00</td>
<td>2 12:00</td>
</tr>
</tbody>
</table>

Figure 24. Thresholds Definition Panel
How SA z/OS Uses Error Thresholds

Thresholds for MVS Components
You can define thresholds for components or functions of z/OS:

- **MVSDUMP**: for z/OS dump data sets
- **SMFDUMP**: for SMF dump data sets
- **LOGREC**: for SYS1.LOGREC data sets
- **SYSLOG**: for system log data sets

For example, a threshold for an z/OS dump data set specifies, how often a full dump data set should be recovered in which time interval, before the operator gets a message, specifying that something is supposed to go permanently wrong.

![Figure 25. Component Definition: MVSDUMP Panel](image)

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.

In our example from Figure 25, if a certain error occurs once per hour, it is considered to be infrequent. However, if an z/OS dump (MVSDUMP) runs full more often than three times an hour. the critical stage is reached. In such a case, SA z/OS will no longer recover the dump, but will notify the operator of this occurrence.

Automation Flags

One of the main tasks of customizing SA z/OS system operations involves setting automation flags. You can access Flag Automation Specification panels when defining Applications (individual applications or application classes), Application Defaults sets, MVS Component sets, System Defaults sets and MVSCOMP Defaults sets.

The Flag Automation Specification panels of the customization dialog define the cumulative effects of your automation flag settings, including:

- When automation is active
- Resources to which automation applies
- 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 assist mode, an SA z/OS option allowing a human operator to monitor and verify 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.

The following topics are discussed:

- “Types of Automation Flags”
- “Setting Automation Flags” on page 37
- “Automation Manager Flags” on page 38
- “Relationship Between Automation Flags and Resource Status” on page 38
- “Automation Flags and Minor Resource Flags” on page 39

For more information on how to use automation flags and when SA z/OS checks automation flags, refer to System Automation for z/OS Customizing and Programming.

Types of Automation Flags

Figure 26 shows an example of a Flag Automation Specification panel that is displayed when the AUTOMATION FLAGS policy item for an Application is selected (refer to Figure 106 on page 151).

<table>
<thead>
<tr>
<th>COMMANDS</th>
<th>ACTIONS</th>
<th>HELP</th>
</tr>
</thead>
</table>
| AOFGFA1   | Flag Automation Specification | Command ==>

Entry Type : Application  PolicyDB Name : DATABASE_NAME
Entry Name : APPC  Enterprise Name : YOUR_ENTERPRISE

Resource: APPC
Enter level of automation desired.

Automation Flags: Y = Yes  N = No  E = Exits
Assist Flags: D = Display  L = Log  N = None

<table>
<thead>
<tr>
<th>Actions</th>
<th>Flag</th>
<th>Auto</th>
<th>Assist</th>
<th>Exits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Recovery</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shutdown</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initstart</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Restart</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Enter or Display times to disable automation. NO  Yes  No

Figure 26. Flag Automation Specification Panel

There is one flag that controls all the other flags:

- **Automation**

Automation is the controlling flag for resource automation. During automated operations, SA z/OS first checks the Automation flag when determining whether automation is on or off for a resource and which operator functions are automated. If the setting for this flag is YES, SA z/OS checks specific flag settings for that resource. If the setting for this flag is NO, automation is completely turned off for the resource, and SA z/OS does not check any other flags.
Automation Flags

The other flags which apply to each resource are:

- **Recovery**
  This flag determines if SA z/OS monitors a resource for predefined situations requiring recovery activity and takes necessary corrective actions when these situations occur. Some examples of situations requiring recovery activities include write-to-operator buffer (WTOBUF) shortages and full MVS system maintenance data sets, such as LOGREC, SYSLOG, and MVS dump, that require offloading. By automatically responding to error conditions, SA z/OS can prevent resource outages, thereby increasing system availability.

- **Start**
  This setting manages the startup process for an application by ensuring appropriate actions are taken, completely initializing the application, and bringing the application to an up or ready state (when the application issues an UP message, indicating that it is ready for use).

  For resources that have a startup process consisting of issuing several commands and handling several replies, you can use the Start flag to determine how much of the startup process is automated. For example, if you want to automate initial startup process for a resource but have an operator handle the rest of the startup process you can set the Initstart flag to YES, and the Start flag to NO. The resource start phase then requires operator intervention. You might want to set your flags like this for situations when you do not want to use the automated predefined startup procedures but want to issue unique startup commands.

- **Shutdown**
  This flag determines if a shutdown command can be accepted and processed for an application. It also determines if secondary automation will be processed if a manual shutdown is issued.

  **Note:**
  If you use the override parameter on INGREQ to force an automated shutdown when the Shutdown flag is off, secondary automation will not occur as a result of the shutdown. This setting manages the shutdown process for an application by ensuring that actions are taken to stop the application in an orderly fashion, such as issuing shutdown commands and replies.

- **Initstart**
  This setting controls the initial startup process for an application following an SA z/OS initial status determination cycle. SA z/OS conducts status determination cycles when:
  - SA z/OS is initialized after an IPL.
  - SA z/OS is initialized when its host NetView is restarted.
  - The automation control file is reloaded.
  - The RESYNC command is issued.

  Those applications found to be inactive during the first SA z/OS initialization after an IPL are assigned an initial status of DOWN, if the status prior to the IPL and the various other options permit.

  Following SA z/OS initialization, those applications that have a status of DOWN will be started if the Initstart flag setting is YES. SA z/OS will ensure that the application is started in the order specified in automation policy and that any prerequisite applications are functional.

  For some resources, the activities controlled by the Initstart setting completely start up an application. That is, the MVS START command is sufficient to bring
Automation Flags

the resource to a STARTED state. For other resources, the startup process is more
involved, with additional commands issued and replies handled. For such
resources, you can use the Start automation flag to automate the rest of the
startup process or indicate that an operator will handle the rest of the process.

• Restart
The setting of this flag controls the restarting of an application, either when it
ends abnormally (an *abend* condition), when automation shuts it down, or when
it shuts down outside the control of SA *z/OS*. By automatically responding to
the application requiring a restart, SA *z/OS* automation decreases the time and
system resources required to get an application up and running again.

Error thresholds influence whether SA *z/OS* restarts resources. During a
condition requiring restart, SA *z/OS* checks whether the number of occurrences
of the condition exceeds the critical threshold defined in the automation policy.
If the critical threshold is exceeded, SA *z/OS* does not attempt restarting a
resource. See "How SA *z/OS* Uses Error Thresholds" on page 31 for more
information.

The *Exits* column displays the number of exits defined. Automation exit data is
added to the Automation Control File (ACF) whenever there are any additional
definitions for that flag. This additional data may be, for example, automation
disable times.

If you want to enable automation, but specify times when it should be disabled,
you can enter YES in the input field at the bottom of the screen. Another panel will
be displayed where you can specify one or more time ranges specifying the times
when automation will be disabled.

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)
• *MVS/COMP Defaults* (using their AUTOMATION FLAGS policy item)
• *Applications* (using their AUTOMATION FLAGS policy item)
• *MVS Components* (using their MINOR RESOURCE FLAGS policy item)

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

Performance Recommendation
It is recommended that you consider carefully which flags are defined
explicitly in this manner. In order to improve runtime performance, the
effective flag value for each resource (Automation, Initstart, Startup, Recovery,
Shutdown, and Restart) is resolved during initialization processing. This can
have a noticeable impact on the time required for SA *z/OS* to initialize.

Since 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.
Automation Flags

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 common routine performs this automation flag check. See 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 for which automation is not to apply 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 System Automation for z/OS Customizing and Programming for more information.

Automation Manager Flags

There are two automation flags that are not set in the customization dialog, but are set either by an operator command or when the automation agents are initialized. These are:

- the global automation flag:
  
  Using the INGLIST or the INGSET command (see System Automation for z/OS User’s Guide or 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. This can be done for resources that reside on systems that are currently inactive, for example, to prevent the startup of the resource at IPL time of the system.

- the hold flag:
  
  This flag can be set with the INGSET command but it is set automatically by SA z/OS in the following way:
  
  - If you initialize the automation agents you get the WTOR message AOF603D where you can select the NOSTART option. If you select this option, SA z/OS internally sets the hold flag for every resource, so that no resource (subsystem) is automatically started until this flag is overridden by an operator’s explicit INGREQ request.

  The purpose of this flag is to enable the startup of manually selected subsystems instead of starting all subsystems at a time.

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 System Automation for z/OS Customizing and Programming and System Automation for z/OS User’s Guide.

SA z/OS provides an exit for processing state changes. Refer to 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 RESOURCE FLAGS policy item) that control responses to specific situations. The advanced automation option AOFMINORCHECK (described in System Automation for z/OS Customizing and Programming) determines whether flags for minor resources are checked. Typical minor resources include message IDs and states.

For example, JES2 is defined as application JES. You have defined a reply to message $HASP099 on the MESSAGE policy item of the Application policy object for JES2. You can change the reply to $HASP099 by:

- Turning the Shutdown flag for JES off. This means you have to shutdown JES manually.
- Turning the Shutdown flag for minor resource JES.$HASP099 off. This allows SA z/OS to shutdown JES normally. When the $HASP099 message is issued, shutdown processing will wait for a manual response.

Event and Trigger Support
This section informs you about the concepts of events and triggers and how you can use them. This information is useful to understand how to define these automation policies using the customization dialog.

How events and triggers are defined is described in sections:
- “Events Policy Object” on page 221
- “Triggers Policy Object” on page 226

Simply spoken, these facilities 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.
Event and Trigger Support

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, then 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 refer to [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 system. 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** are connected to applications, application groups, or monitor resources. 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, then 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 27. How a Trigger Uses Events](image)

In **Figure 27** 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 Parallel 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 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**

A trigger is a kind of a traffic light that controls whether a running application should stop, or whether an application with 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.

A startup condition is satisfied, when all events of the condition are set (and, for SA OS/390 1.3, the current time is within the service window for an eventually connected service period). A shutdown condition is satisfied, when all its events are set (and, for SA OS/390 1.3, the current time is outside of any service window for an eventually connected service period).

---

**Figure 28. Triggers, Events and Conditions**

Figure 28 shows another scenario, how triggers, events and conditions work together. For example, if EVENT1 and EVENT2 are set, then Startup Condition 1 is true and Trigger T1 starts the application - provided that the application’s goal is being available. If EVENT4 is set, then Shutdown Condition 1 is true and Trigger T1 stops the application - provided that the application’s goal is being unavailable.
Event and Trigger Support

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

Assigning System Automation Symbols (AOCCLONE)

You can assign up to 36 system automation symbols to each system (for example, &AOCCLONE., ~, &AOCCLONE1. through &AOCCLONE9., and &AOCCLONEA. through &AOCCLONEZ.). When you use a system automation symbol in a job name, each system can run with a unique job name while the application or subsystem shares a single automation policy across systems. The substitution can occur anywhere within the job name. Any of the 36 available system automation symbols can be used. You may specify more than one system automation symbol.

<table>
<thead>
<tr>
<th>Note:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Be careful when cloning applications controlled by Automatic Restart Manager. In most cases, the applications should be defined to Automatic Restart Manager as restart-in-place only. If you allow cross-system restarts for cloned applications, they may be restarted more than once on the same system.</td>
</tr>
</tbody>
</table>

The usage of system automation symbols is not limited to sysplexes. In addition to qualifying the job name, you can use system automation symbols for substitution anywhere within a defined literal, such as a shutdown command. The value of system automation symbols is also available for your automation through the NetView AOFAOCCLONE common global variables (1 to 9, and A to Z). The AOFAOCCLONE and each AOFAOCCLONEn variable contains the value of the corresponding AOCCLONE variable.

SA z/OS also supports the specification of a procname in addition to a job name. The standard startup command is:

MVS $ jobname

but if a proname is specified, SA z/OS issues:

MVS $ procname,JOBNAMe=jobname

<table>
<thead>
<tr>
<th>Note:</th>
</tr>
</thead>
<tbody>
<tr>
<td>The job name is the result of the substitution of any applicable &amp;AOCCLONE. value and will be the name that is used for the whole automation of the application.</td>
</tr>
</tbody>
</table>

To define an identical application with different job names, follow these steps:

1. Define your system, optionally assigning values to one or more of the system automation symbols.
2. Link the application through an application group to the system.
3. Build the automation policy for the system. The system automation symbols are written to the automation control file as part of the system definition.
4. Build the application policy. If system automation symbols are used in the job name, they are replaced with the appropriate &AOCCLONE. values.
5. Load the automation control file. The system automation symbols are replaced with the appropriate values.
Using System Symbols and System Automation Symbols

You can specify system symbols as well as system automation symbols (AOCCLONE variables) in the following fields in the customization dialog:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Policy Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description (Short and Long) 1</td>
<td>Policy DESCRIPTION of all entry types</td>
</tr>
<tr>
<td>ARM element name</td>
<td>Policy APPLICATION INFO of entry type APL</td>
</tr>
<tr>
<td>WLM resource name</td>
<td>Policy APPLICATION INFO of entry type APL</td>
</tr>
<tr>
<td>Job name</td>
<td>Policy APPLICATION INFO of entry type APL</td>
</tr>
<tr>
<td>JCL procedure name</td>
<td>Policy APPLICATION INFO of entry type APL</td>
</tr>
<tr>
<td>Command prefix</td>
<td>Policy AUTOMATION of entry type APL</td>
</tr>
<tr>
<td>IMS Control Region Name</td>
<td>Policy IMS CONTROL of entry type APL</td>
</tr>
<tr>
<td>IMS Subsystem ID</td>
<td>Policy IMS CONTROL of entry type APL</td>
</tr>
<tr>
<td>DB2 Subsystem name</td>
<td>Policy DB2 CONTROL of entry type APL</td>
</tr>
<tr>
<td>DB2 Active Log Dataset Name</td>
<td>Policy DB2 CONTROL of entry type APL</td>
</tr>
<tr>
<td>CICS Major Node Name</td>
<td>Policy CICS CONTROL of entry type APL</td>
</tr>
<tr>
<td>CICS APPLID</td>
<td>Policy CICS CONTROL of entry type APL</td>
</tr>
<tr>
<td>Minor resource names 2</td>
<td>Policy MINOR RESOURCE FLAGS of entry type APL and MVC</td>
</tr>
<tr>
<td>Message ID</td>
<td>Policy MESSAGES/USER DATA of entry type APL and MVC</td>
</tr>
<tr>
<td>Command text 1</td>
<td>Policy MESSAGES/USER DATA (type CMD), STARTUP, SHUTDOWN, JES2/3 SPOOLFULL,</td>
</tr>
<tr>
<td></td>
<td>JES2/3 SPOOLSHORT, JES3 MONITOR of entry type APL, policy</td>
</tr>
<tr>
<td></td>
<td>MESSAGES/USER DATA of entry type MVC, policy</td>
</tr>
<tr>
<td></td>
<td>TIMERS of entry type TMR</td>
</tr>
<tr>
<td>Reply text 1</td>
<td>Policy MESSAGES/USER DATA (type REPLY), SHUTDOWN of entry type APL, policy</td>
</tr>
<tr>
<td></td>
<td>MESSAGES/USER DATA of entry type MVC</td>
</tr>
<tr>
<td>Code1/2/3, Value Returned 1</td>
<td>Policy MESSAGES/USER DATA (type CODE) of entry type APL and MVC</td>
</tr>
<tr>
<td>Data 1</td>
<td>Policy MESSAGES/USER DATA (type USER) of entry type APL and MVC</td>
</tr>
<tr>
<td>Data</td>
<td>Policy E-T DATA of entry type UET</td>
</tr>
<tr>
<td>Temporary Data Set HLQ</td>
<td>Policy SYSPLEX of entry type GRP</td>
</tr>
<tr>
<td>Monitoring Command 1</td>
<td>Policy USS CONTROL of entry type APL</td>
</tr>
<tr>
<td>Command/Path 1</td>
<td>Policy USS CONTROL of entry type APL</td>
</tr>
<tr>
<td>File Name 1</td>
<td>Policy USS CONTROL of entry type APL</td>
</tr>
<tr>
<td>User ID</td>
<td>Policy USS CONTROL of entry type APL</td>
</tr>
<tr>
<td>Console Name</td>
<td>Policy AUTOMATION CONSOLE of entry type SYS</td>
</tr>
<tr>
<td>UNIX Installation Path 1</td>
<td>Policy AUTOMATION SETUP of entry type SYS</td>
</tr>
<tr>
<td>Activate command 1</td>
<td>Policy MONITOR INFO of entry type MTR</td>
</tr>
<tr>
<td>Deactivate command 1</td>
<td>Policy MONITOR INFO of entry type MTR</td>
</tr>
<tr>
<td>Monitor command 1</td>
<td>Policy MONITOR INFO of entry type MTR</td>
</tr>
</tbody>
</table>
Assigning System Automation Symbols (AOCCLONE)

Table 3. System Symbols and System Automation Symbols (continued)

1. If you are using system automation symbols in fields that are not automatically translated to uppercase, these variables must be entered in uppercase.
2. The ending period of a system automation symbol for minor resource names must be followed by at least one character, for example, DAT&AOCCLONE2.A

Table 4 lists the fields in the customization dialog where you can only specify system symbols.

Table 4. System Symbols

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Policy Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automation Operators</td>
<td>policy OPERATORS of entry type AOP</td>
</tr>
<tr>
<td>NetView Operators</td>
<td>policy OPERATORS of entry type AOP</td>
</tr>
<tr>
<td>Console Name</td>
<td>policy OPERATORS of entry type AOP</td>
</tr>
</tbody>
</table>

Table 5 lists the field in the customization dialog where you can only specify system automation symbols (AOCCLONE variables).

Table 5. System Automation Symbols

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Policy Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supporting Resource Name</td>
<td>Policy RELATIONSHIPS of entry type APL and APG</td>
</tr>
</tbody>
</table>
Chapter 4. Scenarios on How to Use the Customization Dialog

This chapter provides a series of scenarios on how to use the customization dialog to implement the appropriate automation policy:

This section discusses the following scenarios:

- "How to Work with Resources"
- "How to Use Schedules (Service Periods)" on page 69
- "How to Use Triggers and Events" on page 70
- "How to Use User E-T Pairs" on page 72
- "How to Implement Message Processing with User-Defined Data" on page 74

How to Work with Resources

This chapter contains a scenario on how to group resources for applications in a complex configuration. This scenario is based on the grouping scenario from "A Grouping Scenario" on page 24.

Some background information about resources will be given where required.

The complete process of grouping resources is divided into three steps:

- "Step 1: How to Create the Policy Objects that Make Up the Sysplex"
- "Step 2: How to Create the Applications" on page 47
- "Step 3: How to Create the Application Groups" on page 48

Step 1: How to Create the Policy Objects that Make Up the Sysplex

You start on the Entry Type Selection panel. It is assumed that the policy database is empty.

Creating a sysplex consists of:

- creating the systems
- creating the sysplex group
- link the systems to the sysplex group

Step 1.1: How to Create the Systems

In this substep, you create the systems SYSTEM1, SYSTEM2 and SYSTEM3. You may want to read the information in "Creating a New System" on page 105 to see some sample panels of the customization dialog before you continue this scenario.

1. On the Entry Type Selection panel enter option 4 or SYS to get to the Entry Name Selection panel for systems which displays an empty list.
2. Enter NEW SYSTEM1 into the command line.
   The Define New Entry panel for systems is displayed; the Name field is already filled with SYSTEM1.
3. Enter SYSTEM1 into the MVS SYSNAME field and press the PF3 (END) key twice to leave this panel.
Note:
To leave this panel, you have to press PF3 twice because the first PF3 prompts you for the ProcOps name, which is defaulted to the MVS SYSNAME if you do not change it.

Background:
What is a Resource?
The MVS SYSNAME name will be part of the resource names that have a system scope.

This name usually consists of three parts:
- an 11 character identifier
- the entry type from which it was derived
- and usually the System it runs on (that is, the specified MVS SYSNAME for that system).

For sysplex-wide resources (for example, Sysplex Application Groups), the system identifier is omitted.

Resources of type APL, APG, or MTR are automatically generated when applications, application groups, or monitor resources are linked to systems. Resources of other types are generated dynamically when the system operations configuration files are built.

Within the customization dialog, resources of type APL, APG, or MTR can be viewed and they can be linked via special policy items.

The Policy Selection panel for systems is displayed, showing a message indicating that the system is created.

4. Press PF3 or enter END to return to the Entry Name Selection panel showing SYSTEM1 in the list.

5. Repeat the previous steps to create SYSTEM2 and SYSTEM3. Now you have three systems in the list on the Entry Name Selection panel. Press PF3 to return to the Entry Type Selection panel.

Step 1.2: How to Create the Sysplex
You may want to read the information in "Creating a New Group" on page 90 to see some sample panels of the customization dialog before you continue with the scenario.

1. On the Entry Type Selection panel, enter option 2 or GRP to get to the Entry Name Selection panel for groups which displays an empty list.

2. Enter NEW SYSPLEX1 into the command line.

   The Define New Entry panel for groups is displayed; the Name field is already filled with SYSPLEX1.

3. Change the Group Type field to SYSPLEX as you want to create a sysplex group.

   Press PF3 to exit this panel. The Policy Selection panel for Groups is displayed, showing a message that the group is created.
4. On the Policy Selection panel, select policy item SYSTEMS. The Systems for Group panel is displayed showing the three systems defined in “Step 1.1: How to Create the Systems” on page 45.

5. Select all three systems with an ‘S’ and press PF3 twice.
   Now your environment contains a sysplex with three z/OS systems.

---

**Background:**

*Ways to Build a Sysplex*

A sysplex consists of z/OS systems within an XCF group with the same group ID. In the customization dialog such a sysplex can be built by creating a sysplex group and linking its systems.

This happens either "bottom-up" (from a system to a group, with the system’s WHERE USED policy item) or "top-down" (from a group to a system, with the group’s SYSTEM policy item).

For the "bottom-up" approach you must go into the SYS entry type, select all systems you want in the Sysplex, select each system’s WHERE USED policy item where you get a list of sysplex groups and subgroups to which you can link the system.

You can also link groups with systems indirectly. This happens if you create SubGroup objects (SBG entry type) and link these subgroups "bottom-up" to groups AND "top-down" to Systems.

Sysplex-Groups usually consist of multiple systems; however, systems only can be linked into one sysplex group (either directly or indirectly).

```
SYSPLEX1
  1  2
  SYSTEM1 SYSTEM2
  SUBGRP1 SUBGRP2
  SYSTEM3 SYSTEM4
```

The previous figure describes how you can create a sysplex:

1. **SYSTEM1** and **SYSTEM2** are linked directly to sysplex **SYSPLEX1**.
2. **SYSTEM3** and **SYSTEM4** are linked to subgroup **SUBGRP2** and this subgroup is linked to **SYSPLEX1**.

---

**Step 2: How to Create the Applications**

In this step, you create all required application subsystems.

You may want to read the information in “Creating a New Application” on page 145 to see some sample panels of the customization dialog before you continue with the scenario.
1. On the *Entry Type Selection* panel enter option 6 or APL to get to the *Entry Name Selection* panel for applications, which is displayed with an empty list.

2. Enter NEW VTAM into the command line.
   The *Define New Entry* panel for applications is displayed; the *Name* field is already filled with VTAM.

3. Enter VTAM into the *Subsystem Name* field. You must also enter a *Jobname*. This field is mandatory and you can use VTAM, too.

   **Background:**
   
   *Application Resource Name*
   
   The Subsystem Name is used as first part of the application’s resource name. It is recommended to use the same name for the application entry name and subsystem name for better identification of applications and resources that will be derived from applications.

4. Then press the PF3 to leave this panel. The *Policy Selection* panel is displayed, showing a message that the application is created.

5. On the *Policy Selection* panel for applications press PF3 again to return to the *Entry Name Selection* panel. It now shows VTAM in the list.

6. Repeat the previous steps for the rest of the applications:
   - WBSRV (Web Server)
   - DWH (Data Warehouse)
   - CICSMAIN
   - CICSTOR
   - CICSFOR
   - CICSAOR1
   - CICSAOR2
   - CICSAOR3
   - DB2MAIN
   - DB2IRLM
   - TCP/IP

   Now all subsystems (applications) are defined to your environment and the *Entry Name Selection* panel shows 12 applications.

**Step 3: How to Create the Application Groups**

In this step, you create all required application groups and define the behavior of their members.

According to the example from "A Grouping Scenario" on page 24, the groups should run on the following systems:

- WBSRVAPG should run on SYSTEM1.
- DWHAPG should run on SYSTEM2 and alternatively on SYSTEM3.
- DB2 should run on SYSTEM2 and SYSTEM3.
- NET should run in the entire sysplex.
- CICS should run on SYSTEM2 and on SYSTEM3.
### Background:

*Sysplex Application Groups and System Application Groups*

The application group type (SYSTEM or SYSPLEX) sets the boundaries for the automation of the application group and its members.

- Sysplex Application Groups may reside on multiple systems within one or more sysplexes.
- System Application Groups reside on single systems.

Use Sysplex Application Groups for DWHAPG, NET, and CICS. System Application Groups will be used for WBSRVAPG and DB2.

### Step 3.1: Create Sysplex Application Group CICSAOR

You start with the creation of CICSAOR because CICSAOR contains only applications (CICSAOR1, CICSAOR2, CICSAOR3) that already have been defined. Then you create the application group with a nature of BASIC.

You may want to read the information in “Creating a New ApplicationGroup” on page 125 to see some sample panels of the customization dialog before you continue this scenario.

1. On the *Entry Type Selection* panel enter option 5 or APG. The *Entry Name Selection* panel is displayed with an empty list.

2. On the *Entry Name Selection* panel enter NEW CICSAOR into the command line. The *Define New Entry* panel for application groups is displayed; the *Name* field is already filled with CICSAOR.

3. Check that the *Application Group Type* field contains SYSPLEX.

4. Enter BASIC into the *Nature* field.
Background: Nature of an Application group

SA z/OS knows three natures:

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

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

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

The nature of an application group can be changed later on. However be careful with such a change, because the role of the application group will change too (and definitions made for example for a MOVE group may not be appropriate for a SERVER group).

In the example, the application group consists of three applications, (CICSAOR1, CICSAOR2, and CICSAOR3) which all should be available. You should therefore decide to take a BASIC Group.

5. Enter CICSAOR into the Automation Name field.

6. Press PF3 to exit the panel. The Policy Selection panel for Groups is displayed showing a message that the application group is created.

7. On the Policy Selection panel select the WHERE USED policy item. The Where Used panel is displayed with sysplex group SYSPLEX1.

8. Select the sysplex group and press PF3. For a moment, you can see the Command Progress Display popup window telling you that the resource has
been created. This popup is just a quick information that will disappear automatically. You can see the generated resources under the corresponding policy item of the application group.

Finally, you are back on the Policy Selection panel for CICSAOR.

Background:

Generating Sysplex Application Group Resources

A Sysplex Application Group resource is created automatically if:
• there is an Automation Name for the application group, and
• there is a link to a sysplex group.

Only one resource will be created (not one resource per system) and the resource name has no system identifier.

CICSAOR has got an Automation Name; therefore a resource for this application group will be generated.

The current configuration with application group CICSAOR is now:

Figure 29. Current Configuration with Sysplex Application Group CICSAOR

9. On the Policy Selection panel for CICSAOR select the APPLICATIONS policy item.
   The Applications for ApplicationGroup panel will be displayed with all twelve applications.

10. Select all of
    CICSAOR1
    CICSAOR2
    CICSAOR3

and press PF3 (END).
You are now back on the Policy Selection panel for CICSAOR.
Background:

Generating Application Resources

Application Resources are generated when Applications are linked to Systems. This now is the case for CICSAOR1, CICSAOR2, CICSAOR3: all now are linked to CICSAOR, and indirectly via sysplex SYSPLEX1 they are linked to SYSTEM1, SYSTEM2 and SYSTEM3.

For each linked system, one resource is generated.

Again, you can see the Command Progress Display popup window telling you that the resources have been created. For example, for CICSAOR1, the following resources have been generated:

- CICSAOR1/APL/SYSTEM1
- CICSAOR1/APL/SYSTEM2
- CICSAOR1/APL/SYSTEM3

For each application three resources are generated and linked to the application group resource.

Note: These Application Resources are deleted when:

- either the application is deleted,
- or the application group CICSAOR is deleted,
- or if other parts of the link chain APL-APG-GRP-SYS are deleted or de-linked (that is, if SYSPLEX1 or Systems SYSTEM1...3 or links between them are deleted).

The current links are:

![Figure 30. Current Configuration with Applications CICSAOR1 through CICSAOR3](image-url)
11. On the Policy Selection panel for CICSAOR select the RESOURCES policy item. The Select Resources panel is displayed with 9 application resources selected. You can see this panel in Figure 31.

12. On the Select Resources panel for CICSAOR, deselect the three resources on SYSTEM1, because CICS should not run on SYSTEM1. Type an M (which is the action code for REMOVE) into the Action column in front of each resource that you want to remove from the group.

When you press ENTER, the field Number of selected Resources now shows a 6.

13. Press PF3 (END) to return to the Policy Selection panel for CICSAOR.

14. Select policy item APPLGROUP INFO to get to the Application Group Information panel for CICSAOR.

15. Change the field Automatically link Application-Resources to NO.

Reason: Application group definitions are complete now. Further modifications within the sysplex should not affect this completed resource selections on panel Select Resources.
How to Work with Resources

Background:

*Automatically link Application-Resources*

Set this input field to NO if the definition of an application group resource (with its linked applications, its nature and its behavior and so on) is complete. This prevents possible changes in the automation policy, like changed system links or application links, from having an influence on this application group resource.

If this field is set to YES, the application group resource will be changed if system links or application links occur, namely, the number of generated application resources, the number of linked application resources and possibly the behavior of the application group resource will change.

For example, if you link another system to SYSPLEX1, you would get three more resources into your application group CICSAOR, one for each linked application.

Later on, you will specify preference values for application resources, where automatically generated resources with their default preferences could adversely affect your complete application group definition.

16. Press PF3 until you are back on the *Entry Type Selection* panel.

Now application group CICSAOR is complete: if a start request is issued against CICSAOR, also start requests to CICSAOR1, CICSAOR2 and CICSAOR3 are automatically issued on SYSTEM2 and SYSTEM3.

**Step 3.2: Create System Application Group DB2**

The application group DB2 runs on two specific systems, so you define it as a System Application Group.

1. On the *Entry Type Selection* panel, enter option 5 or APG.

   The *Entry Name Selection* panel is displayed with the application group CICSAOR.

2. Enter ’NEW DB2’.

   The *Define New Entry* panel for application groups is displayed; the *Name* field is already filled with DB2.

3. In the scenario you decided to let DB2 permanently run only on two out of the three systems of SYSPLEX1, namely on SYSTEM2 and SYSTEM3. Thus you type "SYSTEM" into the *Application Group Type* field.

   You merely want DB2 be started with all of its components, without specific backup considerations. Thus you leave its *Nature* field at "BASIC".

   For resource generation, enter DB2 into the *Automation Name* field.

4. Press PF3 to reach the *Policy Selection* panel.

**Note:**

No resource is created for the APG at this point in time, because the application group is not yet linked to one or more systems.

The *Policy Selection* panel is displayed indicating that the application group is created.
5. Select the WHERE USED policy item. The Where Used panel is displayed with the three systems.

6. DB2 should only run on SYSTEM2 and SYSTEM3. Thus select SYSTEM2 and SYSTEM3 and press PF3 (END). You will see the Command Progress Display popup window telling you that two resources have been created:
   DB2/AGP/SYSTEM2
   DB2/AGP/SYSTEM3

   **Background:**

   **Generating System Application Group Resources**

   A System Application Group resource is automatically generated
   • if there is an Automation Name for the application group, and
   • if there is link to a system.

   For each linked system, one resource is generated.

   There are two ways to link a System Application Group to a system:
   • For the application group use the WHERE USED policy item on the Policy Selection panel or the WHEREUSED (W) command on the Entry Name Selection panel to get to the list of systems, and then select at least one system.
   • If you are on the Policy Selection panel for a system, select the APPLICATION GROUPS policy item and then select an application group.

   The current configuration is shown in **Figure 32**

   ![Figure 32. Current Configuration with Application Group DB2](image)

   7. On the Policy Selection panel for DB2, select the APPLICATIONS policy item to link the required applications.
      
      The Applications for ApplicationGroup panel is displayed with all twelve applications.

   8. Select DB2MAIN and DB2IRLM and press PF3 (END) to return to the Policy Selection panel.
      
      Now DB2MAIN and DB2IRLM are linked to two systems via the application group DB2. For the application group, the field **Automatically link Application-Resources into APG** is set to YES, which is the default.
How to Work with Resources

One resource is generated for each system-link of each application. The Command Progress Display popup window indicates that the following four resources are generated:

- DB2IRLM/APL/SYSTEM2
- DB2IRLM/APL/SYSTEM3
- DB2MAIN/APL/SYSTEM2
- DB2MAIN/APL/SYSTEM3

![Figure 33. Current Configuration with Applications Linked to DB2](image)

9. On the Policy Selection panel for application group DB2, select the RESOURCES policy item.

The Select System panel with an indication of the number of resources selected for each system is shown. You can compare the resulting panel on your system with Figure 34.

![Figure 34. Select System Panel for DB2 Application group](image)

A System Application Group can be linked to several systems which generates application resources for those systems. So to work with the resource links, you have to specify first which system you want to get the resources for. This is done on the Select System panel by selecting one of the systems.

10. On the Select System panel, select either of the SYS Resources SYSTEM2 or SYSTEM3 and press ENTER.

The Select Resources panel with, for example, the two APL Resources for SYSTEM2 is displayed. You can compare the resulting panel on your system with Figure 35 on page 57.
Both applications should be started, thus no change on this panel is required.

11. On the Select Resources panel, Press PF3 (END) or CANCEL. This returns you to the Select System panel, from where you can view the Select Resources panel for the APL resources in SYSTEM3.

12. On the Select Resources panel with APL Resources for SYSTEM3, press PF3 (END) twice now to return to the Policy Selection panel for application group DB2.

13. Select policy item APPLGROUP INFO to reach the Application Group Information panel. Change the field Automatically link Application-Resources into APG to NO.

The DB2 application group is now complete.

14. Now press PF3 until you are back at the Entry Type Selection panel.

15. Now the application group DB2 is complete: if a start request is issued against DB2 on SYSTEM2, start requests to the application resources DB2MAIN and DB2IRLM are also automatically issued for SYSTEM2. A start request against application group DB2 on SYSTEM3 does the same for SYSTEM3.

**Step 3.3: Create System Application Group CICS**

You will create this application group with the nature of BASIC and with links to three applications (CICSMAIN, CICSTOR, CICSFOR). These applications are already defined.

1. On the Entry Type Selection panel, enter option 5 or APG. To create the CICS System Application Group you have to repeat steps 2 through 10 of "Step 3.2: Create System Application Group DB2" on page 54 accordingly.

On the Applications for ApplicationGroup panel, select applications

CICSMAIN
CICSTOR
CICSFOR

2. The Select System panel is displayed as shown in Figure 36 on page 58
How to Work with Resources

Select SYSTEM2. A panel is displayed similar to the one shown in Figure 37.

Check the created links for SYSTEM3 too. The selected links should be the same.

3. Press PF3 to get back to the Policy Selection panel and select policy item APPLGROUP INFO. Once again, change the field Automatically link Application-Resources to NO.

4. Press PF3 until you are back on the Entry Type Selection panel.

Now application group CICS is complete: if a start request is issued against CICS, start requests are also issued to application resources CICSFOR, CICSMAIN, CICSTOR (on SYSTEM2 and SYSTEM3).
Step 3.4: Create Sysplex Application Group NET

The creation of Sysplex Application Group NET follows the previous examples, therefore, no detailed description of the required steps is contained here.

The Sysplex Application Group NET should run in the entire SYSPLEX1, and the applications VTAM and TCP/IP should be linked to it.

After your definitions are complete, the Select Resources panel for the NET application group should look like Figure 39.

![Figure 39. Select Resources Panel for Application Group NET](image)
How to Work with Resources

Step 3.5: Create MOVE Group DWHAPG

The Data Warehouse Group DWHAPG should run on one system (SYSTEM2) and, if something goes wrong on SYSTEM2, alternatively on SYSTEM3. This application group is created with nature MOVE.

Background:

What are Move Groups?

All previous application groups were of nature BASIC: BASIC means that all of its components are peers and all of them need to be available before the application group is considered to be available.

A MOVE Group, as a contrast, is designed for backup solutions. A MOVE Group can have many components, but exactly 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).

The applications needed for the DWHAPG application group are available. Thus no additional preparation seem to be needed (an error, as you will see).

1. On the Entry Type Selection panel, enter option 5 or APG. The Entry Name Selection panel is displayed with four application groups: CICS, CICSAOR, DB2, NET.

2. Enter NEW DWHAPG into the command line. The Define New Entry panel is displayed; the Name field is already filled with DWHAPG.

3. The application group DWHAPG should run alternatively on two out of the three systems. Thus leave the Application Group Type field as "SYSPLEX". You want to start the components of DWHAPG exactly once, but with backups. Thus you change its nature to "MOVE".

For resource generation you need an automation name. Therefore enter DWHAPG in the Automation Name field.

4. Press PF3 to exit the panel. The Policy Selection panel for Groups is displayed showing a message that the application group has been created.

5. On the Policy Selection panel select the WHERE USED policy item. The Where Used panel is displayed with sylsplex group SYSPLEX1.

6. Select the sylsplex group and press PF3.

You will see the Command Progress Display panel telling you that the DWHAPG/APG resource has been created.

7. Select the RESOURCES policy item. You will see the Select Resources panel with an empty list of resources. The reason why no resource seems to be available is that you did not link any applications to the application group - thus no application resources were generated.

Note that the Availability Target is 1 (exactly 1 resource is supposed to be up in a MOVE Group). You cannot change this value.

8. Change the field Show all Resources to YES.

The Select Resources panel is displayed with all valid application group resources. Valid application group resources are Sysplex Application Group resources on the same sylsplex (CICSAOR/APG, NET/APG) and System Application Group resources on one of the systems in the Sysplex (CICS/APG/SYSTEM2, CICS/APG/SYSTEM3, DB2/APG/SYSTEM2 and DB2/APG/SYSTEM3). [Figure 40 on page 61] shows this panel.
Figure 40. Select Resources Panel for DWHAPG with valid APG-Resources

However, the resulting panel shown in Figure 40 is not yet what you wanted: The target of your group was to start all of these components plus the DWH Application on either SYSTEM2 or SYSTEM3.

The target, to start the whole package alternatively on SYSTEM2 or SYSTEM3 cannot be reached with the described definitions.

Solution: You have to define two additional application groups with the same contents - one with focus on SYSTEM2 and one with focus on SYSTEM3. Their names will be DWHAPG2 and DWHAPG3. DWHAPG will have these two APGs as member and start them alternatively.

However, you must define these groups as System Application Groups because they must have the System Application Group CICS on SYSTEM2 and SYSTEM3.

9. Press PF3 until you are back on panel Entry Name Selection and enter NEW DWHAPG2 on the command line.
Or, as a fast path, specify =APG;NEW DWHAPG2.

The Define New Entry panel is displayed, with DWHAPG2 already filled in the Name field.

10. DWHAPG2 should run on a single system. Therefore, select type SYSTEM on the Define New Entry panel.

As you want to start all of DWHAPG2’s components, select BASIC.

For resource generation you need an automation name. Type DWHAPG2 in the related field and press PF3 to exit the panel.

11. On the Policy Selection panel select the WHERE USED policy item and on the Where Used panel select sysplex group SYSTEM2.

12. Press PF3 to exit the panel. The Command Progress Display panel is displayed, showing that the resource DWHAPG2/APG/SYSTEM2 has been generated.

13. Select policy item APPLICATIONS. From the displayed list, select application DWH. Press PF3 to exit the panel.

As a result, the application resource DWH/APL/SYSTEM2 is generated and linked to the application group resource DWHAPG2.

14. Select the RESOURCES policy item. The Select Resources panel with the APL Resource DWH/APL/SYSTEM2 is displayed.

15. Select this APL Resource, if it is not already selected.
16. Change the *Show all Resources* field to YES. As a result, two application group resources are displayed together with the one application resource, as shown in Figure 41.

![Figure 41. Select Resources Panel for DWHAPG2](image)

17. You want to use DWHAPG2 to start the application groups CICS and DB2 on SYSTEM2.

Select CICS on SYSTEM2 (that is, resource CICS/APG/SYSTEM2), DB2 on SYSTEM2 (that is, resource DB2/APG/SYSTEM2) and DWH on SYSTEM2 (see Figure 41).

**Background:**

*Defining Loops in Groups*

The application group CICS is now linked to DWHAPG2. However, when you are using the resource selection panel for CICS, be careful not to link DWHAPG2 to CICS, because this would result in a loop.

![Diagram showing CICS/APG and DWHAPG2/APG](image)

18. Change the field *Show selected Resources only* to YES to check the result. The *Select Resources* panel with one application resource and two application group resources selected is displayed, as shown in Figure 42 on page 63.
Figure 42. Select Resources Panel for DWHAPG2

19. Press PF3 until you return to the *Entry Name Selection* panel.

20. Now create application group DWHAPG3 in the same way as application group DWHAPG2, as described in steps 9 on page 61 to 12 on page 61 and link it to System SYSTEM3.

21. For application group DWHAPG3, you want the same application members as you already defined for DWHAPG2, but with focus on SYSTEM3. For this purpose, you select the COPY policy item and in the subsequent *Select Entry for Copy* panel, you select DWHAPG2.

You should be aware that COPY only copies parts of source-objects data and only parts of the links. For more information, see “Common Policy Items” on page 18.

After copying and pressing PF3, you are now back on the *Policy Selection* panel for DWHAPG3.

22. Select the RESOURCES policy item and link the resources for SYSTEM3. The result should look like the panel shown in Figure 43.

Figure 43. Select Resources Panel for DWHAPG3

Press PF3 until you are back on the *Entry Name Selection* panel.

23. Now you have all components for your Move Group DWHAPG, and you can select DWHAPG on the *Entry Name Selection* panel. Select its RESOURCES policy item.
How to Work with Resources

24. The Select Resources panel for DWHAPG is displayed. 
   Change the Show all Resources field to YES.
   The Select Resources panel with eight APG-Resources is displayed (see Figure 44).

<table>
<thead>
<tr>
<th>COMMANDS</th>
<th>ACTIONS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFGAR55</td>
<td>Select Resources</td>
<td>Row 1 to 8 of 8</td>
</tr>
<tr>
<td>Command ===</td>
<td>SCROLL ===</td>
<td>PAGE</td>
</tr>
</tbody>
</table>

   Entry Type : ApplicationGroup   Entry Name : DWHAPG
   Sysplex    : SYSPLEX1

   Number of selected Resources : 0
   Availability Target : 1 (Move Application Group)
   Show all Resources .......... YES (YES NO)
   Show only Resources with String.

   Action Preference Resource Name
   _____________________________
   CICS/APG/SYSTEM2
   CICS/APG/SYSTEM3
   CICSAPG/APG
   DB2/APG/SYSTEM2
   DB2/APG/SYSTEM3
   DWHAPG2/APG/SYSTEM2
   DWHAPG3/APG/SYSTEM3
   1APG

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

Figure 44. Select Resources Panel for DWHAPG with Valid APG-Resources
Background:

Using Preferences to Define Specific Roles

The Resources panel for MOVE groups looks slightly different than the panels for BASIC groups. It has an additional field (Availability Target) that cannot be modified, and the second column in the scrollable part of the panel is called Preferences.

Preferences are used to define specific roles of the resources linked to a group. You want DWHAPG2 on SYSTEM2 be started first and, if this fails, you want to start DWHAPG3 on SYSTEM3. To define this, start with preference values 700 and 500. They have the following meaning:

700  Resource is started when the group is started. The resource is not stopped unless a very high preference (greater than 950) alternative becomes available. (Use for the resource that normally runs).

500  Resource is started when the higher preference alternative becomes unavailable. It continues running when normal Preference (700) alternatives become available again. It stops at next (scheduled) outage. (Use for the backup resource).

For more details about preference values, refer to “More about Policy Item RESOURCES” on page 139.

The SECON(D)ARY command sets a preference of 400, which has the following meaning:

400  Resource is started when the higher preference alternative becomes unavailable. The difference to preference 500 is that the resource is stopped when the normal Preference (700) alternative becomes available again.

25. Set the preferences: Select the DWHAPG2 group on SYSTEM2 as primary (using action P in the Action column) and the DWHAPG3 group on SYSTEM3 as secondary (using preference value 500 in the Preference column). Figure 45 on page 66 shows the Select Resources panel with the selections made and Figure 46 on page 66 shows the Select Resources panel after you pressed ENTER: with the action P converted to the corresponding numeric value 700.

Note:

You can either set preferences directly in Preference column or use the Action column for important values:

- PRIMARY...(P) - Sets "primary" preference of 700.
- SECONDARY.(D) - Sets "secondary" preference of 400.
How to Work with Resources

Press PF3 to return to the Policy Selection panel for DWHAPG.

26. Select policy item APPLGROUP INFO and change the field *Automatically link Application-Resources* to NO.

27. Now press PF3 until you are back on the Entry Type Selection panel.

Application group DWHAPG is complete now: if a start request is issued against DWHAPG, its primary resource DWHAPG2 is started. If this fails, a start vote is issued against DWHAPG3.

Step 3.6: Create Sysplex Application Group WBSRVAPG

You will create the Sysplex Application Group WBSRVAPG as a SERVER group.
Background:

What is a SERVER Group?

SERVER groups are created when not all but only a certain number of its linked resources need to be available in order to consider the application group as being available.

Thus a SERVER group with an Availability Target of 1 and the preference values set accordingly can behave like a MOVE group.

On the other hand, a SERVER group with Availability Target set to *ALL and all the Preferences of its members set to 700, will act like a BASIC group.

The Web server group WBSRVAPG should run only on SYSTEM1. The WBSRV application and the NET application group are essential parts that must be up. In contrast, the DB2 application group is not necessarily needed.

1. On the Entry Type Selection panel, enter 5 or APG.
   The Entry Name Selection panel is displayed and shows seven application groups.

2. Enter NEW WBSRVAPG at the command line.
   The Define New Entry panel is displayed; the Name field is already filled with WBSRVAPG.

3. The application group WBSRVAPG runs mainly on SYSTEM1. But WBSRVAPG will have some members that run on other systems from the sysplex. Thus you leave the Application Group Type field at “SYSPLEX”.
   WBSRVAPG should start with all of its components, but it is acceptable if DB2 cannot be started. Thus you change the nature of the WBSRVAPG group to “SERVER”.
   On the same panel, in the field Default Preference, change the default preference to 444 as the initial value for all resources that will be linked automatically.
   For resource generation you need an automation name; enter WBSRVAPG into the Automation Name field.
   The input in the Define New Entry panel should now look similar to Figure 47 on page 68.
### How to Work with Resources

![Define New Entry Input Panel for WBSRVAPG](image)

4. Press PF3 to create WBSRVAPG. The Policy Selection panel for WBSRVAPG is displayed with a message indicating that the application group has been created.

5. On the Policy Selection panel select the WHERE USED policy item. The Where Used panel is displayed with sysplex group SYSPLEX1.

6. Select the sysplex group and press PF3. You will see the Command Progress Display popup window telling you that the resource WBSRVAPG/APG has been created.

7. Select policy item APPLICATIONS of WBSRVAPG and link application WBSRV. Press PF3 to return to the Policy Selection panel for WBSRVAPG. You will be notified that the following resources have been generated:

   - WBSRV/APL/SYSTEM1
   - WBSRV/APL/SYSTEM2
   - WBSRV/APL/SYSTEM3

8. Select the RESOURCES policy item of WBSRVAPG to get to the Select Resources panel for WBSRVAPG.

   As you can see, the Preference for the linked resources is the value you entered on the Define New Entry panel for application groups.

   You can use the Preference column to fine-tune the automatic linking of application resources into an application group.

   By setting the Show all Resources field to YES, you get the complete list of (twelve) selectable resources.

   Change the preferences so that the WBSRVAPG application group is started on SYSTEM1 with WBSRV and NET as mandatory components and DB2 as not-mandatory. To achieve this:

   - Set the Availability Target to “2”
   - Select mandatory components (WBSRV/APL/SYSTEM1 and NET/APG) with a preference value of 1000.
   - Deselect WBSRV/APL/SYSTEM2 and WBSRV/APL/SYSTEM3 (action m).
   - Select DB2 (DB2/APG/SYSTEM2 and DB2/APG/SYSTEM3) with a preference value between 501 and 999.
How to Work with Resources

When the application group is started, all selected members with at least Preference 600 will also be started, but members with Preference 1000 are mandatory.

If you change the Show all Resources field to NO, your definitions should look like the sample screen shown in Figure 48.

<table>
<thead>
<tr>
<th>COMMANDS</th>
<th>ACTIONS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0FGARS2</td>
<td>Select Resources</td>
<td>Row 1 to 4 of 4</td>
</tr>
<tr>
<td>Command ==》</td>
<td></td>
<td>SCROLL==》 PAGE</td>
</tr>
</tbody>
</table>

Entry Type : ApplicationGroup  Entry Name : WBSRVAPG
Sysplex : SYSPLEX1

<table>
<thead>
<tr>
<th>Number of selected Resources</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability Target . . . . .</td>
<td>2</td>
</tr>
<tr>
<td>Satisfactory Target . . . . .</td>
<td>(Level at which group is available)</td>
</tr>
<tr>
<td>Show all Resources . . . . .</td>
<td>NO</td>
</tr>
<tr>
<td>Show only Resources with String.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Action</th>
<th>Preference</th>
<th>Resource Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>WBSRV/APL/SYSTEM1</td>
<td></td>
</tr>
<tr>
<td>700</td>
<td>DB2/APG/SYSTEM2</td>
<td></td>
</tr>
<tr>
<td>700</td>
<td>DB2/APG/SYSTEM3</td>
<td></td>
</tr>
</tbody>
</table>

Figure 48. Final Select Resources Panel for WBSRVAPG

9. The WBSRVAPG application group is now complete. Therefore, press PF3 in the Select Resources panel, select policy item APPLGROUP INFO in the Policy Selection panel and once again change the field Automatically link Application-Resources into APG to NO on the upcoming Application Group Information panel.

Now the application group WBSRVAPG is complete: if a start request is issued against WBSRVAPG, at least WBSRV and NET must be successfully started to make WBSRVAPG available.

How to Use Schedules (Service Periods)

Have a look at the scenario described in "A Grouping Scenario" on page 24. You extend this scenario using service periods that automatically control the startup and shutdown of the resources: the data warehouse application is to be available from 9am to 10pm, the web server is to be available from 4am through to 11pm.

Three service periods are needed:
- SWBSRV linked to the WBSRVAPG (web server) application group
- SDWH linked to the DWHAPG (data warehouse) application group
- SDB2DWN linked to the DB2 application group

The SDB2DWN service period should be for the resource to be down 24 hours a day, seven days a week. This will serve to shutdown DB2 and CICS when they are not needed to support an application (WBSRV or DWH or anything else that is dependent upon them).

Schedule Override

The web server needs to be available until midnight. So the operator just needs to override the SWBSRV service period (changing 11pm to 12am). An appropriate action would be the only one required to reverse the override.
How to Use Schedules (Service Periods)

If you do not want to use service periods, you could just issue a high priority request for WBSRVAPG to be available sometime before 11pm. To remove this request at midnight, you could either use INGREQ command with the expiration time parameter, which will automatically remove the request. Or, at 12am, you could use the INGVOTE command to remove this request.

Early Start

At 3am the web server application needs to be started.

The operator simply creates an override for SWBSRV that says to start the web server at 3am and keep it up until 11pm. Everything will then happen automatically. This can be done several hours (or even days) in advance.

If the operator does not want to use service periods to achieve this, he or she just needs to issue a start command against the WBSRVAPG application group and all the required resources will be made available (and kept available) until the operator removes their request (if this is after 4am, the resources will not be stopped as the desired status of the resource will not change. The requested AVAILABLE from the operator will be replaced with a requested AVAILABLE from the schedule).

How to Use Triggers and Events

This chapter contains a scenario on how to define automatic start and stop based either on fixed time intervals (service periods) or on special conditions (events).

Scenario:

Let us assume you have a group of three CICS applications: two application owning regions: one AOR running on SYS1, one AOR running on SYS2 and one terminal only region TOR running on SYS3. These need to be available between 7:00 AM and 6:00 PM during the weekdays and between 8:00 AM and 2:00 PM on Saturdays.

But before the three systems can be started, a backup of some related data sets is required.

You can have NetView trap the successful completion of the backup process. Now the two application owning regions can be started.

Once either of the AORs is up, SA z/OS is able to start the TOR on SYS3.

In emergency situations the systems need to be automatically stopped in a controlled way.

So to enable the automatic startup and shutdown of the systems and applications when required, the system administrator must specify certain conditions and definitions described in the following list. These conditions are checked by the system, and if met, the CICS applications are started. If for any reason some of these applications are already available, then no new startup is triggered by SA z/OS.

To implement the scenario, perform the following steps:

- Define applications AOR and TOR.
- Define a Sysplex Application Group AOR_MOVE of nature MOVE for the sysplex consisting of SYS1 and SYS2. This will generate the resource
How to Use Triggers and Events

AOR_MOVE/APG

- Link application AOR to this MOVE-group. This will generate the resources
  AOR/APL/SYS1
  AOR/APL/SYS2

- Define a System Application Group TOR_GRP of nature BASIC. This will generate
  the resource
  TOR_GRP/APG

- Connect system SYS3 to TOR_GRP and put application TOR into this group. This
  will generate the resource
  TOR/APL/SYS3

- Define a HasParent relationship between TOR/APL/SYS3 and AOR_MOVE/APG (with
  AOR_MOVE/APG being the parent of TOR/APL/SYS3). This will make TOR/APL/SYS3
  available as soon as its parent is available.

- Create application group resource CICS_APL/APG of nature BASIC and let the
  application group resource TOR_GRP/APG and AOR_MOVE/APG be members in this
  group.

- Define an EVENT1: This event, if set, will signal to the system that the backup is
  successfully performed and the applications AOR can be started on their systems
  SYS1 and SYS2. The administrator can have SA z/OS automatically set this
  event by trapping message IEF404I from the backup process. You can set events
  using the INGEVENT command.

  For each event you also define when SA z/OS should unset the event. In the
  scenario, you could specify that EVENT1 should be unset if the applications are
  up.

- Define a trigger TRIGGER1 and link it to the resources AOR/APL/SYS1 and
  AOR/APL/SYS2. This trigger specifies that the linked applications are to be started
  if EVENT1 is set.

  How to define triggers and to determine their condition lists is described in
  “Triggers Policy Object” on page 226.

  How to link a trigger to an application is described in chapter “Policy Items for
  Applications” on page 151 under policy item TRIGGER.

- Define a service period SERVP1 that specifies that the applications should be
  started on weekdays and on Saturdays as specified in the scenario. Link this
  service period to CICS_APL/APG.

  How to define service periods is described in “Service Periods Policy Object” on
  page 223.

Now you are done. The definitions are illustrated in Figure 49 on page 72

1. If the time of day is within the time frame defined by the service period
   SERVP1, SA z/OS will start CICS_APL/APG.

2. This in turn will start TOR_GRP/APG and AOR_MOVE/APG. AOR_MOVE/APG is available,
   as soon as one of its members AOR/APL/SYS1 or AOR/APL/SYS2 is available.
   Availability of these two resources is controlled by event/trigger condition.

3. As soon as AOR_MOVE/APG is available, SA z/OS will start TOR/APL/SYS3 because
   its parent is available.
How to Use User E-T Pairs

This chapter describes how user-specified data, that does not fit into one of the standard entry types (for example, APL, APG, or SYS) can be entered into a policy database and retrieved later on by the automation.

In this example, the data for a job card shall be modified.

**Step 1: Specifying User Data in the Customization Dialog**

Select entry type User E-T Pairs (99) from the Entry Type Selection panel. In the example, assume that an entry JOBCARDS for a set of jobcards has already been created. Select this entry now.

If you select policy item E-T DATA from the Policy Selection panel for the JOBCARDS User E-T Pairs, The UET Entry-Type Selection panel similar to Figure 50 is displayed.

Now enter an E for Edit in the Action column for the User E-T Pairs pair that you want to edit. In the example, you want to edit JOBCARD DEFAULTS. SA z/OS
displays the UET Keyword-Data Specification panel into which you can enter the keywords and user data for each Entry-Type pair data as shown in Figure 51.

A keyword is an identifier for the data that follows it. The set of keyword/data pairs may be null, and each keyword does not have to be unique; that is, each keyword may have multiple sets of data assigned to it.

<table>
<thead>
<tr>
<th>COMMANDS</th>
<th>ACTIONS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFGU0D</td>
<td>UET Keyword-Data Specification</td>
<td>Row 48 from 130</td>
</tr>
</tbody>
</table>

- Entry Type: User E-T Pairs
- PolicyDB Name: DATABASE_NAME
- Entry Name: JOBCARDS
- Enterprise Name: YOUR_ENTERPRISE
- UET Entry: JOBCARD
- UET Type: DEFAULTS
- Action: Keyword/Data(partial)
- JOBNAME
- HAIAOCA3*
- CLASS
- A
- MSGCLASS
- N
- MSGLEVEL
- (1,1)
- REGION
- 4096K

Figure 51. UET Keyword-Data Specification Panel

**Step 2: Access the User Data**

In a NetView session, you can use the command DISPACF to retrieve the data from the automation control file.

You can also retrieve the data using ACFFQRY from within a REXX exec as shown in the following example:

```
Get_keyword: /* name of a subroutine in the REXX ecec */
    Arg keyword .
    len = Length(keyword)
    */ get the data within a pipe */
    'PIPE NetView ACFFQRY JOBCARD DEFAULTS | STEM ALL_DATA.',
    ' ' SEPARATE ' LOCATE 1', len ' keyword','/
    ' STEM KEY.'
If all_data.0 < 1 Then
    Call Terminal_Error 'PIPE 1 Failed'
If all_data.1 <> 'ACFFQRY:0' Then
    Return
If key.0 = 0 Then
    Say 'No Match for 'keyword
Else Do i = 1 to key.0
    Say 'Match! 'key.i
End

Return
```

This method returns the following data in the pipe:

```
| IPSNO
ACFFQRY:0
JOBCARO DEFAULTS
JOBNAME=HAIAOC3*
```

Chapter 4. Scenarios on How to Use the Customization Dialog 73
After this processing, the data is available in the key. stem variable and you can use it for further processing.

### How to Implement Message Processing with User-Defined Data

This section describes how user-defined data is used to automatically execute commands if an application issues a certain message.

**Example:**

The HSM application issues the ARCO909E message that indicates the utilization percentage of the journal data set. You now want to automate the recovery of the HSM journal data set if it is more than 97% full. For this purpose, you want SA z/OS to automatically issue two commands to HSM that will perform the recovery:

- **HOLD ALL**
  - To stop all further activities for HSM.

- **BACKVOL CDS**
  - To create a backup of the full journal data set and allocate a new one.

To implement this scenario, perform the steps described in the following sections.

#### Step 1: How to Specify the User-Defined Data

Select the application for which you want to automate the recovery and then select its policy item MESSAGES/USER DATA. This will lead you to panel **Message Processing** where you enter the message ID ARCO909E as shown in Figure 52.

**Figure 52. Message Processing Panel**
To specify your user-defined data, type USER in the Action column of the Message Processing panel shown in Figure 52 on page 74. Pressing ENTER will lead you to the User Defined Data panel where you specify your input as shown in Figure 53.

![Command Panel](image)

**Figure 53. User Defined Data Panel**

In the REXX exec that you will write in “Step 4: How to Write the REXX Exec” on page 78, you will compare the percentage value returned from the ARC0909E message with the value specified here, and, if it exceeds your specified value, you will cause SA z/OS to issue the commands specified in “Step 3: How to Update the NetView Automation Table” on page 76. If you later decide to start recovery if 95% usage is exceeded, you need not edit your REXX exec, but only your customization dialog input and then rebuild your system operations configuration files.

Pressing PF3 will store your input and bring you back to the Message Processing panel.

**Step 2: How to Specify the Commands**

In the Message Processing panel (Figure 52 on page 74) now enter CMD in the Action column for message ARC0909E. This leads to the CMD Processing panel where you specify your input as shown in Figure 54 on page 76.
Step 3: How to Update the NetView Automation Table

You need to specify a NetView automation table entry to trigger a REXX exec called YOUR_CLIST, which you will create in “Step 4: How to Write the REXX Exec” on page 78.

The AT entry can be specified by using the Automation Processing panel to override the predefined AT entry. To do this, enter OVR in the Action column for message ARC0909E in the Message Processing panel, as shown in Figure 55.

This leads to the Automation Processing panel, as shown in Figure 56 on page 77. This shows the default values for the message.
The definitions that are needed to create an AT entry are as shown in the action fields in Figure 57. Pressing Enter will save your override and the fields are highlighted in a different color, and the NetView AT source field is changed to USER to indicate the override.

Alternatively, you can manually specify this automation table entry either in INGMSGU1 (which is empty, but already included in INGMSG01), or in your own automation table, which then needs to be specified in the AUTOMATION SETUP policy for entry type SYS.

This entry should look like the following:

```
IF MSGID = 'ACR0909E'
THEN EXEC(CMD('YOUR_CLIST') ROUTE(ONE *));
```
Step 4: How to Write the REXX Exec

The value specified on the User Defined Data panel (Figure 53 on page 75) is accessed in the PIPE statement using the ACFFQRY command and stored in the variable limit (1).

To issue the commands specified on the CMD Processing panel (Figure 54 on page 76), the ACFCMD command is used (2).

```rexx
/* REXX */
/* debugging ... */
......
lrc = 0
/* Get the message from the safe */
"GETMLINE txt 1"
svmsgid = msgid()
svjobname = jobname()
Parse Var txt . type . ' IS ABOUT ' usage '%'.

/* This sample recovers just type JRNL */
If type = 'JRNL' Then Exit 0

/* Check the RECOVERY automation flag */
"AOCQRY "svjobname"."svmsgid" RECOVERY"
lrc = rc
Select
  /* Automation allowed */
  When lrc = 0 Then Do
    /* Get TGlobal variable from AOCQRY */
    "GLOBALV GETT SUBSAPPL"
    /* Get the user-data from the ACF */
    "PIPE (STAGESEP | NAME GETUDATA)*", 
    "NETV ACFFQRY "subsappl","svmsgid",
    "| SEPARATE", 
    "| LOCATE 1.6 /LIMIT="/",
    "| EDIT 7.2 1",
    "| VAR limit"
    /* If no limit was defined use default */
    If symbol('limit') = 'LIT' Then limit = '95'
    /* Check if limit is exceeded. If so, issue commands */
    If usage >= limit Then Do
      "ACFCMD MSGTYP="svmsgid"
      2
      lrc = rc
      /* Error occurred? */
      If lrc \= 0 Then Do
        ....
      End
  /* Automation not allowed */
  When lrc < 3 Then Do
    ....
  End
  /* Resource not known to SA */
  When lrc = 3 Then Nop
  /* Error occurred */
  Otherwise Do
    ....
```

How to Implement Message Processing with User-Defined Data
How to Implement Message Processing with User-Defined Data

End
End
Exit lrc
How to Implement Message Processing with User-Defined Data
Chapter 5. Defining Automation Policy

This chapter describes how to define SA z/OS policy objects. They are described in the following sections in the order they appear on the Entry Type Selection panel shown in Figure 58.

Figure 58. Entry Type Selection Panel

This chapter discusses the following policy objects:

- “Enterprise Policy Object” on page 82
- “Group Policy Object” on page 90
- “SubGroup Policy Object” on page 103
- “System Policy Object” on page 105
- “ApplicationGroup Policy Object” on page 125
- “Application Policy Object” on page 145
- “Events Policy Object” on page 221
- “Service Periods Policy Object” on page 223
- “Triggers Policy Object” on page 226
- “Processor Policy Object” on page 229
- “Monitor Resource Policy Object” on page 237
- “Product Automation Policy Object” on page 241
- “Timers Policy Object” on page 256
- “Timeout Settings Policy Object” on page 258
- “Tape Attendance Policy Object” on page 260
- “MVS Component Policy Object” on page 262
- “MVSCOMP Defaults Policy Object” on page 269
- “System Defaults Policy Object” on page 270
- “Application Defaults Policy Object” on page 272
- “Auto Operators Policy Object” on page 273
- “Notify Operators Policy Object” on page 279

© Copyright IBM Corp. 1996, 2004
Enterprise Policy Object

You can define one enterprise for each policy database. An enterprise definition includes:
- Enterprise description
- Operator profile and command parameters for INGSEND
- Processor operations focal point information

Policy Items for an Enterprise

Figure 59 shows the policy items that are displayed when Enterprise is selected from the Entry Type Selection panel.

<table>
<thead>
<tr>
<th>ACTIONS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command ===&gt;</td>
<td>Policy Selection</td>
</tr>
</tbody>
</table>

**Figure 59. Policy Selection Panel for Enterprises**

The policy items for Enterprises are:

**SEND COMMAND OPERS**

This policy item allows you to define default RMTCMD autotasks for operators sending commands to target domains, so that they do not have to remember which target operator ID to use for a particular domain. This gives you more flexibility than the RMTCMD default of taking the task ID of the issuing operator.

These definitions are also used for cross domain command routing to systems that are outside the local sysplex. This is valid for all operator commands supporting the TARGET parameter.

For more information, see “More about Policy Item SEND COMMAND OPERS” on page 83.

**INGSEND PARMS**

This policy item allows you to define default parameters for the INGSEND command.
It also allows you to specify the CRM to be used by INGSEND when communicating with target domains. A wildcard character, *, can be used for NETID. Additionally, NETID is optional and may be left blank. Your last entry in this policy definition will probably be * * ROUTE until you have migrated your entire enterprise to the current SA z/OS release and have changed your INGSEND usage to be based entirely on RMTCMD.

RMTCMD-based INGSEND exploits NetView PIPES for some invocations. You can define a value to be used on the CORRWAIT PIPE stage if you need a shorter or longer CORRWAIT value than the SA z/OS default of ten seconds. CORRWAIT only applies if your command routing mechanism is RMTCMD.

For more information refer to “More about Policy Item INGSEND PARMS” on page 85.

**PROCESSOR OPS INFO**

This policy item allows you to certain enterprise-wide processor operations data. For more information, refer to “More about Policy Item PROCESSOR OPS INFO” on page 86.

**AUTO MSG CLASSES**

This policy item allows you to define your own message classes. Message classes that are defined here can be assigned to operators by using policy NOTIFY OPERATORS of the entry type NFY. For more information, see “More about Policy Item AUTO MSG CLASSES” on page 87.

**SYMBOL DESCRIPTION**

This policy item allows you to enter system automation symbol descriptions. For more information, see “More about Policy Item SYMBOL DESCRIPTION” on page 88.

**MPF DEFINITIONS**

This policy item allows you to define MPF header and trailer information. For more information, see “More about Policy Item MPF DEFINITIONS” on page 89.

**More about Policy Item SEND COMMAND OPERS**

Figure 60 on page 84 shows the panel that is displayed when the SEND COMMAND OPERS policy item is selected.
Enterprise Policy Object

Figure 60. Operator Profile for Sending Commands Panel

This policy item allows you to define default RMTCMD autotasks for operators sending commands to target domains, so that they do not have to remember which target operator ID to use for a particular domain. This gives you more flexibility than the RMTCMD default of taking the task ID of the issuing operator.

These definitions are also used for cross domain command routing to systems that are outside the local sysplex. This is valid for all operator commands supporting the TARGET parameter.

This policy item is intended to help you migrate your cross domain command routing to RMTCMD, by allowing you to use any naming scheme for RMTCMD autotasks. By default, RMTCMD uses the operator ID of the invoking task to determine the RMTCMD autotask to which to issue a command. Use the Operator Profile for Sending Commands panel to define an association between an origin operator and a target RMTCMD autotask. The INGSSEND command uses predefined RMTCMD associations to resolve the OPER keyword (target operator ID to execute the routed command), if it has not been specified on an invocation. If INGSSEND does not find an RMTCMD association for the invoking operator to the target domain in the automation control file, the RMTCMD default will be used to resolve the target operator ID. For more information on INGSSEND, refer to System Automation for z/OS User’s Guide and System Automation for z/OS Operator’s Commands. For more information on RMTCMD, refer to Tivoli NetView for OS/390 Command Reference Volume 1.

The definitions you enter here apply to all systems in the enterprise. They are included in the automation control file built for each system. At run time, INGSSEND on each origin domain will use only the information intended for that particular origin Netid.Domain.

On the Operator Profile for Sending Commands panel (Figure 60), the following fields are defined for both origin and target:

**NETID**

The Network ID is optional. The maximum length is eight characters. The first character must be alphanumeric. The rest must be either alphanumeric or national characters.
DOMA†N
The NetView Domain ID is optional. The maximum length is five characters. The first character must be alphanumeric. The rest must be either alphanumeric or national characters.

OPER
The Origin Operator ID is required. The maximum length is eight characters.

The Target Operator ID is optional. If entered, it must be eight characters or less. If the field is left blank, it will resolve to the ORIGIN OPER value.

The operator ID you define here should have matching OPERATOR statements in the relevant DSIOPF for each origin and target domain.

Note:
For TARGET OPER, PPT or target domain concatenated with PPT is not allowed.

A wildcard character, *, can be used for NETID and DOMAIN. All fields are optional except for ORIGIN OPER. TARGET OPER will resolve to ORIGIN OPER if it is not entered. This is the NetView default.

More about Policy Item INGSEND PARMS
If you select policy item INGSEND PARMS from the Policy Selection panel for enterprises, a panel similar to Figure 61 is shown.

The System Operations INGSEND command is used to route commands across domains. Only one command routing mechanism is supported: the NetView RMTCMD command.

To ensure compatibility with systems running SA OS/390 2.2 or earlier the specification for the NetView ROUTE command is still allowed.

RMTCMD-based INGSEND invocations can involve usage of the NetView PIPE command. Since RMTCMD produces asynchronous command output, it is
necessary to use the CORRWAIT Pipe stage to trap the asynchronous messages. You can customize the CORRWAIT time that INGSEND uses per target domain.

Use wildcards for target netid and domain to specify the predominant CRM in your enterprise.

**TARGET NETID**
The Target Network ID is optional and, if entered, must be eight or less characters. The first character must be alphanumeric. The rest must be either alphanumeric or national characters. If left blank, it will default to ‘*’.

| Note: | Wildcard value of ‘*’ is allowed. This field can only be blank or ‘*’ for a CRM of ROUTE. |

**TARGET DOMAIN**
The Target Domain is optional and, if entered, must be five or less characters. The first character must be alphanumeric. The rest must be either alphanumeric or national characters. If left blank, it will default to ‘*’.

| Note: | Wildcard value of ‘*’ is allowed. |

**CRM**
The Command Routing Mechanism is optional. This can be either RMTCMD or ROUTE. If left blank, it will default to RMTCMD.

| Note: | The value of this field will only be interpreted on systems running SA OS/390 2.2 or earlier. For newer SA z/OS releases the value is ignored as only RMTCMD is supported. |

**CORRWAIT**
The corrwait value is optional. This value is used by INGSEND for the NetView PIPE CORRWAIT stage. If left blank, it will default to 10 seconds.

| Note: | This field is only allowed for a CRM of RMTCMD. |

**More about Policy Item PROCESSOR OPS INFO**
A panel as shown in Figure 62 on page 87 is displayed if you select policy item PROCESSOR OPS INFO from the Policy Selection panel for enterprises.
Figure 62. Processor Operations Information Panel

The panel contains the following fields:

**Focal point name (primary or backup)**

Enter the NetView domain name of the NetView that is the name of the focal point for your SA z/OS configuration. Names can contain 5 characters, starting with an alphabetic character, #, $, or @. This field is required if you are using SA z/OS processor operations functions. The customization dialog does not verify your entry.

**Control file**

Identifies the data set containing configuration information by processor operations under NetView.

**Control file log**

Identifies the data set containing information about errors in the configuration definition. When you generate a control file, this error file contains the errors or messages created during the generation procedure. Check this file after each control file generation.

More about Policy Item AUTO MSG CLASSES

A panel as shown in Figure 63 on page 88 is displayed if you select policy item AUTO MSG CLASSES on the Policy Selection panel for enterprises.
This panel defines the user message classes that are available to be assigned to notify operators by using policy NOTIFY OPERATORS of entry type NFY. The Auto Msg Classes policy object allows you to define your own message classes.

You can also enter a short description of each message class. Message classes must consist of an alpha and a numeric character. All double numeric message classes are reserved for use by IBM. See “Notify Operators Policy Object” on page 279 for more information on the message classes that SA z/OS defines.

Notes:
1. No duplicate message classes are allowed.
2. Message classes used within any Notify Operators cannot be deleted.

For details about the definition restrictions of message classes, refer to the online help of SA z/OS.

More about Policy Item SYMBOL DESCRIPTION

If you select policy item SYMBOL DESCRIPTION from the Policy Selection panel for enterprises, a panel similar to Figure 64 on page 89 is shown.
This panel allows you to enter or update a description for all system automation symbols.

**Description**

This allows you to specify a description for each system automation symbol. This description will be shown for the system automation symbols on all systems in the policy DB.

**More about Policy Item MPF DEFINITIONS**

If you select policy item MPF DEFINITIONS from the Policy Selection panel for enterprises, a panel similar to Figure 65 is shown.

This panel allows you to enter or update definitions which will be included in the MPFLSTSA member during MPF build.

**MPF Header Lines**

Allows you to specify up to 3 MPF statements that will be included at the top of the MPFLSTSA member, for example:

`.NO_ENTRY,SUP(NO),RETAIN(1,CE),AUTO(YES)`
Enterprise Policy Object

or

.DEFAULT,SUP(YES),RETAIN(I,CE),AUTO(NO)

MPF Trailer Lines

Allows you to specify up to 3 MPF statements that will be included at the bottom of the MPFLSTSA member.

Note:

Changes in MPF header and trailer lines are taken into account during the next build of NetView automation tables when the MPF member is also rebuilt.

Group Policy Object

A group is the largest subdivision of an SA z/OS enterprise. It is a set of systems, or subgroups, or both, that you want to associate with one another for purposes of automation, monitoring on the NMC workstation or that you want to use with the SA z/OS processor operations functions. An SA z/OS enterprise may have many groups, each consisting of one or more systems and subgroups. You define a group by assigning a name to it and by selecting the member systems and subgroups.

The status of a group depends on the status of the systems or applications of which it is comprised. The status of a group is indicated by appropriate colors on the NMC workstation. You specify the number of systems or applications of a group that must be in an unsatisfactory state to cause SA z/OS to change the color of a group in the graphic display.

Creating a New Group

How to create new policy objects in general, is described in "How to Create New Policy Objects" on page 14.

Groups can be defined as either standard groups or sysplex groups. A standard group contains any combination of systems and subgroups. A sysplex group contains only systems or subgroups that belong to the same XCF group. For a sysplex group, there are additional policy items to describe certain sysplex-specific resources. A sysplex group may also have Sysplex Application Groups connected to it. Sysplex Application Groups are groups of applications that can reside on more than one system in the sysplex. A group must be defined as either a sysplex group or a standard group when it is created. Figure 66 on page 91 is an example of the Define New Entry panel for groups.

A system may belong to many standard groups, but can belong to only one sysplex group.

If you want to monitor your newly created sysplex group on the NMC workstation, you need to specify an entry for this sysplex in the INGTOPOF file (see System Automation for z/OS Planning and Installation).
The unique fields are:

**Name**  This is the name of the entry as used by the SA z/OS dialogs. Each Group must be assigned a unique, non-blank name. The name must be unique among the names of all groups, subgroups, systems (including both the entry names and the Image/ProcOps names), and processors in the current policy database.

**Group type**  Use this field to specify the type of group you want to create. You can create either a Sysplex group if the group is to be associated with a sysplex, or a Standard group. Each type of group has specific policy items and restrictions associated with it. The value you enter here cannot be changed after the group is created.

   The available values are:

   **SYSPLEX**  If the Group is being used to represent a logical sysplex (a group of systems within the same XCF group ID).

   **STANDARD**  For all other Groups, for example, to represent a physical sysplex (a group of systems within different XCF group IDs, such as a GDPS site).

**ProcOps Commands**  Indicates whether processor operations commands issued against this group are to be accepted for processing. The default is NO. See System Automation for z/OS Operator’s Commands for information about the processor operations commands.

**Note:**  If you specify YES, then a processor operations command directed to that group will affect all systems in that group, even if those systems are associated with a subgroup of the group and that subgroup is *not* enabled for processor operations commands.
Policy Items for Standard Groups

Figure 67 shows the policy items that are displayed for a standard group.

The unique policy items for groups are:

GROUP INFO
This displays the type of group that was specified when the group was created. This can be either standard or sysplex. It also displays and allows you to update the current value of the processor operations commands indicator.

For more information, see “More about Policy Item GROUP INFO” on page 94.

SUBGROUPS
When you select this policy item, a list of all subgroups in the SA z/OS enterprise is displayed. Those subgroups that are related to the current group are marked with SELECT. You can add subgroups to the group, or remove subgroups from the group.

For more information, see “More about Policy Item SUBGROUPS” on page 94.

SYSTEMS
When you select this policy item, a list of all systems in the SA z/OS enterprise is displayed, indicating by SELECTED which are related to this group. You can add systems to the group, or remove systems from the group.

For more information, see “More about Policy Item SYSTEMS” on page 95.

Policy Items for Sysplex Groups

You define a sysplex group when creating a new group object by specifying “sysplex” in the group type field of the Define New Entry panel. There are certain restrictions for sysplex groups:

- A system can be linked to only one sysplex group. The same system can be linked to other standard groups. This applies if the system is linked directly to the system group, or indirectly through a subgroup.
- A subgroup may be linked to only one sysplex group.
Figure 68 is an example of the Policy Selection panel for Sysplex groups.

The policy items for sysplex groups are the same as those for standard groups with the following additions:

**SYSPLEX**

This policy item allows you to describe certain resources associated with the sysplex. For more information, see “More about Policy Item SYSPLEX” on page 95.

**NMC DEFINITIONS**

This policy item allows you to make NMC definitions for the primary and secondary NMC Focal Point Domains. For more information, see “More about Policy Item NMC DEFINITIONS” on page 98.

**APPLICATION GROUPS**

This policy item displays a list of all Sysplex Application Groups, indicating which application groups are selected for this sysplex. For more information, see “More about Policy Item APPLICATION GROUPS” on page 99.

**LOCAL PAGE DATA SET**

With this policy item you can predefine local page data sets that are used in recovery situations to prevent auxiliary storage shortage outages. For more information, see “More about Policy Item LOCAL PAGE DATA SET” on page 99.

**JOB DEFINITIONS**

With this policy item you can define the jobs that can or cannot be cancelled (KEEP/CANCEL), in case the job is responsible for the shortage condition. For more information, see “More about Policy Item JOB DEFINITIONS” on page 100.

**JOB/ASID DEFINITIONS**

With this policy item you can define the jobs and address space IDs that
can or cannot be cancelled. You can specify KEEP if the job must not be cancelled, DUMP to dump the job after the wait time has expired, or NODUMP to suppress the dump before the job is cancelled.

For more information, see “More about Policy Item JOB/ASID DEFINITIONS” on page 101.

**RESOURCE DEFINITIONS**

With this policy item you can define the major and minor resource names together with their wait time (30-999 seconds) after which an ENQ is treated as a long running ENQ.

For more information, see “More about Policy Item RESOURCE DEFINITIONS” on page 102.

**IEADMCxx SYMBOLS**

With this policy item you can define the IEADMCxx symbols together with their substitution value for selective jobs or address space IDs. For more information, see “More about Policy Item IEADMCxx SYMBOLS” on page 103.

### More about Policy Item GROUP INFO

A panel as shown in Figure 69 is displayed if you select policy item GROUP INFO on the Policy Selection panel for standard groups. (If the group type was defined as sysplex, this is reflected in the panel.)

```
COMMANDS     HELP
------------- -------------------------------
AOFGGPA0     Group Information
Command ==>                                          

Entry Type : Group PolicyDB Name : DATABASE NAME
Entry Name : STANDARD_GROUP Enterprise Name : YOUR_ENTERPRISE

The following field was specified when the Group was defined and cannot be modified:
Group Type : STANDARD

ProcOps Commands . . . . NO (YES NO)
```

**Figure 69. Group Information Panel for Standard Groups**

This panel displays the following information:

**Group Type**

This field cannot be changed; it was set when this group was created. For more details, see “Group Type” on page 91.

**ProcOps Commands**

For more details, see “ProcOps Commands” on page 91.

### More about Policy Item SUBGROUPS

A panel as shown in Figure 70 on page 95 is displayed if you select policy item SUBGROUPS on the Policy Selection panel for standard groups. (If the group type was defined as sysplex, this is reflected in the panel.)
This panel allows you to manage the links between the current Group entry and the SubGroups entries. From this panel you can:

- Add a link between the entries with the ADD (A) command in the Action column.
- Remove a link between the entries with the REMOVE (M) command in the Action column.

More about Policy Item SYSTEMS

A panel as shown in Figure 71 is displayed if you select policy item SYSTEMS on the Policy Selection panel for standard groups. (If the group type was defined as sysplex, this is reflected in the panel.)

This panel allows you to manage the links between the current Group entry and the Systems entries. From this panel you can:

- Add a link between the entries with the ADD (A) command in the Action column.
- Remove a link between the entries with the REMOVE (M) command in the Action column.

More about Policy Item SYSPLEX

A panel as shown in Figure 72 on page 96 is displayed if you select policy item SYSPLEX on the Policy Selection panel for sysplexes.
Group Policy Object

The fields on the *Sysplex Policy Definition* panel are as follows:

**Sysplex Name**
Defines the name of the sysplex.

**Sysplex Timer Monitoring**
If you have one or more sysplex timers in your sysplex and you want them monitored by the NMC workstation, enter YES and specify the number of monitored sysplex timers in the next field. Enter NO, if you do not want monitoring. This is the default.

**Number Monitored Sysplex Timers**
Enter ‘2’ if you are using the sysplex timer with the Expanded Availability Option and you want to have two sysplex timers monitored by the NMC workstation. Enter ‘1’, if you only want one sysplex timer monitored.

**Temporary Data Set HLQ**
Defines the high level qualifier to be used to assemble a data set name for allocating temporary data sets needed by programs running as started tasks. The qualifier can consist of up to 17 characters according to the z/OS data set naming rules (hlq1.hlq2.hlq3). Specify the name without quotes.

**Started Task Job Name**
Defines the job name to be used for programs running as started tasks. The name can consist of up to 8 characters according to the z/OS job naming rules.

**Couple Data Set HLQ**
Specify the high level qualifier to be used for the allocation of alternate couple data sets (CDS). The qualifier can consist of up to three parts according to the z/OS data set naming rules. When allocating an alternate...
CDS, this qualifier is appended with the type (for example, SYSPLEX) and CDS\(n\), where \(n\) is a sequence number.

**CDS type - Alternate volumes - Desired monitoring**

Specify the types of couple data sets to be used, and the monitoring to be done.

The types of supported couple data sets are:

- **Sysplex**
  Contains the policy and status for basic sysplex functions and points to the other couple data sets.

- **ARM**
  Contains the policy for automatic restart management.

- **CFRM**
  Contains the policy for coupling facility resource management.

- **LOGR**
  Contains an inventory for the sysplex logger function.

- **SFM**
  Contains the policy for sysplex failure management.

- **WLM**
  Contains the policy for the workload manager.

In the column **Alternate volumes** you can enter up to 8 volume names for allocating alternate couple data sets for types Sysplex, ARM, CFRM, LOGR and SFM. For WLM, no alternate volumes can be specified. The names must be in accordance to the z/OS volume naming rules. In case of allocating alternate couple data sets, the allocation starts on the first specified volume and continues in the specified order.

The information you enter in column **Desired monitoring**, determines when the NMC workstation of SA z/OS displays the associated icon for a couple data set as satisfactory or unsatisfactory. For example, if you have a couple data set QUAL1.ARM.CDS0, which you want to be monitored as PRIMARY, but currently this data set is not available because the operator had to switch to an alternate ARM CDS, then the icon for QUAL1.ARM.CDS0 is marked as being unsatisfactory (colored red).

You specify the monitoring that is to be done for each type of couple data set as follows:

- **ALTERNATE**
  Indicates that you want to monitor both the primary and the alternate data set. In this case, when either the primary or the alternate fails, the related icon will turn red.

- **PRIMARY**
  Indicates that you want to monitor only the primary couple data set. In this case, there is only one icon (for the primary CDS) on the NMC workstation and the operator will only see when the primary itself is unavailable, no matter whether an alternate CDS is available to take over.

- **NONE**
  Indicates that none of the couple data sets are monitored. This is the default for all types of CDSs, except for the sysplex CDS. The default for the sysplex CDS is PRIMARY.

**Logical Sysplex Group ID**
A logical sysplex group ID may be specified to indicate that a subset of the
members of an actual z/OS sysplex is defined in this sysplex group. If
specified, the ID may contain 1 or 2 characters. Valid characters are A-Z,
0-9, and the national characters ($, # and @).

If a logical sysplex group ID is specified, it will be prefixed with the string
INGXSG to construct the XCF group name used for cross system
synchronization, for example, INGXSGxy.

If no logical sysplex group ID is specified, the default group name
INGXSG will be used.

This group needs to be associated to an appropriate XCF transport class by
the person responsible for the sysplex setup (refer to the manual z/OS MVS
Setting Up a Sysplex).

More about Policy Item NMC DEFINITIONS

A panel as shown in Figure 73 is displayed if you select policy item NMC
DEFINITIONS on the Policy Selection panel for sysplexes.

Figure 73. NMC Definitions Panel for Sysplex Groups

Anchors are special objects that exist in RODM. They are displayed as icons on
the NMC topology client. They are not automated by the automation manager. They
play the role of major resources and serve these purposes:

- Anchors are used to display and group minor resources that derive from events
  such as WTORs and tape mount requests.
- Lost&Found anchors are used to display minor resources for which either the
  specified major resource or sysplex anchor does not exist in RODM.
Couple data sets (CDSs), coupling facilities (CFs) and external timer references (ETRs) are displayed as minor resources and are attached to the sysplex anchor.

It is possible to attach a minor resource to a user-defined anchor with the INGPOST command. The major resource specified to INGPOST can be a regular resource (for example, APL, APG, SYS, etc.) or an anchor.

Heartbeats (minor resources) are attached to a heartbeat anchor.

Specify the name of an anchor on this panel for these cases:
- To enable the display of WTORS and tape mount requests in the NMC topology client, the anchor names that must be specified are: WTOR and TAPE
- To define a sysplex anchor for ‘homeless’ minor resources or to display CDSs, CFs, and ETRs, the name of the sysplex anchor is: SYSPLEX
- To display minor resources that are created with INGPOST. The name used for the resource= parameter should be the name of the anchor that is specified here. For example, with resource=ANCHOR/MJR/CRITMSG the name of the anchor is: CRITMSG

Do not specify an anchor on this panel for:
- Heartbeats. The anchor is automatically created by SA z/OS.
- Lost&Found anchors.

**More about Policy Item APPLICATION GROUPS**

A panel as shown in Figure 74 is displayed if you select policy item APPLICATION GROUPS on the Policy Selection panel for sysplex groups.

This panel allows you to manage the links between the current Sysplex entry and the Sysplex AppGroups entries.

**More about Policy Item LOCAL PAGE DATA SET**

With the automation of local page data sets, SA z/OS prevents auxiliary storage shortage outages, by dynamically allocating spare local page data sets when needed.
Group Policy Object

![Figure 75. Local Page Data Set Recovery](image)

Fill in the following fields:

**Page Data Set HLQ**

Defines a high level qualifier which is used to assemble a data set name for creating and allocating a page data. The qualifier can consist of up to 23 characters and must follow the naming conventions of MVS data sets (hlq.hlq2.hlq3). The high level qualifier must point to the master catalog and must not be SMS managed.

**Number of Cylinders**

Defines the maximum number of cylinders (100 to 999) used for the dynamic allocation of a local page data set. The default value is 400. Note, that on a 3390 DASD, 100 cylinders are adequate to 70 MB and the formatting process takes approximately 20 seconds for this amount of space.

**Page Data Set Volumes**

Defines a list of volume IDs which are eligible when the automation is creating and allocating a new page data set. The volumes must be shared by all systems in the sysplex.

**Data Set Names for Spare Page Data Sets**

Defines the pre-formatted spare page data set to be used in the recovery situation. Specify the names without quotes. You may include system symbols and system automation symbols (see "Assigning System Automation Symbols (AOCCLONE)" on page 42). The data sets must be allocated on a volume shared by all systems in the sysplex and must be catalogued in the master catalog. In case the systems in the sysplex do not share the master catalog, the page data set must be re-catalogued on each system except the one which created the data set using the following IDCAMS statement:

```
DEF PAGESPACE(NAME(ipdsn) VOLUME(volume) RECATALOG)
```

**More about Policy Item JOB DEFINITIONS**

You can customize the automation to cancel or keep jobs that are blocking a resource.
Job Name

Defines the jobs which can or cannot be cancelled in case they are one of those jobs responsible for the shortage condition.

Handling

Specify KEEP if the job cannot be cancelled. Otherwise specify CANCEL. You can also specify a wildcard character (*). You must define a default with Job Name=* and either KEEP if all jobs not explicitly defined cannot be cancelled, or CANCEL if all jobs can be cancelled.

More about Policy Item JOB/ASID DEFINITIONS

If you want to automate long running ENQs, you can specify dump options, a title for each dump being taken, and job names or address space IDs of the jobs that are kept or cancelled.
Group Policy Object

Fill in the following fields:

**Dump Options**
Defines the DUMP option that is used for the SDATA parameter on the dump command. The following options are allowed: ALLNUC, COUPLE, CSA, GRSQ, LPA, LSQA, NOPSA, NOSQA, NOSUM, NUC, PSA, RGN, SERVERS, SQA, SUM, SWA, TRT, WLM and XESDATA. This value applies only to jobs or ASIDs defined with DUMP in the following list.

**Dump Title**
Defines the title of each dump taken with the default dump options.

**Job Name/ASID**
Defines the jobs and address space IDs that can or cannot be cancelled in case they are one of those responsible for the long running ENQ.
- Specify KEEP if the job or ASID cannot be cancelled.
- Specify NODUMP if the job or ASID can be cancelled without a dump.
- Specify DUMP if the job or ASID can be cancelled with a dump using the default dump options, otherwise specify the dump IDs if the job or ASID can be cancelled using the IEADMCxx PARMLIB members.

Specification of a trailing wildcard (*) is allowed. You must define a default with Job name/ASID=* together with the handling which will be appropriate for all jobs and ASIDs not explicitly defined.

**More about Policy Item RESOURCE DEFINITIONS**
You can specify which resources you want to monitor and define a value for the maximum time a job can lock a resource while other jobs are waiting for the resource. If the amount of time is exceeded, the recovery takes place.

---

**Figure 78. Long Running ENQ Resource Definitions**

**Major Resource Name**
Defines the major resource name of the resource to be checked for long running enqueues.

**Minor Resource Name**
Defines the minor resource name of the resource to be checked for long running enqueues.
More about Policy Item IEADMCxx SYMBOLS

**COMMANDS HELP**
AOFGENQS | Long Running ENQ IEADMCxx Symbols | Row 1 to 9 of 20
---|---|---
Entry Type: Group | Policy DB Name: DATABASE_NAME
Entry Name: SYSPL1 | Enterprise Name: YOUR_ENTERPRISE

Define the IEADMCxx symbols together with their substitution value for
selective jobs or address space IDs.

**Job Name/ASID**
Defines the jobs and address space IDs together with their symbols and
their substitution value for use with the dump command parmlib members
IEADMCxx. You can specify a trailing wildcard character (*).

**Symbol and Value**
Defines the symbol names and their substitution value in the format
&symbol.='value'. Valid characters for the symbol are A-Z, 0-9, @, # and $.
The value must be enclosed in single quotes and its length must not be
longer than the symbol name.

**SubGroup Policy Object**

A subgroup is an intermediary grouping of the systems within a group. You define
a subgroup by selecting the member systems and assigning a name to the
subgroup. You do not have to define subgroups, but it may be useful to do so for a
large enterprise. For example, a sysplex defined as a group may have subgroups
consisting of systems running on one or more S/390 microprocessors within the
sysplex so they can be managed separately from systems running on the other
CPC types. Another option is to form subgroups based on system type, for
example, all VM systems in one subgroup.

A system in a sysplex can belong to more than one subgroup within the sysplex
group.

**Creating a New SubGroup**

How to create new policy objects in general, is described in “How to Create New
Policy Objects” on page 14.
SubGroup Policy Object

Figure 80 is an example of the Define New Entry panel for subgroups.

![Define New Entry Panel for a Subgroup](image)

The unique field on this panel is:

**ProcOps Commands**

Indicates whether processor operations commands issued against this subgroup are to be accepted for processing. The default is NO.

---

**Note:**

If a group with which this subgroup is associated is enabled for processor operations commands, then a processor operations command directed to that group will affect all systems in the group even if this subgroup is not enabled for processor operations commands.

---

**Policy Items for SubGroups**

Figure 81 shows the policy items that are displayed when a subgroup is selected from the Entry Name Selection panel.

![Policy Selection Panel for Subgroups](image)

The unique policy items for subgroups are:

**SUBGROUP INFO**

When you select this policy item, information about subgroups is displayed for verification and updating.
When you select this policy item, a list of all systems in the SA z/OS enterprise is displayed, indicating those systems that are related to this subgroup. You can add systems to the subgroup, or remove systems from the subgroup.

System Policy Object

A system is any host system in the SA z/OS enterprise. The operating systems can be either MVS or non-MVS. For system operations, the system must be MVS. For processor operations functions, the system can be MVS, VM, VSE, LINUX, or CF (coupling facility). A system definition includes:

- The operating system type
- Processor operations information
- Details of which IBM automation products are operating on the system
- Member application groups
- NetView and z/OS-related information
- Automation policy for the system (optional)

The information you supply here will determine what automation policy set objects are included in your system and how your system is automated.

Creating a New System

How to create new policy objects in general, is described in “How to Create New Policy Objects” on page 14.

Figure 82 shows the panel that is displayed when you create a new system from the Entry Name Selection panel for systems.

If you want to monitor your newly created system as a standalone system on the NMC workstation, you need to specify an entry for this sysplex in the INGTOPOF file (see System Automation for z/OS Planning and Installation).

<table>
<thead>
<tr>
<th>COMMANDS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFGDNZ</td>
<td>Define New Entry</td>
</tr>
<tr>
<td>Command ===</td>
<td>More:</td>
</tr>
</tbody>
</table>

Define new entry of type System

Name . . . . . . . . . . . . . . . . . .

Operating system . . . MVS (MVS VM VSE LINUX CF)
MVS system name . . . . . . . . . . . . (MVS systems only)
Image/ProcOps name . . .

Specify information for NMC Focal Point Communication (MVS systems only):

Heartbeat Interval . . . 5 (1 - 60 minutes)
Missing Heartbeat Delay . . 30 (1 - 3600 seconds)

Short description . .
Long description . .
Long description . .
Long description . .
Long description . .

Figure 82. Define New Entry Panel for a System

The fields are:
System Policy Object

Name  This required name gives a unique name to the system you are defining. Valid characters are any alphabetic or numeric character or any of @, #, $, ?, or _ (underscore), with the first character being alphabetic, numeric, or @, #, or $. The entry name must be unique among the names of all groups, subgroups, systems, or processors, defined in the current policy database. It must also be unique among processor operations names, with the exception that this system’s processor operations name may match this system’s entry name.

Operating system  A required identifier of the operating system to be run on the system. The allowed values are MVS, VM, VSE, LINUX, or CF. The supported default is MVS.

CF indicates the coupling facility control code. LINUX identifies the Linux for zSeries operating system.

MVS SYSNAME  Required for MVS systems, this value must match the system name (SYSID) as specified in the IEASYSxx member of SYS1.PARMLIB. Valid characters are any alphabetic or numeric character or any of @, #, or $.

Note:  The first three characters of MVS SYSNAME must not be A0F because these are reserved for internal SA z/OS processing.

To support an alternate configuration of a system, special notation with a trailing ‘plus-sign’ is allowed. This trailing character will be removed when the automation files are built. If you are going to enable automation for IXC102A/IXC402D the MVS SYSNAME must not start with a numeric character. For more information see Automation for z/OS: System Automation for z/OS V2R3.0 Defining Automation Policy.

Image/ProcOps name  Specifies the name of the system for hardware related functions, for example, processor operations functions.

Like the system’s Entry name, this name must be unique among the names of all groups, subgroups, systems, processors and image names defined in the current policy database. For MVS systems the name is defaulted to the MVS SYSNAME but can be changed.

Valid characters are any alphabetic or numeric character or any of @, # or $. The name may start with a numeric character except for systems of type CF.

Heartbeat Interval  This specifies the time interval at which the SA z/OS system operations functions sends a generic alert from this system to the NMC focal point system. The SA z/OS system operations functions on the NMC focal point system use heartbeat alerts to verify the status forwarding path from each sysplex or single remote system. For more information see “The SA z/OS Heartbeat Function” on page 107. The heartbeat interval is a required field in the range 1 to 60 minutes that defaults to 5 minutes. Heartbeat information is relevant for z/OS systems only.

The smaller the heartbeat interval, the faster SA z/OS can detect a status forwarding path failure. For example, if you define an interval of two
minutes then, on average, SA z/OS will take one minute, plus the missing heartbeat delay time, to detect a failure in the status forwarding path.

**Missing Heartbeat Delay**

This specifies the time that the SA z/OS system operations functions on the NMC focal point system will wait after a heartbeat was expected. If this time expires without receiving the heartbeat, the system operations functions will begin missing heartbeat processing for the remote system. The field is provided as a way to buffer against temporary network delays. For more information see "The SA z/OS Heartbeat Function." The missing heartbeat delay is a required field in the range 1 to 3600 seconds that defaults to 30 seconds. Heartbeat information is relevant for z/OS systems only.

The smaller the missing heartbeat delay, the more sensitive SA z/OS is to temporary status forwarding path delays. The longer the delay, the slower SA z/OS will be to react to a missing heartbeat. Again, you will have to balance these two considerations.

- Consideration must be given to the number of systems and subsystems being monitored in your enterprise. The larger the enterprise being monitored, the larger the delay should be. This is to ensure that at resync time with high status forwarding to the focal point, any delays due to this do not cause unnecessary missing heartbeat processing.

**The SA z/OS Heartbeat Function**

The heartbeat function validates the path between remote systems and the focal point system. This path is known as the status forwarding path.

SA z/OS validates the status forwarding path by sending an alert, called the heartbeat alert, from each single remote system or from a designated system in the sysplex at a regular interval.

If the heartbeat is missing from the designated sysplex system, the SA z/OS topology manager tries to connect to another system in the sysplex. Only if all systems are not responding, SA z/OS invokes the missing heartbeat processing.

SA z/OS on the NMC focal point system expects heartbeats from each remote system at the interval defined for each system, but it expects a heartbeat only from one system in the sysplex. This interval is known as the heartbeat interval.

Receiving a heartbeat on the NMC focal point system indicates that the status forwarding path from the remote system is up, and hence status for resources associated with that system can be considered to be current.

If SA z/OS on the NMC focal point system does not detect an expected heartbeat, it waits to allow for temporary network slowdowns, or any other temporary problems in the status forwarding path that may have delayed the heartbeat. The wait period is known as the missing heartbeat delay. When the missing heartbeat delay time expires, (or for a sysplex, when the missing heartbeat delay time expires for all systems in the sysplex) SA z/OS invokes its missing heartbeat processing for the remote system or the sysplex and changes the status of all associated resources to Unknown because SA z/OS is no longer able to receive status updates for those resources.

When setting the missing heartbeat delay period, consideration must be given to the number of sysplexes or single systems, and subsystems being monitored in your enterprise. The larger the enterprise being monitored, the larger the delay should be.
System Policy Object

When SA z/OS detects that heartbeats from a remote system have resumed, it resynchronizes the NMC workstation status displayed for resources associated with the system by requesting that the remote system should resend all status information.

You can define both the heartbeat interval and the missing heartbeat delay for each MVS system in your enterprise.

Policy Items for Systems

Figure 83 on page 109 shows the z/OS policy items that are displayed when an MVS system is selected from the Entry Name Selection panel.

Notes:

- Not all policies are displayed on one screen. Scroll down to see the second part of this panel.
- Several of the policy items available from this panel allow you to select and manage the links between the system and a variety of entry types. The panels associated with these policy items are all very similar and straightforward to use and are not covered in this section.
Figure 83. Policy Selection Panel for Systems

The policy items for systems are:

**SYSTEM INFO**

This displays and allows you to update the operating system type for this system definition.

You can also use this option to change the heartbeat interval and the missing heartbeat delay times. Refer to "System Policy Object" on page 105 for more information.

**AUTOMATION SYMBOLS**

You can enter or update system automation symbols that can be set to various system-specific values. These can then be substituted into several values, like MVS automatic restart management element names, MVS workload manager resource names, job names, user specified commands, and others. Refer to "More about Policy Item AUTOMATION SYMBOLS" on page 114 for more information.

**NETVIEW**

This policy item lets you enter the following details required for interacting
System Policy Object

with NetView. This information is required only if you are using enterprise monitoring. For more information on this policy item see “More about Policy Item NETVIEW” on page 115.

AUTOMATION CONSOLE
The automation console definitions are used when SA z/OS notification messages are to be sent as write-to-operator (WTO) messages to the MVS consoles. This policy item defines the route codes and MCSFLAGS necessary for this process. WTO messages are sent to the Automation Console when none of the notification operators defined in the Notify Operators policy item are logged on or when SYSOP is defined as a notification operator.

For more information on this policy item refer to “More about Policy Item AUTOMATION CONSOLE” on page 116.

AUTOMATION SETUP
This policy item is used to enter details of your operating environment. For more information refer to “More about Policy Item AUTOMATION SETUP” on page 117.

APPLICATION GROUPS
When you select this policy item, a list of all System Application Groups is displayed, indicating which are related to this system. You can add application groups to the system or remove application groups from the system.

MONITOR RESOURCES
This policy option allows you to manage the links from the current System entry to the Monitor Resource entries. Refer to “Monitor Resource Policy Object” on page 237 for more information.

AUTOMATION TIMERS
This policy item displays a list of defined Automation Timer sets and allows you to choose which sets are to be associated with this system object. See “Timers Policy Object” on page 256 for more information.

NNT SESSIONS
This policy item displays a list of defined NNT Session sets and allows you to choose which sets are to be associated with this system object. See “NNT Sessions Policy Object” on page 286 for more information.

Note: The NNT Sessions entry type is provided for downlevel SA OS/390 systems only and will no longer be supported.

USER E-T PAIRS
This policy item displays a list of defined User E-T Pairs sets and allows you to choose which sets are to be associated with this system object. See “User E-T Pairs Policy Object” on page 292 for more information.

AUTOMATION TIMEOUT
This policy item displays a list of defined Automation Timeout sets and allows you to choose which sets are to be associated with this system object. See “Timeout Settings Policy Object” on page 258 for more information.

RESIDENT CLISTS
This policy item displays a list of defined Resident CLIST sets and allows
you to choose which sets are to be associated with this system object. See “Resident CLISTs Policy Object” on page 288 for more information.

TAPE ATTENDANCE
This policy item displays a list of defined Tape Attendance sets and allows you to choose which sets are to be associated with this system object. See “Tape Attendance Policy Object” on page 260 for more information.

APPLICATION DEFAULTS
This policy item displays a list of defined Application Defaults sets and allows you to choose a set to be associated with this system object. If an application belongs to an application class, the class inheritance takes precedence. See “Application Defaults Policy Object” on page 272 for more information.

SYSTEM DEFAULTS
This policy item displays a list of defined system defaults sets and allows you to choose a set to be associated with this system object. If an application belongs to an application class, the class inheritance takes precedence. See “System Defaults Policy Object” on page 270 for more information.

MVSCOMP DEFAULTS
This policy item displays a list of defined MVSCOMP Defaults sets and allows you to choose a set to be associated with this system object. See “MVSCOMP Defaults Policy Object” on page 269 for more information.

MVS COMPONENT
This policy item displays a list of defined MVS Component sets and allows you to choose which sets are to be associated with this system object. See “MVS Component Policy Object” on page 262 for more information.

NETWORK
This policy item displays a list of defined Network sets and allows you to choose which sets are to be associated with this system object. See “Network Policy Object” on page 282 for more information.

AUTO OPERATORS
This policy item displays a list of defined Auto Operator sets and allows you to choose which sets are to be associated with this system object. See “Auto Operators Policy Object” on page 273 for more information.

SCREENS
This policy item displays a list of defined SDF Status Definition sets and allows you to choose which sets are to be associated with this system object. See “Status Details Policy Object” on page 289 for more information.

NOTIFY OPERATORS
This policy item displays a list of defined notification operator sets who receive SA z/OS notification messages.

See “Notify Operators Policy Object” on page 279 for more information.

DB2 SYSTEM DEFAULTS
If you select this policy item, SA z/OS will display a list of DB2 system defaults which previously have been defined as described in “DB2 System Defaults” on page 242. From this list you can link one DB2 system default entry to the current system. Or you can also remove such a link and link a different entry to the system. Only one such entry should be linked to a system, otherwise multiple entries will be built with SA z/OS only loading the last one processed.
IMS XRF RESOURCES
If you select this policy item, SA z/OS will display a list of Extended Recovery Facility (XRF) resources which previously have been defined as described in "IMS XRF Resources” on page 250. From this list you can link one or more XRF resources to the current system. Or you can also remove such links and link different resources to the system.

IMS XRF STATUS FILES
If you select this policy item, SA z/OS will display a list of Extended Recovery Facility (XRF) status files which previously have been defined as described in “IMS XRF Status Files” on page 249. From this list you can link one or more XRF status files to the current system. Or you can also remove such links and link different status files to the system.

OPC SYSTEM DETAILS
If you select this policy item, SA z/OS will display a list of OPC system details which have been previously defined as described in "OPC System Details” on page 252. From this list you can link one such OPC system detail entry to the current system. Or you can also remove such a link and link a different entry to the system. Only one such entry should be linked to a system, otherwise multiple entries will be built with SA z/OS only loading the last one processed.

CONTROLLER DETAILS
If you select this policy item, SA z/OS will display a list of OPC controller details which have been previously defined as described in "Controller Details” on page 253. From this list you can link one such OPC controller detail entry to the current system. Or you can also remove such a link and link a different entry to the system. Only one such entry should be linked to a system, otherwise multiple entries will be built with SA z/OS only loading the last one processed.

WORKSTATION DOMAINS
If you select this policy item, SA z/OS will display a list of workstation domain IDs which have been previously defined as described in "Workstation Domains” on page 256. From this list you can link one such workstation domain to the current system. Or you can also remove such a link and link a different entry to the system. Only one such entry should be linked to a system, otherwise multiple entries will be built with SA z/OS only loading the last one processed.

INCLUDES
If you select this policy item, SA z/OS will display the Includes for System panel. This panel displays the entries that are defined via the Includes entry type. So with this policy item, you can manage the links between the current System entry and the Includes entries.

From the Includes for System panel you can:
• Add a link between the entries.
• Remove a link between the entries.

Note:
The Includes entry type is provided for downlevel SA z/OS systems only and will no longer be supported.

PROCESSOR
Selection of this policy item displays a panel that lists all processors and
partitions defined in the enterprise. You can select one processor or partition on which the system should run.

**TARGET SYSTEM INFO**
This policy item displays basic information about the target system, for example, time zones, operators receiving messages from the system and operator consoles.

**IPL INFO**
With this policy item you can define IPL information for the MVS or VM target system.

**More about Policy Item SYSTEM INFO**
The panel shown in Figure 84 is displayed when you select policy item SYSTEM INFO on the Policy Selection panel for Systems [Figure 83 on page 109].

<table>
<thead>
<tr>
<th>COMMANDS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFGSPDO</td>
<td>System Information</td>
</tr>
</tbody>
</table>

Command ===>

Entry Type : System  PolicyDB Name : DATABASE_NAME
Entry Name : SYSTEM_01  Enterprise Name : YOUR_ENTERPRISE

- Operating system . . . . : MVS  (MVS VM VSE LINUX CF)
- MVS SYSTYPE. . . . . . . . : FOCl  (MVS systems only)
- Image/ProcOps name . . . : FOCl

Specify information for NMC Focal Point Communication (MVS systems only):
- Heartbeat Interval . . . : 5  (1 - 60 minutes)
- Missing Heartbeat Delay . . . : 30  (1 - 3600 seconds)

*Figure 84. SYSTEM INFO Panel*

The following information can be entered:

**Operating system**
Identifies the operating system that runs on the system. Valid values are MVS, VM, VSE, LINUX and CF. CF indicates the coupling facility control code for the coupling facility in a Parallel Sysplex. LINUX indicates the Linux for zSeries operating system distributed for 390-CMOS and zSeries processors. This field is required and its default value is MVS.

**MVS SYSTYPE**
This is required for MVS systems. This value must match the system name (SYSID) as specified in the IEASYxx member of SYS1.PARMLIB. Valid characters are any alphabetic or numeric character or any of @, # or $.

Note that the first three characters of the MVS SYSTYPE must not be "AOF", as these are reserved for internal SA z/OS processing. To define a system for an alternate configuration, it must be marked with a trailing "+" in its MVS SYSTYPE. This trailing character will be removed when the automation files are built.

If you are going to enable automation for IXC102A/IXC402D the MVS SYSTYPE must not start with a numeric character. For more information see [System Automation for z/OS Customizing and Programming](#).

**Image/ProcOps name**
Specifies the name of the system for hardware related functions, for example, processor operations functions. As the entry name for the system, this name must be unique among the names of all groups, subgroups,
systems, processors, and image names defined in the current policy
database. For MVS systems, the name is defaulted to the MVS SYSNAME
but it can be changed. Valid characters are any alphabetic or numeric
character, or any of @, # or $.

Heartbeat Interval
Specifies the time interval at which the SA z/OS system operations
functions will send a generic alert from this system to the NMC Focal
Point. The SA z/OS system operations functions on the NMC Focal
Point use heartbeat alerts to verify the status forwarding path from each remote
system. The heartbeat interval is a required field in the range of 1 to 60
minutes. The default is 5 minutes. Heartbeat information is only needed
for MVS systems.

Missing Heartbeat Delay
Specifies the time that the SA z/OS system operations functions on the
NMC Focal Point will wait after a heartbeat was expected. If this time
expires without receiving the heartbeat, system operations will begin
missing heartbeat processing for the remote system. The field is provided
as a way to buffer against temporary network delays. The missing
heartbeat delay is a required field in the range of 1 to 3600 seconds. 30
seconds is the default. Heartbeat information is only needed for MVS
systems.

More about Policy Item AUTOMATION SYMBOLS
The panel shown in Figure 85 is displayed when you select policy item
AUTOMATION SYMBOLS from the Policy Selection panel for Systems (Figure 83 on
page 109).

<table>
<thead>
<tr>
<th>COMMANDS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFGSSYM</td>
<td>System Automation Symbols (AOCCLONEx)</td>
</tr>
<tr>
<td>Command ====&gt;</td>
<td></td>
</tr>
<tr>
<td>Entry Type : System</td>
<td>PolicyDB Name : DATABASE_NAME</td>
</tr>
<tr>
<td>Entry Name : SAMPLE_SYSTEM_01</td>
<td>Enterprise Name : YOUR_ENTERPRISE</td>
</tr>
<tr>
<td>More: +</td>
<td></td>
</tr>
<tr>
<td>Symbolname</td>
<td>General Description</td>
</tr>
<tr>
<td>Value</td>
<td>Specific Description</td>
</tr>
<tr>
<td>AOCCLONE</td>
<td>Used for VTAM startlist suffix</td>
</tr>
<tr>
<td>01</td>
<td></td>
</tr>
<tr>
<td>AOCCLONE1</td>
<td></td>
</tr>
<tr>
<td>AOCCLONE2</td>
<td></td>
</tr>
<tr>
<td>AOCCLONE3</td>
<td></td>
</tr>
<tr>
<td>AOCCLONE4</td>
<td></td>
</tr>
<tr>
<td>AOCCLONE5</td>
<td></td>
</tr>
</tbody>
</table>

Figure 85. Automation Symbols Panel

This panel allows you to enter or update system automation symbols that can be
set to various system-specific values. These can then be substituted into several
values, such as MVS automatic restart management element names, MVS workload
manager resource names, job names, user specified commands and others.
To specify that AOCCLONE value x is to be substituted, enter &AOCCLONE\text{Ex}. at the appropriate point in the field. You must specify both the leading ampersand (&) and the trailing period (\text{.}).

You may, additionally, specify the tilde (~) to get the 0th (unnumbered) AOCCLONE value substituted. For more details refer to “Using System Symbols and System Automation Symbols” on page 43.

This panel has the following fields:

**General Description**
This shows the general description of the system automation symbol for all systems in the policy database. It can be defined in the entry type \text{ENT} using the policy item \text{SYMBOL DESCRIPTION}, see “More about Policy Item SYMBOL DESCRIPTION” on page 88.

**Value**
This allows you to specify the value of the system automation symbol (&AOCCLONE\text{Ex} variable)

**Specific Description**
This allows you to specify a specific description of the automation symbol for this system only.

### More about Policy Item NETVIEW
The panel shown in Figure 86 is displayed when you select policy item NETVIEW from the Policy Selection panel for Systems Figure 83 on page 109.

<table>
<thead>
<tr>
<th>COMMANDS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFGSPGO</td>
<td>NetView Information</td>
</tr>
</tbody>
</table>

Specify NetView-related information about this system.

\begin{tabular}{ll}
\text{SA NetView Domain} . . . . & Name of the NetView domain  \\
\text{SA NetView Network} . . . . & Network name for \text{SA} z/OS  \\
\text{Network NetView Domain} . . & Name of the NetView domain  \\
\end{tabular}

under which \text{SA} z/OS functions run  
\!
\!
\!
NetView domain  
under which network automation runs

**Figure 86. NetView Information Panel**

The following information can be entered:

**SA NetView Domain**
This field is optional. It allows you to enter the NetView domain name of the NetView under which the \text{SA} z/OS functions run on this system. This information is applicable only if you are using a dual NetView setup. The \text{SA} z/OS NetView name can be up to five characters long.

**SA NetView Network Name**
This name is required only if you are using enterprise monitoring. It specifies the network on which the system operations NetView domain resides. It can be up to eight characters long.

**Network NetView Domain**
This field is optional. It identifies the NetView domain that runs network...
automation. It can be up to five characters long. When the automation control file is loaded, this value is assigned to the common global variable AOF_NETWORK_DOMAIN_ID.

More about Policy Item AUTOMATION CONSOLE

The panel shown in Figure 87 is displayed when you select policy item AUTOMATION CONSOLE from the Policy Selection panel for Systems on page 109.

In order to define an automation console, you need to specify:

**Console ID**

This parameter defines the logical console ID of the MVS console that is to receive the WTO messages. The logical console ID can be found by entering an MVS Display Console command from the console that is to receive the messages. The default is 01.

**Console Name**

This parameter defines the 1-8 alphanumeric console name specified in the CONSOLEnn member of the z/OS PARMLIB. The system console name *MASTER* or *ANY* can also be used. Otherwise the first character must be alphabetic. If a console name is specified, the Console ID field must remain blank.

**Route Codes**

The appropriate z/OS route codes can be selected from this panel. These route codes define which kind of messages the automation console is to receive. Each notification message has already been defined as one of these kinds of WTOs. The console can also be defined to receive all of the
notification messages regardless of how they have been defined. The last four codes on the panel are reserved for user-defined codes.

The default routing code is 1 (only messages that appear as Master Console Action WTOs are displayed on the console). Further information on the route codes can be found in z/OS MVS Routing and Descriptor Codes.

MCSFLAGS
The appropriate MVS MCSFLAG codes can be selected from the Environment Definition: CONSOLE (MCSFLAGS) panel which you reach by pressing ENTER in the Environment Definition: CONSOLE (Route Codes) panel. In this panel, you simply place an S in the A column beside the desired selection. The MCSFLAG codes are used to provide additional information for the WTO messages (for example, if flag 6 is set, the time is not included with the WTO).

If flag 1 is set, the messages are queued to the console defined in Console ID (ignoring the route codes).

Further information on the MCSFLAG codes can be found in Tivoli NetView for z/OS Customization: Using REXX and the NetView Command List Language.

More about Policy Item AUTOMATION SETUP
The panel shown in Figure 88 is displayed when you select policy item AUTOMATION SETUP from the Policy Selection panel for Systems (Figure 83 on page 109).

<table>
<thead>
<tr>
<th>COMMANDS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFPIESO</td>
<td>Environment Setup</td>
</tr>
<tr>
<td>Command ==&gt;&gt;</td>
<td></td>
</tr>
</tbody>
</table>

- **Entry Type**: System
- **Entry Name**: SAMPLE_SYSTEM_01
- **Primary JES**: JES2
- **System Monitor Time**: 00:30
- **Gateway Monitor Time**: NONE
- **Monitor Option**: INGPJMON
- **Automation Table**: INGMSG01

- **PolicyDB Name**: DATABASE_NAME
- **Enterprise Name**: YOUR_ENTERPRISE

- **Primary JES2/JES3 subsystem name**
- **Time between SYSTEM monitoring cycles (hh:mm or NONE)**
- **Time between GATEWAY monitoring cycles (hh:mm or NONE)**
- **Default routine used to monitor application status**

- **NetView automation table members**
- **The MVS SYSID in SYS1.PARMLIB**
- **Root of this system's SDF tree**

- **Environment setup user exit names**

- **System Automation UNIX installation path**

Figure 88. Environment Setup Panel
You can enter the following data:

**Primary JES**
The name of the primary Job Entry Subsystem as specified for a JES2 or JES3 subsystem. The named subsystem is the default scheduling subsystem.
System Policy Object

System Monitor Time
The time between system monitoring cycles. The default is every 59 minutes. Monitoring may be suspended by specifying NONE.

Gateway Monitor Time
The time between gateway monitoring cycles. The default is 15 minutes. Monitoring may be suspended by specifying NONE.

Monitor Option
The name of the default routine used to monitor application status. Such routines are referred to as application monitor routines. The default application monitor routine if Monitor Option is left blank is INGPJMON. If an application belongs to an application class, inheritance from class overrides the default. Or, if an application has specified an individual monitor routine (see "More about Policy Item AUTOMATION INFO" on page 157), then this routine overrides what you specify here. The following application monitor routines are valid:

AOFADMON
This routine determines the status of an application by issuing the MVS D A, jobname command. The job name used is the job name defined in the customization dialog for the application. Possible values for the application monitor status as determined by this routine are Active, Starting, Inactive.

You should avoid using this routine.

AOFAPMON
This routine determines the status of a program-to-program interface (PPI) receiver. It calls DISPPI and checks if a specific PPI receiver is active.

AOFATMON
This routine is used to determine the status of a task operating within the NetView environment. The task status is determined by issuing the NetView LIST taskname command. Note that if this routine monitor is to be used for an application, when the application is defined using the customization dialog, the application's job name must be defined to be the NetView task name.

Possible values for the application monitor status as determined by this routine are Active, Starting, Inactive.

Note: If the NetView LIST taskname command indicates that the task is not known, this condition maps to Inactive status.

AOFCPSM
This routine is a dedicated routine used to monitor the status of the SA z/OS processor operations applications using the ISQCHK service.

Possible values for the application monitor status as determined by this routine are Active, Inactive.

AOFUXMON
This routine determines the status of a resource with application type USS. This resource can either be an z/OS UNIX process, a file...
in the UNIX filesystem (HFS), or a TCP port. Depending on the nature of the resource (process, file, or port) AOFUXMON decides which internal monitoring method to use.

**INGPJMON**
This routine determines the status of an application by running a routine that searches the MVS address space control blocks (ASCBs) for address spaces with a particular job control name. The job name used is the job name defined in the customization dialog for the application.

Possible values for the application monitor status as determined by this routine are Active, Starting, Inactive.

**INGMTSYS**
With this routine, IMAGE applications for BCPII usage can be monitored.

**ISQMTSYS**
With this routine, a processor operations target system resource represented by its proxy can be monitored. You can find examples of how to use a proxy definition in System Automation for z/OS Customizing and Programming. Active operator console connections are mandatory and will be used for sending a z/OS command (for example, d t) and receiving the related response.

**NONE**
No application monitor routine is to be used (unless overridden in the Application Automation Definition panel)

**user-name**
A user-defined and created application monitor routine

**Automation Table**
You may specify multiple automation tables that are loaded and used for message automation. The automation tables are activated in the sequence as specified. The sequence of the automation tables is not changed during a configuration refresh (INGAMS REFRESH).

The first automation table of the list defined here replaces the SA z/OS initial automation table INGMSG00 that was loaded during initialization. The default is INGMSG01.

**MVS Sysname**
The name used by SA z/OS for the system name for purposes of automation.

The MVS SYSNAME for each system under SA z/OS control:
- Must be the same as the system name defined in the IEASYSxx member of the SYS1.PARMLIB data set used to IPL the system.
- May have numeric or national characters in any position.

MVS SYSNAME is also used by SA z/OS to map to the master include file built for the system. When SA z/OS is initialized, it looks for the automation control file that maps to the SYSNAME of the system it is running on. See System Automation for z/OS Planning and Installation for more information on how SA z/OS determines the default automation control file.
System Policy Object

SDF Root Name
The name used by SDF for the system name. The system name entered in the MVS Sysname field is the default.

Exit name(s)
An installation exit or exits invoked to perform additional environment setup. This item is optional. Refer to System Automation for z/OS Customizing and Programming for more information.

UNIX install path
Defines the path where SA z/OS can find its UNIX System Services programs.

More about Policy Item PROCESSOR
The panel shown in Figure 89 is displayed when you select policy item PROCESSOR on the Policy Selection panel for Systems (Figure 83 on page 109).

This panel allows you to link a system to the target hardware; it displays a list of:
- All processors together with their LPARs.
- All processors in basic mode.

The panel has the following unique fields:

Processor
Shows the processor entry name.

CPC Name
Shows the CPC Name/NAU as specified on the Processor Information panel. This field may be blank, if data for the Processor Information policy has not yet been specified.

LPAR Name
Shows the name of the logical partition as specified on the LPAR Definitions panel. For basic processors, this field is empty.
More about Policy Item TARGET SYSTEM INFO

The panel shown in Figure 90 is displayed when you select policy item TARGET SYSTEM INFO on the Policy Selection panel for Systems (Figure 83 on page 109).

- **Initialized Target System**: Enter YES to indicate that the target system is automatically initialized when SA z/OS processor operations starts. Otherwise, enter NO.

  **Note:**
  If you choose to initialize the target system automatically, be aware that when you start SA z/OS, the ISQSTART command may be completed before the target system is initialized.

- **Load Profile Name**: Enter the load profile name defined at the support element. The name you choose must correspond with the load profile defined at the support element. Leave this field blank to take the default defined at the support element.

  **Note:**
  This profile name can contain only the characters A to Z and 0 to 9.

- **Target time same as host**: If the target system is in the same time zone as the focal-point system, type YES or Y. If not, leave this field blank. For systems that run as guest systems under VM this field must be set to YES.

- **Target time earlier**: Enter the number of hours that the target system is earlier than the focal point system. (A target system in Los Angeles is three hours earlier than a focal-point system in New York.)
System Policy Object

Target time later
Enter the number of hours that the target system is later than the focal point system. (A target system in New York is three hours later than a focal point system in Los Angeles.)

Enter operators to receive messages from console
Enter the user IDs of the operators you want to receive system console and operator console messages. You can enter any valid NetView user ID.

System Console
Enter the user IDs of the operators you want to receive system console messages. The operators listed receive messages only from the system console. You can enter any valid NetView user ID. This field is optional.

Operator Console
Enter the user IDs of the operators you want to receive operator console messages. You can enter any valid NetView user ID. This field is optional.

Notes:
1. For OCF-based systems using only console integration, the operating system messages are forwarded to the system console interested operator list.
2. Operators not on these lists do not see messages from these consoles unless their user IDs are added using the ISQXMON command when SA z/OS is running.

More about Policy Item IPL INFO
The panels you can use to define your target system IPL information depends on the operating system you are running.

MVS Target System IPL Information
The panel shown in Figure 91 on page 123 is displayed when you select policy item IPL INFO on the Policy Selection panel for Systems (Figure 83 on page 109) for MVS.
Use this panel to define responses used during the MVS or z/OS target IPL process.

During the start process, the target system requests information from the operator. By providing the responses at this panel, the operator enables SA z/OS to respond to these messages. Target systems using VSE, LINUX, or CF do not request information from the operator during startup.

**IEA101A specify system parameters**

Enter the desired response to the IEA101A message. All IEA101A message responses begin with the characters R 0. You do not have to begin your response with these characters, they are provided by SA z/OS automation. Quotation marks are not necessary. The customization dialog does not verify your entry.

**IEA347A SPECIFY MASTER CATALOG PARAMETER**

ENTER THE DESIRED RESPONSE TO THE IEA347A MESSAGE. THE customization dialog DOES NOT VERIFY YOUR ENTRY.

**IEA212A or IEA213A/IEA214A duplicate volume**

Specify the device addresses to be kept online if duplicate DASD volumes exist on a z/OS system at IPL time. You have 24 entries to specify real or generic DASD device addresses. Example for a generic device address: 012*. This indicates that all addresses from 0120 to 012F must remain online.

This entry works in conjunction with the automatic response selection.

**Provide default response when neither device is specified.** Specify YES or NO to indicate if a default response is required if neither or both of the devices in the IEA213A/IEA214A message match those identified as required devices. For IEA214A messages, the current IPL (SYSRES) device is considered as being implicitly defined. If YES is selected, an automated response will always be provided. For a SYSRES device, the device shown in the message is dismounted. Otherwise the following applies:

- If neither device is in the reply list, the first device is demounted.
- If only one device is in the reply list, the remaining device is demounted.
If both devices are in the reply list, the first device in the list as scanned from left to right remains mounted and the other device is demounted.

If both devices are in the reply list due to the same generic device mask in the list as scanned from left to right, the second device is demounted.

A selection of NO will result in a response allowing the IPL to continue only if one and only one device is identified as being required. Otherwise, message ISQ1203 will be issued and the IPL process will be suspended. In such a case, the operator needs to react to system message ISQ1203.

A selection of NO and a generic device address of **** will always stop the IPL process if a duplicate volume is detected.

Notes:

1. For valid responses, see "z/OS MVS System Messages, Vol 3 (ASB-BPX)."
2. Responses to IEA101A, IEA347A, IEA212A, and IEA213A can be changed without generating and loading a new SA z/OS processor control file by the use of the host ISQXOPT command before IPLing the target system. However, these changes are not saved if SA z/OS processor is stopped or restarted.
3. You receive either message IEA212A or IEA213A depending on the level of your MVS system.
4. All message responses begin with the characters R 0. You do not have to begin your response with these characters; SA z/OS automation provides them. Quotation marks are also not necessary.

VM Target System IPL Information

To define the response to the VM IPL START message, use the panel shown in Figure 92. During the start process, the target system requests information from the operator. By providing the responses at this panel, the operator enables SA z/OS to respond to these messages.

Figure 92. VM Target System IPL Information Panel

VM IPL START message response

Enter the desired response to the VM IPL START message. The customization dialog does not verify your entry.
Note:
The VM IPL START message response field can be changed without generating and loading a new processor operations control file. You can do this by issuing the host ISQXOPT command before you IPL the target system. However, these changes are not saved if processor operations or SA z/OS is stopped or restarted.

ApplicationGroup Policy Object

An application group is a set of applications to which you assign a name. There are two types of ApplicationGroups:

1. System Application Groups - application groups that are associated with particular systems.
2. Sysplex Application Groups - application groups that are associated with sysplexes.

A Sysplex Application Group is a special type of application group that is associated with one sysplex. Sysplex Application Groups are used to monitor and automate applications and application groups that can run on multiple systems in a sysplex. Using Sysplex Application Groups means you do not need to connect the application group to each system of the sysplex. Instead, all applications of this Sysplex Application Group are known on all systems of the sysplex. You can then monitor the group at the sysplex level, rather than at system level.

The SA z/OS customization dialog allows concurrent user access capability for entry type ApplicationGroup. See also Appendix A, “Concurrent Customization by Multiple Users,” on page 365 for further information.

Creating a New ApplicationGroup

How to create new policy objects in general, is described in “How to Create New Policy Objects” on page 14.

Figure 93 on page 126 shows the panel that is displayed when you create a new application group with the New command from the Entry Name Selection panel for application groups.
The unique fields are:

Name
This is the name of the application group. It must be unique within the policy database. Valid names may contain up to twenty alphanumeric characters.

Application Group Type
This indicates whether the Application Group will be used for Systems or for a Sysplex. This attribute sets the boundaries for the selection of Resource members of the Application Group, and should be set as follows:

SYSPLEX
Use this type if the Application Group is being used to define sets of Applications and Resources that may reside on multiple systems within a sysplex.

SYSTEM
Use this type if the Application Group is being used to define sets of Applications and Resources that reside on single systems.

Application Group Types cannot be changed after this panel has been left.

Nature
In this field you can define the availability behavior of application groups and applications. You can specify one of the following attributes:

BASIC
The group is available when all of its resource members are available.

MOVE
The group is available when one of its resource members is available.

All members should be instances of the same application.

To avoid the following conflict, you should not define an application to be a member in two MOVE groups: Assume that resource RES1 is a member in two MOVE groups. One of the MOVE groups starts a member which is not shared by the other MOVE group. This implies that RES1 should be down for this
ApplicationGroup Policy Object

MOVE group, because only one member should be available. The second MOVE group now starts the shared member RESI. This causes the first MOVE group to run with two applications. If you want to assure that only one group member of a MOVE group is available, you need to insert the relationship MAKEAVAILABLE/WhenObservedDown (passive) between all group members and the group resource.

SERVER
This is like a move group. Additionally, you need to define an availability target which determines how many of its members should be available to define the server group as being available.

All members should be instances of the same application.

Depending on your selection for the Nature field, your process of entering resources into a group will be different, as described in “How to Work with Resources” on page 45. Refer to this chapter for examples on how to use the different natures appropriately.

Default Preference
This entry field specifies what preference should be used for application resources when automatically linking applications into ApplicationGroups (note that the Automatically link Application-Resources into APG entry field below must be set to YES for automatic linking; also note that preference values are only used for SERVER and MOVE ApplicationGroups). *DEF preference corresponds to default preference 700. This is the default for this field. For information about preference values and their meanings refer to “More about Policy Item RESOURCES” on page 139 and to page 65. Or, if you want to read how an operator can override preference values, refer to System Automation for z/OS User’s Guide.

Automation Name
If you want to automate and monitor the current application group, then you need to specify an automation name in this field.

At the time the application group is created, it is not yet known as to which systems you want to link it. So you need to use its WHERE USED policy item to connect it to the systems you want. Let us suppose, you created a System Application Group called SYST_APG and linked it to systems SYSTEM1 (with the same MVS SYSNAME) and SYSTEM2 (with the same MVS SYSNAME). After pressing the END key, SA z/OS generates the following resources:
SYST_APG/APG/SYSTEM1
SYST_APG/APG/SYSTEM2

Automatically link
This entry field specifies whether applications that will later be linked to the current ApplicationGroup should automatically be linked as resources too. This field defaults to YES. This field should be changed to NO if no more resources should automatically be added in case additional systems become linked to the ApplicationGroup (for System Application Groups) or to associated Sysplex-Group (for Sysplex Application Groups).

Behavior
With this field you can define whether requests to the application group should be propagated to the group members or not. With this option you can avoid having a resource unintentionally stopped if it is required for some other process outside the group.
Application Group Policy Object

ACTIVE
Propagates requests to all members of the application group according to the application group’s nature, for example, a start request to a BASIC group is propagated to all members, as a BASIC group is considered to be available when all members are available. A start request to a MOVE group is forwarded to the member with the highest preference, as a MOVE group is considered to be available when exactly one member is available.

ACTIVE is the default.

PASSIVE
Does not propagate the requests to the group members. This enables monitoring and disables the automation of the application group.

Policy Items for Application Groups
As described in “Creating a New Application Group” on page 125 there are two types of application groups:
• System Application Groups
• Sysplex Application Groups

Figure 94 shows the policy items that are displayed when a System Application Group is selected from the Entry Name Selection panel.

<table>
<thead>
<tr>
<th>ACTIONS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFGEPOL Command =&gt; Policy Selection Row 1 to 13 of 13</td>
<td>SCROLL =&gt; PAGE</td>
</tr>
</tbody>
</table>

Entry Type: ApplicationGroup  PolicyDB Name: DATABASE_NAME
Entry Name: APPL_GROUP_NAME  Enterprise Name: YOUR_ENTERPRISE

Action  Policy Name  Policy Description
DESCRIPTION  Enter description
APPLGROUP INFO  Define application group information
APPLICATIONS  Select applications for sysplex APG
TRIGGER  Select trigger
SERVICE PERIOD  Select service period
RELATIONSHIPS  Define relationships
RESOURCES  Select resources and set preferences
GENERATED RESOURCES  List resources generated for this entry
MEMBER OF  List resources where this entry is a member
WHERE USED  List sysplex groups linked to this entry
COPY  Copy data from existing entry

Figure 94. Policy Selection Panel for System Application Groups

The unique policy items for application groups are:

APPLGROUP INFO
This policy item displays information about the Application Group, as specified when creating the application group. The panel is very similar to the one used for creating the application group. You may update indicated values in this panel. See “More about Policy Item APPLGROUP INFO” on page 130 for more information.

APPLICATIONS
When you select this policy item, a list of all applications in the SA z/OS enterprise is displayed, indicating which, if any, are associated with this
ApplicationGroup Policy Object

application group. You can add applications to the application group or remove applications from the application group. If the current application group has a resource name, then the appropriate APL-resources are generated. According to the application group’s settings, the application resources (APL-resources) are linked as members into the application group resource (APG-resource). See “More about Policy Item APPLICATIONS” on page 130 for more information on this policy item.

TRIGGER

Each application group may be linked to one trigger. With this policy item, you can use the Trigger for ApplicationGroup panel to link the application group to a trigger so that all of its applications will be started or stopped depending on the defined trigger conditions.

SERVICE PERIOD

With this policy item, you can use the Service Period for ApplicationGroup panel to link a service period to the application group. Its resources are then automatically started up or shut down in the time frame specified by the service period.

RELATIONSHIPS

This policy item has the same functionality for entry types Application, ApplicationGroup, and Monitor Resource. For each application, application group, or monitor resource you may specify several relationships. They will be used to automate the startup or stopping of those applications, application groups, and monitor resources. If you select this policy item, in the Relationship Selection List panel, you can view existing relationships or define new relationships for the current application, application group, or monitor resource.

You define relationships between dependent resources and supporting resources. In the customization dialog you define relationships always from the point of view of the dependent resource. So the resource that you have currently selected will become the dependent resource, when you define a new relationship.

Refer to “More about Policy Item RELATIONSHIPS” on page 132 for more information on viewing existing or defining new dependency relationships.

RESOURCES

This policy item allows you to select Resources for current ApplicationGroup and set their preferences or name.

The panels displayed look different depending on the nature of the current application group. Refer to Chapter 4, “Scenarios on How to Use the Customization Dialog,” on page 45 and “More about Policy Item RESOURCES” on page 139 for information on how to work with these different panels.

GENERATED RESOURCES

This policy item displays a list of all resources that have been generated for the current application group and for the applications that are linked to this application group. See “More about Policy Item GENERATED RESOURCES” on page 143 for a description of the Generated Resources panel.

MEMBER OF

As resources derived from application groups can be members of other APG-resources, this policy item lets you see those resources of which the
current application group resource is a member. Refer to panel Member of in “More about Policy Item MEMBER OF” on page 144 for more details.

More about Policy Item APPLGROUP INFO
A panel like Figure 95 will be displayed if you select policy item APPLGROUP INFO from the Policy Selection Panel for Application Groups.

![Figure 95. Application Group Information Panel](image)

All of the fields apart from Owner on this panel are explained in “Creating a New ApplicationGroup” on page 125. You can update the values shown in the lower part of this panel.

**Owner**

Specifies who the operator should contact if there is an error.

More about Policy Item APPLICATIONS
If you select an application group’s policy item APPLICATIONS, you reach the Applications for ApplicationGroup panel. Figure 96 shows an example.

![Figure 96. Applications for ApplicationGroup Panel for System Application Groups](image)
Here you can add or remove applications from an application group. If you add an application to an application group which is already linked to a system, then SA z/OS will generate a resource for this application. If you previously set the option Automatically link Application-Resources to YES when creating the application group, SA z/OS will also add the new resource into this application group. Otherwise, you simply have an independent application resource that you can start, stop, or automate as required.

**More about Policy Item TRIGGER**

If you select an application group’s TRIGGER policy item, you reach the Trigger for ApplicationGroup panel. Figure 97 shows an example. For more details about triggers, refer to “Event and Trigger Support” on page 39.

Here you can add triggers to or remove them from an application group.

**More about Policy Item SERVICE PERIOD**

If you select an application group’s SERVICE PERIOD policy item, you reach the Service Period for ApplicationGroup panel. Figure 98 shows an example.

Here you can add service periods to or remove them from an application group.
More about Policy Item RELATIONSHIPS

The Relationship Selection List panel (Figure 99) is displayed if you select policy item RELATIONSHIPS from the Policy Selection panel for application groups. It shows information about existing relationships and allows you to define new relationships.

The ACTION pull-down choice offers actions to edit and delete selected relationships. The Delete command immediately deletes the relationship. The Edit command leads to the Define Relationship panel (as shown in Figure 99).

<table>
<thead>
<tr>
<th>Entry Type</th>
<th>PolicyDB Name</th>
<th>Entry Name</th>
<th>Enterprise Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ApplicationGroup</td>
<td>DATABASE_NAME</td>
<td>APPL_GROUP_NAME</td>
<td>YOUR_ENTERPRISE</td>
</tr>
</tbody>
</table>

Action #  Type    | Supporting Resource | Auto  | Chain |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>HASPARENT</td>
<td>VTAM/APG/=</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAKEAVAILABLE</td>
<td>S/APG/=</td>
<td>ACTIVE WEAK</td>
</tr>
</tbody>
</table>

Figure 99. Relationship Selection List Panel

This panel displays information about existing relationships. In all of the relationships for the current resource, this resource is the dependent resource. The columns have the same meaning as described in "Defining Relationships."

Entering the NEW command from this panel to define new relationships takes you to the Define Relationship panel described in "Defining Relationships."

Defining 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. So the resource that you are currently editing will become the dependent resource for the relationship dependency being defined. If you select this policy item, in the upcoming Define Relationship panel, (Figure 100 on page 133) you will specify the name of the supporting resource (in field Supporting Resource).
This panel shows either the automation name (Autoname) for ApplicationGroups or the subsystem name (Subsystem) for Applications.

This panel contains the following fields that can be filled in:

**Description**
You can enter a forty-character text to describe the relationship.

**Relationship Type**
Here you define the relationship between the application group, referred to as the dependent resource, with any other resource referred to as the supporting resource. The supporting resource can either be an application or an application group.

All relationships and conditions are only observed by automation if the requested goal of the involved resources allow them. Dependencies are only checked if they are of importance for the existing desired goal.

Select one of the following values:

**MAKEAVAILABLE**
The dependent resource cannot be started until the specified Condition has been fulfilled for the supporting resource. So for example, if you specify WhenAvailable as the relationship condition, then SA z/OS will start the dependent resource as soon as the supporting resource itself is available, if no request or goal contradicts.

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

**MAKEUNAVAILABLE**
The dependent resource cannot be stopped until the specified Condition has been fulfilled for the supporting resource.
PREPAVAILABLE
If the desired goal of the resource is MAKEAVAILABLE, then the PRESTART commands for the dependent resource are issued as soon as the specified Condition has been satisfied for the supporting resource.

PREPUNAVAILABLE
If the desired goal of the resource is MAKEUNAVAILABLE, then the SHUTINIT commands for the dependent resource are issued as soon as the specified Condition has been satisfied for the supporting resource.

FORCEDOWN
To force a shutdown in case the supporting resource has reached the specified satisfactory condition.

EXTERNALLY
The supporting resource will, at certain times, perform various automated actions for the dependent resource. These actions are specified through the relationships condition. For example, with this selection you specify that an external resource is responsible for starting or stopping the current (dependent) resource.

HASMONITOR
For relationships defining connections between application or application group resources and monitor resources.

<table>
<thead>
<tr>
<th>Note:</th>
</tr>
</thead>
<tbody>
<tr>
<td>This relationship type can only be used for applications (APLs) and application groups (APGs).</td>
</tr>
</tbody>
</table>

HASPARENT
With this selection you specify that the current application or application group has a parent. The parent is the supporting resource that you specify in the Supporting Resource entry field (see “Supporting Resource” on page 135).

The dependent resource cannot be started until the supporting resource is available.

The supporting resource cannot be stopped before the dependent resource is unavailable.

The supporting resource will, at certain times, perform various automated actions for the dependent resource. These actions are specified through the relationships condition (see “Condition” on page 137). For example, if you specify StartsMe as the relationship condition, then the supporting resource will start the dependent resource whenever the supporting resource is started.

A request to start the dependent resource results in issuing a start request against the supporting resource if the supporting resource is down.
Note:
This relationship type implements the parent/child relationship from earlier SA z/OS releases. Those are converted into HASPARENT, when HASPARENT is specified in the APL Relationship Type field on the PDB Convert Options panel (see “Initial Policy Database Conversion” on page 355).

HASPARENT
The only difference between the HasPassiveParent and HasParent relationship is that a request to start the dependent resource will not result in the issuing of a start request against the supporting resource if the supporting resource is down. A start request must be issued against the supporting resource before the start request for the dependent resource is processed.

Note:
This relationship type implements the parent/child relationship from earlier SA z/OS releases. Those are converted into HASPASSIVEPARENT when HASPASSIVEPARENT is specified in the APL Relationship Type field in the PDB Convert Options panel (see “Initial Policy Database Conversion” on page 355).

Supporting Resource
Here you enter the name of the resource that the dependent resource has the previously defined relationship with. This name consists of two or three parts, separated by slashes, where the last part shows the system name. You can use the equal sign '=' which then stands for the local system of the dependent resource. The syntax for valid formats is similar to the syntax for generated resources:

- APG-AutomationName/APG (for sysplex APG)
- APG-AutomationName/APG/MVSSysname (for system APG)
- APG-AutomationName/APG/= (for system APG)
- SubSystemName/APL/MVSSysname (for APL)
- SubSystemName/APL/= (for APL)
- MonitorName/MTR/MVSSysname (for MTR)
- MonitorName/MTR/= (for MTR)

Note:
If the supporting resource is an APL resource, a MTR resource, or a system APG resource, one system automation symbol (&AOACLEONE, ~ or &AOACLEONE with x=1 through 9, A through Z) can be specified in the name. System symbols cannot be specified.

For MVSSysname the special notation with a trailing ‘plus-sign’ is allowed.

Sequence Number
This number is used when the automation control file is built. Each relationship that has been assigned a number will be put into the ACF...
ApplicationGroup Policy Object

PARENT=(.) field. The position within the list is determined by this number. Valid values are 1 through 99 and blank.

The first entry in the list that is defined in the current sysplex is used for AOCQRY and for ACF compatibility with prior SA z/OS releases. The position within the list is given by this number. All relationships with the same number are sorted within their sequence number according to their type and alphabetically.

Note:
The number is forced to blank for relationships where the supporting resource is an APG or MTR resource.

Automation
Defines the processing of SA z/OS in order to satisfy the relationship:

ACTIVE
Specifies that SA z/OS will attempt to bring the resource to the desired condition to satisfy the dependency. This is the default for MAKEAVAILABLE and MAKEUNAVAILABLE.

PASSIVE
Specifies that SA z/OS will wait until the resource is in the desired condition to satisfy the dependency. That is, the request to bring the resource into the desired condition needs to come from outside, such as an operator command.

This field is forced to blank for HASMONITOR, HASPARENT, HASPASSIVEPARENT, EXTERNALLY, and FORCEDOWN relationships.

This field is forced to PASSIVE for WhenGroupHasNotFailed, WhenHardDown or any WhenObserved... condition.

Chaining
Defines whether the status of the supporting resource, or of the attached subtree is to be considered:

STRONG
Specifies that SA z/OS considers the status of the attached subtree.

WEAK
Specifies that SA z/OS considers just the status of the supporting resource (this is the default for MAKEAVAILABLE and MAKEUNAVAILABLE).

This field is forced to blank for HASMONITOR, HASPARENT, HASPASSIVEPARENT, EXTERNALLY, and FORCEDOWN relationships.

This field is forced to WEAK for PREPAVAILABLE and MAKEAVAILABLE relationships with a WhenGroupHasNotFailed condition.

Strong chaining causes SA z/OS to check all relationships of the involved resources along the attached subtree. Weak chaining simply requires that the action is permitted on the dependent resource.
In the example shown in Figure 101 for strong chaining, SA z/OS will also consider the dependency between resources B and C and B and D. So A will only be started if B, C and D are up. Or C will only be stopped if A and B are down.

Condition
Defines the condition the supporting resource must have in order to satisfy the relationship. Enter a ’?’ to get to the Relationship Condition Selection panel from where you can select an appropriate value.

Refer to "Relationship Conditions" for more information on available condition values and the context in which they may be used. If no value is given for this field the following default values will be set:

- **blank** for HASPARENT
- **StartsMe** for EXTERNALLY
- **WhenAvailable** for MAKEAVAILABLE and PREPAVAILABLE
- **WhenDown** for MAKEUNAVAILABLE and PREPUNAVAILABLE
- **WhenObservedDown** for FORCEDOWN

This field is forced to blank for HASMONITOR relationships.

**Relationship Conditions:** The relationship conditions that are available depend on the relationship.

The relationship conditions for HASPARENT, HASPASSIVEPARENT, and EXTERNALLY are:

- **StartsMe**
The dependent resource will be started whenever the supporting resource is started.

- **StopsMe**
The dependent resource will be stopped whenever the supporting resource is stopped.

- **StartsAndStopsMe**
The supporting resource will start the dependent resource whenever it is started itself. The supporting resource will stop the dependent resource whenever it is stopped itself.

Relationship conditions valid for Prepare/Make/Available/Unavailable are summarized in Table 6 on page 138.
Observed status means the observed status for the supporting resource. Desired status means the desired status for the supporting resource.

Read this table as in the following examples:

- For the condition WhenAvailable, the action specified by the Relationship Type may be taken when the supporting resource’s observed status is Available and its desired status is also Available.
- For the condition WhenRunning, the action specified in Relationship Type may be taken when the supporting resource’s observed status is Available or Degraded and its desired status is Available.

Table 6. Relationship Conditions

<table>
<thead>
<tr>
<th>Condition</th>
<th>Observed Status</th>
<th>Desired Status</th>
<th>Passive Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>WhenAvailable</td>
<td>Available, WasAvailable</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td>WhenRunning</td>
<td>Available, WasAvailable, Degraded</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td>WhenSoftDown</td>
<td>SoftDown</td>
<td>Unavailable</td>
<td></td>
</tr>
<tr>
<td>WhenHardDown</td>
<td>HardDown</td>
<td>Unavailable</td>
<td>Yes</td>
</tr>
<tr>
<td>WhenDown</td>
<td>SoftDown, HardDown, SysGone, Unknown</td>
<td>Unavailable</td>
<td></td>
</tr>
<tr>
<td>WhenAssumedDown</td>
<td>SoftDown, HardDown, SysGone, Unknown</td>
<td>Unavailable</td>
<td></td>
</tr>
<tr>
<td>WhenAvailableOrStarting</td>
<td>Available, WasAvailable OR SoftDown, Unknown AND start dependency satisfied</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td>WhenRunningOrStarting</td>
<td>Available, WasAvailable, Degraded OR SoftDown, Unknown AND start dependency satisfied</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td>WhenDownOrStopping</td>
<td>Stopping, SoftDown, HardDown, SysGone</td>
<td>Unavailable</td>
<td></td>
</tr>
<tr>
<td>WhenAssumedDownOrStopping</td>
<td>Stopping, SoftDown, HardDown, SysGone, Unknown</td>
<td>Unavailable</td>
<td></td>
</tr>
<tr>
<td>WhenStoppableOrSoftDown</td>
<td>SoftDown OR Available, WasAvailable, Degraded, Starting, Stopping, Problem AND stop dependency satisfied</td>
<td>Unavailable</td>
<td></td>
</tr>
<tr>
<td>WhenStoppableOrDown</td>
<td>SoftDown, HardDown, SysGone OR Available, WasAvailable, Degraded, Starting, Stopping, Problem AND stop dependency satisfied</td>
<td>Unavailable</td>
<td></td>
</tr>
<tr>
<td>WhenStoppableOrAssumedDown</td>
<td>SoftDown, HardDown, SysGone, Unknown OR Available, WasAvailable, Degraded, Starting, Stopping, Problem AND stop dependency satisfied</td>
<td>Unavailable</td>
<td></td>
</tr>
<tr>
<td>WhenObservedDown</td>
<td>HardDown, SoftDown, SysGone</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>WhenObservedAssumedDown</td>
<td>HardDown, SoftDown, SysGone, Unknown</td>
<td>n/a</td>
<td></td>
</tr>
</tbody>
</table>
Table 6. Relationship Conditions (continued)

<table>
<thead>
<tr>
<th>Condition</th>
<th>Observed Status</th>
<th>Desired Status</th>
<th>Passive Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>WhenObservedSoftDown</td>
<td>SoftDown</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>WhenObservedHardDown</td>
<td>HardDown</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>WhenObservedAvailable</td>
<td>Available</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>WhenObservedRunning</td>
<td>Available, Degraded</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>WhenObservedWasAvailable</td>
<td>WasAvailable</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>WhenObservedWasAvailableUnknownOrSysgone</td>
<td>WasAvailable, Unknown, Sysgone</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>WhenGroupHasNotFailed</td>
<td>Group/Failed flag not set</td>
<td>n/a</td>
<td></td>
</tr>
</tbody>
</table>

Any of these conditions can be used with any of the action relationships (PrepareAvailable, MakeAvailable, PrepareUnavailable, MakeUnavailable).

**More about Policy Item RESOURCES**

When you select policy item RESOURCES, you will first be presented with a selection panel if the application group is linked to more than one system or sysplex. There are three different panels depending on whether the application group is of type Server, Move, or Basic. The panel shown in Figure 102 is for an application group of type Server.

If you want to add resource members into an application group of nature MOVE or SERVER, you can attach preference values to these resources (see the Preference column in Figure 102).

![Figure 102. Select Resources for ApplicationGroup Panel](image)

This panel has the following fields (unless stated, the fields apply to all application group types):

**Number of selected Resources**

This shows the current number of selected Resources.

**Availability Target**

This Availability Target is different depending on the type of application group:
Server  Valid values for Server application groups are:

*ALL  This means that all selected Resources must be available before the current application group-derived Resource is considered as available.

number  This (positive) number is the number of selected Resources that must be available before the current application group-derived Resource is considered as available. This number must be less than or equal to the value in the Number of selected Resources field.

If the value entered is greater, it will be changed to the maximum allowed when you leave the panel.

-number  This (negative) number is the number of selected Resources less than the total that must be available before the current application group-derived Resource is considered as available. This number must be less than the value in the Number of selected Resources field.

If the value entered is less than or equal to the minimum value that is allowed, it will be changed to the minimum when you leave the panel.

Move  The Availability Target is always 1.

Basic  The Availability Target is not applicable.

Satisfactory Target
The Satisfactory Target is only applicable to Server application groups. It 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.

Valid values for server application groups are the same as for the Availability Target.

The Satisfactory Target may be less than, or equal to the Availability Target, but cannot be higher.

If the Satisfactory Target is of type -number then it is treated relative to the adjusted Availability Target.

Show all Resources
This can be:

YES  Shows all selectable Resources.

NO   Only shows selected Resources.

Show only Resources with String
If you enter a string in this field, only matching Resources are listed. No wildcards are supported.

Action
The following Actions are valid from the ACTIONS menu and Action prompt:

(P)RIMARY  Sets preference value to 700.
SECONDARY
    Sets preference value to 400.

ADD
    Sets preference value to 700.

REMOVE
    Removes preference value.

SELECT
    Sets preference value to 700.

Preference
    A Resource becomes SELECTED when a positive number is entered to this
    field. The preference value says what "role" the selected Resource has. The
    maximum valid value is 3200. Any value greater than 3200 will be changed
    to that valid maximum.

The following roles are related to the preference values given:

>2600
    The resource always selected regardless of its state. Use this value if
    the APL- or APG-Resource must run.

>2000
    The resource always selected unless the observed status of the resource
    is SYSGONE or HARDDOWN. It is only selected when no other viable
    member exists. A viable member is a resource that can be started and
    that is allowed to be started. In principle, the observed status is one of
    the following: Softdown, Starting, Degraded, Available or
    WasAvailable.

1000
    The resource always started, other resources stopped. (Use if you really
    want the APL- or APG-Resource to run).

700
    The resource is selected for startup after an IPL and after recovery
    situations. Members with preference 300 will be stopped in favour of
    this group. It is not stopped unless a very high preference alternative
    becomes available. This value is the default. Use it in a MOVE group
    for a resource that normally runs.

600
    Used as a threshold. A resource with a preference of 600 and lower
    will not be selected at group startup.

500
    This resource is not selected for startup after IPL but it is started as a
    backup after failure and continues running even though normal
    preference (700) alternatives become available. It is stopped at the next
    (scheduled) outage. Use this value in a MOVE group for the backup
    resource.

300
    Like 200, but a move to an alternate backup (Preference 500) would
    only occur at a (scheduled) outage.

200
    Resource started as backup, stopped as soon as better alternative
    (Preference 450 and greater) becomes available. (Use where you can
    have a backup, but you don’t really want one).
Application Group Policy Object

1. This means that the member will always be deselected and will always have a MakeUnavailable vote propagated to it. This can be useful if you want complete manual control of the move process.

0. This means that the member is a passive member and will never get a vote propagated to it by the group. If a member has its preference changed so it becomes passive, any votes that have been propagated to it from the group will be withdrawn.

Preferences other than those listed are accepted. You can find more information on the meaning of preferences in System Automation for z/OS User’s Guide. Various scenarios are described in Table 7.

### Table 7. Preference Value Scenarios

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Absolute manual control**    | Member to be activated gets preference 3200, all others get preference 1.  

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

This does not change even if the system with the selected member leaves the sysplex. |
| **Move only after system failure** | Primary member defined preference 2800, backups preference 500.  

As long as the primary is not SYSGONE it will be selected.  
If the primary is SYSGONE, 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**    | Primary member defined preference 2800, backups preference 700.  

As long as the primary member is not SYSGONE it will be selected.  
If the primary member is SYSGONE, 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** | Primary member defined preference 2801, backups preference 2800.  

As long as the primary member is not SYSGONE, it will be selected.  
If the primary becomes SYSGONE, 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, in which case the primary will be re-selected if it is available otherwise another backup would be picked. |
### Table 7. Preference Value Scenarios (continued)

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move only after system or application failure</td>
<td>Primary member defined with preference 2400, backups preference 500.</td>
</tr>
<tr>
<td></td>
<td>As long as the primary member is not SYSGONE or HARDDOWN it remains selected. If it is HARDDOWN or SYSGONE its preference is still &lt;2600, so a viable alternative can be selected.</td>
</tr>
<tr>
<td></td>
<td>The backups are only viable if you are in recovery mode (that is, the member was active when the failure occurred).</td>
</tr>
<tr>
<td>Move only if system down or application failure</td>
<td>Primary member defined with preference 2400, backups preference 700.</td>
</tr>
<tr>
<td></td>
<td>As long as the primary member is not SYSGONE or HARDDOWN it remains selected. If it is HARDDOWN or SYSGONE its preference is still &lt;2600, so a viable alternative can be selected.</td>
</tr>
<tr>
<td>Sticky move if system down, move after application failure</td>
<td>Primary member defined with preference 2401, backups preference 2400.</td>
</tr>
<tr>
<td></td>
<td>As long as the primary is not SYSGONE or HARDDOWN it remains selected. If it is SYSGONE or HARDDOWN its preference is &lt;2600, so a viable alternative may be chosen.</td>
</tr>
<tr>
<td></td>
<td>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, at which point another viable alternative will be selected, with the primary member having preference.</td>
</tr>
<tr>
<td></td>
<td>The individual shutdown of the active member will not trigger a move.</td>
</tr>
<tr>
<td>Sticky move group, unless member stopped</td>
<td>Primary member defined with preference 701, backups preference 700.</td>
</tr>
<tr>
<td></td>
<td>As long as the primary member is startable, it remains selected. If it becomes non-startable, another member is selected and that remains selected as long as it is startable.</td>
</tr>
<tr>
<td></td>
<td>Individually stopping the active member will trigger a move. If the members were defined with preferences &gt;1500, the bonus would be 175 points rather than 25 points.</td>
</tr>
<tr>
<td>Sticky move group, until told otherwise</td>
<td>Primary member defined with preference 3200, backups with preference 1600.</td>
</tr>
<tr>
<td></td>
<td>With base preferences of 3200, you get it picking a member when the sysplex is IPLd and retaining that one until an operator uses INGROUP to make it pick another one.</td>
</tr>
<tr>
<td></td>
<td>Beware of timing problems as the first system with a member to register will probably get it.</td>
</tr>
</tbody>
</table>

**More about Policy Item GENERATED RESOURCES**

The *Generated Resources* panel appears if you select policy item *GENERATED RESOURCES* from the *Policy Selection* panel for *ApplicationGroups*. This panel shows all resources that are generated by SA *z/OS for the application group itself and all
the resources for the applications that are members of this application group. This panel is for information only. No actions are allowed.

```
Figure 103. Generated Resources Panel

This panel displays the following information:

Resources for THIS entry
Shows the resources generated for the application group. The display is different for application groups of type SYSPLEX or SYSTEM: No system information is shown for Sysplex Application Groups, as several systems can be part of the sysplex.

Resources for APPLICATIONS (generated by links to this entry)
Shows the resources generated for applications that are linked to the application group.

More about Policy Item MEMBER OF

The Member of panel appears if you select policy item MEMBER OF from the Policy Selection panel for application groups.

```

Figure 104. Member of Panel for ApplicationGroups

This panel lists all Resources, where this ApplicationGroup is a member ’resource-wise’; that is, it lists all relations from Resources that have been derived from the current ApplicationGroup to other Resources. This panel is for information only. No data can be changed.

A “Member of” relation can be added or removed in the RESOURCES policy of the entry in the list’s “member of” column (that is in the RESOURCES policy of the APG Entry Name, from which the “member of” Resource was derived).
Creating Policy Name When

Figure from Figure 105 on page 145 shows the panel that is displayed when you create a new application from the Entry Name Selection panel for applications.

When you define a new application, you provide the following information:

**Name**  This field is required. It assigns a unique twenty-character name to the application you are defining. It is often convenient to use the same name...
for both the subsystem name and the application name. In some cases, however, it may be useful to have multiple applications with the same subsystem name. This enables you to have a standard "tree" of subsystems but to drop the right characteristics of an application on a specific system. The first character of the application name may not be numeric.

Subsystem Name

The Subsystem Name is required. This gives a unique, eleven-character name to the Application you are defining. This name is used by SA z/OS automation. You may have multiple applications with the same subsystem name. However, you should take care to ensure that only one of them is linked to any given system. Although the same subsystem name can be allocated to an instance and a linked class, this should be avoided as the policy inheritance can produce unpredictable results. Valid characters are any alphanumeric character or any of @, #, $ or _ (underscore), with the first character being alphanumeric or one of @, #, $ or ?.

When defining applications on SNMP target systems, it is recommended that the subsystem name be the same as the name of the process or subsystem on the SNMP target system.

The subsystem name of an existing application can be modified by overtyping the field subsystem name in the Application Information policy panel.

Notes:

1. When renaming a subsystem, the new name will be checked to be unique within your policy database, no matter which system the application is linked to. If you want to change the subsystem name to a name that already exists, you need to unlink the application from all application groups first.

2. The following values are reserved for SA z/OS and cannot be used:

   | $$$OPCA | @EXTCOND | @SERVICE | @TRIGGER |
   | ABCODEPROG | ABCODESYSTM | ABCODETRAN | ADJNETV |
   | ALL | APPLGROUP | APPLICATION | ASSOCIATION |
   | AUTOMATION | AUTOOPS | CICSCNTL | CONNECTION |
   | DB2CNTL | ENTERPRISE | ENVIRON | EXTEND |
   | FORWARD | FULLSESS | GATEWAY | GROUP |
   | HEALTHCHK | IMAGECNTL | IMSCNTL | IMSRSENM |
   | INITSTART | LISTSHUT | MONITOR | MVSCOMP |
   | MVSESA | NNT* | NONE | NTFYOP |
   | OPCAPARM | OPCCNTL | PROCESSOR | RCVRAUTO |
   | RCVRSOS | RCVRTRAN | RCVRVIOL | RECOVERY |
   | RESIDENT | RESTART | SDF | SENDCMD |
   | SERVICE | START | STSFILE | SUBGROUP |
   | SUBSYSDEF | SUBSYSTEM | SYSTEM | TERMINATE |
   | THRESHOLDS | TIMER | TRIGGER |
Object Type

CLASS
If the entry represents common policy to be inherited by one or more other application entries defined as INSTANCES, select this object type. CLASSES are used only to define inherited policy; they do not represent applications that are managed by SA z/OS.

INSTANCE
If the entry represents an application to be managed by SA z/OS, select this object type. If you link an application defined as an INSTANCE to an application defined as a CLASS you can omit policy items from the INSTANCE which allows that policy to be inherited from the specified CLASS.

Use of the object type CLASS allows you to define a common policy for multiple applications. This is advisable if these applications share common characteristics and functions. It decreases effort during:

• Initial definition.
• Administrative changes (using the customization dialog).
• Run time changes (using the SA z/OS automation engine).

An application instance is linked to a class using the LINK TO CLASS or LINK TO INSTANCES policy item. If a policy item is not specified for the instance, it is then inherited from the class. Conversely, a policy item specified for an instance overrides any class specification.

Flag definitions for classes or defaults cannot be merged with specific resource definitions. For example specifying a RESTART flag with an exit at class level and a RESTART flag with a NOAUTO period at resource level will not result in a resource definition with both. The presence of any definition at resource level will take precedence over those at class and default level. Likewise a class definition will take precedence over a default. The definitions for a specific flag type will not be merged.

The inheritance of policy from an application class occurs during SA z/OS initialization. If an application policy item is not specified in either the application instance or the class, then defaults are used from the application defaults, system defaults, and environment setup objects. Some policy items are required to be specified for an application instance. Except for job name, these same items need to be specified for an application class, but note they will never be inherited.

SA z/OS supports run time changes to application class. If the ACF command is used to change a policy item for a class, all those instances belonging to the class and inheriting that policy item are also updated.

Application Type

The application type is required. The default type value is STANDARD. The value STANDARD can be changed once, all other values cannot be changed after the application is created. Other type values are:

• JES2 if the application represents a JES2 application, see “Policy Items for JES2 Subsystems” on page 190.
• JES3 if the application represents a JES3 application, see “Policy Items for JES3 Subsystems” on page 194.
• DB2 if the application represents a DB2 application, see “Policy Items for DB2 Subsystems” on page 204.
Application Policy Object

- CICS if the application represents a CICS application, see "Policy Items for CICS Subsystems" on page 199.
- IMS if the application represents an IMS application, see "Policy Items for IMS Subsystems" on page 208.
- OPC if the application represents an OPC application, see "Policy Items for OPC Subsystems" on page 217.
- USS if the application represents an USS application, see "Policy Items for USS Subsystems" on page 218.
- IMAGE if the application represents a system image (to enable the enhanced Parallel Sysplex functions). For details, see System Automation for z/OS Customizing and Programming.

Not specifying STANDARD causes some additional policy options, specific to the automation of those applications, to be displayed on the SA z/OS Policy Selection panel. Note that once you have specified the value and created the Application entry, the value cannot be changed. Only the value STANDARD can be changed afterwards.

Subtype

Specifies the region subtype for CICS, DB2 and IMS or the controller type for OPC applications. The following values may be specified:

- **CICS**: AOR TOR FOR DOR QOR
- **DB2**: MSTR SPAS IRLM DBM1 DIST WLMS
- **IMS**: CTL TP DBRC DLS FP BMP
- **OPC**: CONTROLLER TRACKER SERVER DATASERVER

For other applications the value is forced to blank.

**JobType**

This may be one of:

**MVS**  
This is the standard MVS job type that SA z/OS normally handles. It is typically monitored using the ASCB checker. This job type can be specified explicitly with a JOBTYPE=MVS for an application but is the assumed default if the JobType entry is omitted from the definition of the application.

**NONMVS**  
This is a type of resource that may be started by SA z/OS but which cannot be monitored using an ASCB check or using the MVS D A method. For this type of resource to be monitored, if the AOFATMON routine cannot be used, either a user-provided application monitor routine is required or the message automation table must be correctly set up to maintain the job status accurately. This status is the only way of monitoring this class of job. NONMVS jobs must have STARTUP and SHUTDOWN commands defined.

**TRANSIENT**  
This is a type of job that may be started by SA z/OS but terminates without the intervention of SA z/OS. It is a short running job that will generally, but not necessarily, terminate before SA z/OS does. It can be monitored using an ASCB check.

A TRANSIENT job, as far as monitoring is concerned, is treated as an z/OS job, but it has special properties: Transient jobs must START and END before their dependent applications are started.
Transient jobs can have shutdown passes specified if there is a possibility that they may need to be shut down by SA z/OS. These passes are optional.

**Job Name**

The job name is required. This gives a unique name to the application to be run. An MVS job name is defined by MVS. There is a one-to-one mapping between the MVS job name and the subsystem name. SA z/OS automation interprets messages from MVS by taking the MVS job name and mapping it to the subsystem name.

The job name may be up to eight alphanumeric and national characters, including @, #, and $, after clone resolution, provided the first character is not numeric.

If the job type is NONMVS you can also use _ and ?.

You may include system symbols and system automation symbols (see “Assigning System Automation Symbols (AOCCLONE)” on page 42).

The special value *MASTER* is allowed for applications of type STANDARD. If the object type is CLASS then a Job Name is not allowed. See also the JCL Procedure Name if you are using replicated applications.

You may have multiple applications with the same job name but you must ensure that only one of them is linked to a certain system. Duplicates are ignored by the BUILD function.

The job name defaults to the subsystem name if possible (eight character name length and character restrictions apply).

For applications of type IMAGE the value is forced to the subsystem name.

**Transient Rerun**

This field specifies whether or not a transient job may be rerun within the lifetime of an SA z/OS. The Transient Rerun option introduces the concept of the one-shot transient job.

YES The job will be rerun as appropriate, depending on the flag settings and the IPL/RECYCLE options. SETSTATE can then be used to set it to RESTART status, and it can be part of a SHUTDOWN request with the RESTART option set to YES.

NO The job will only be run once in the lifecycle of a particular z/OS.

**Note:**
The job will also be rerun if the automation status file is deleted.

A NO setting can be overridden by the RECYCLEOPT=START option. If RECYCLEOPT=START is specified, then the transient job will be rerun when SA z/OS is recycled. If this is not desired, then the SA z/OS default can be enforced by omission of the RECYCLEOPT entry.

A value of NO affects the processing of the SETSTATE and INGREQ commands. NO is the default value for transient jobs.
**Important Note**

The following feature applies to all SA z/OS-controlled jobs, but has particular significance for one-shot transients.

On an initial automation control file load, such as at initialization, a one-shot transient will run, terminate and go to ENDED status. If a SETSTATE or INGREQ with restart is attempted it will be rejected because the status is ENDED, as the value for a one-shot is always NO.

The status information for a job can be lost if an automation control file that does not contain the job is loaded. The status information is deleted from the automation status file because the automation control file does not contain it. If the original automation control file contains a one-shot transient job then it will be restarted due to the lack of previous data in the automation status file.

If you have a job that MUST not run more than once, then you must take care with automation control file reloads. They must either be limited, or they must never be without the one-shot job entries.

When a one-shot job terminates and goes to ENDED status, it remains in that state across shutdowns. It also remains in ENDED state across an SA z/OS recycle if the START ON RECYCLE option is ‘NO’. In this case, children may be started.

An automation control file reload causes status information about applications that are not defined in the automation control file to be forgotten. If a later reload is used to redefine the application to automation, the application status is set to DOWN. Note also that some transient statuses (such as BREAKING, ABENDING, or STOPPING) may be collapsed to their final, static state (STOPPED or BROKEN) if the application address space is not active when the status determination phase of the reload runs.

**Scheduling Subsystem**

This is optional. It is the name of the subsystem that is to schedule this application, as follows:

**MSTR**  Indicating that the application is scheduled directly by the MVS/ESA master scheduler.

**blank**  Indicating that the application is scheduled by the primary JES.

**other**  Indicating that the application is scheduled by another scheduling subsystem. The value entered is the name of that subsystem. The subsystem specified must be defined as being type JES2 or JES3.

For applications of type IMAGE the value is forced to blank.

**JCL Procedure Name**

The JCL Procedure Name is optional. This specifies the name of the JCL procedure used to create this instance of the application. If specified, the default start command for the application will be:

```
MVS S procname,JOBNAME=jobname
```
where procname is the JCL Procedure Name and jobname is the Job Name specified above. If no JCL Procedure Name is specified, the default start command is:

MVS S jobname

You may include system symbols and system automation symbols (see “Assigning System Automation Symbols (AOCCLONE)” on page 42).

For applications of type IMAGE the value is forced to blank.

### Policy Items for Applications

Figure 106 shows the policy items that are displayed when an Application is selected from the Entry Name Selection panel.

Some of the policy items may be set for all application automation on a certain system. As an example, Thresholds can be set to a default value for all applications. If the definition of an application has no value specified for Thresholds, the value specified as the Application Defaults is used. If no value is specified in the Application Defaults any value specified in System Defaults is used. See also “Application Defaults Policy Object” on page 272 and “System Defaults Policy Object” on page 270.

<table>
<thead>
<tr>
<th>ACTIONS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFGEPOL</td>
<td>Policy Selection</td>
</tr>
<tr>
<td>Command</td>
<td>SCROLL</td>
</tr>
</tbody>
</table>

| Entry Type : Application | PolicyDB Name : DATABASE_NAME |
| Entry Name : APPl1 | Enterprise Name : YOUR_ENTERPRISE |

Action | Policy Name | Policy Description |
-------|-------------|--------------------|
| DESCRIPTION | Enter description |
| LINK TO CLASS | Link instance to class |
| APPLICATION INFO | Define application information |
| AUTOMATION INFO | Define application automation information |
| AUTOMATION FLAGS | Define application automation flags |
| TRIGGER | Select trigger |
| SERVICE PERIOD | Select service period |
| RELATIONSHIPS | Define relationships |
| MESSAGES/USER DATA | Define application messages and user data |
| STARTUP | Define startup procedures |
| SHUTDOWN | Define shutdown procedures |
| THRESHOLDS | Define error thresholds |
| MINOR RESOURCE FLAGS | Define application sub-component flags |
|uerdoa | Define primary and secondary associations |
| GENERATED RESOURCES | List resources generated for this entry |
| MEMBER OF | List resources where this entry is a member |
| WHERE USED | List application groups linked to this entry |
| COPY | Copy data from an existing entry |

Figure 106. Applications Defined as Instance Objects and of Type STANDARD

The unique policy items for Applications are:

**LINK TO CLASS**

This field is displayed if the application was defined as an INSTANCE (refer to “Creating a New Application” on page 145). It lists all entries that were defined as resource classes and allows you to select one class to which this resource instance should be linked.
Application Policy Object

LINK TO INSTANCES
This policy item is displayed if the application was defined as a CLASS (refer to "Creating a New Application" on page 145). It lists all instance entries with the instances that are currently linked to this class displayed with SELECTED under the status column. You can add or remove instances linked to this class.

APPLICATION INFO
This policy item displays the Application Information panel, which shows the policy for the current application as entered when it was created. You can edit a part of the policy information on this panel. The fields that contain policy information that you must not modify are read-only. To change read-only information, you must delete and recreate the application. Refer to "More about Policy Item APPLICATION INFO" on page 154 for more information.

AUTOMATION INFO
This policy item is used to define the application to SA z/OS automation. Refer to "More about Policy Item AUTOMATION INFO" on page 157 for an example and the description of the panel that is displayed when the AUTOMATION INFO policy item is selected.

AUTOMATION FLAGS
Automation flags allow you to enable and disable automation. You can disable automation for specific time periods or turn it off entirely. You can also enable or disable automation for specific automation phases, such as Initstart, Start, Recovery, Shutdown, and Restart, and for responses to specific messages and status changes within these phases.

Determine whether the automation flags currently set for your system at the system wide defaults and application levels are appropriate for this application. If they are not, define one or more application automation flags in either the application instance or class policy object. For more information, refer to "More about Policy Item AUTOMATION FLAGS" on page 162 and "Automation Flags" on page 34.

TRIGGER
Each application may be linked to one trigger. This policy item allows you to link the current application to a trigger. Only one trigger can be selected.

Triggers are used to control the starting and stopping of applications in a single system or a Parallel Sysplex. They determine the behavior of 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. For more information on how to use triggers, refer to "More about Policy Item TRIGGER" on page 164 and "Event and Trigger Support" on page 39.

SERVICE PERIOD
Service periods allow users to schedule the availability of applications, application groups, or monitor resources. With this policy item you can link a service period to the application so that it is automatically started up or shut down in the time frame specified with the service period. For more information, refer to "More about Policy Item SERVICE PERIOD" on page 164.

RELATIONSHIPS
Relationships are used to define dependencies between resources that are
used to perform in a controlled way startup or shutdown actions of
dependent and supporting applications, application groups, and monitor
resources.

This policy item has the same functionality for entry types Application,
ApplicationGroup, and Monitor Resource. For each application, application
group, or monitor resource you may specify several relationships. If you
select this policy item, in the upcoming Relationship Selection List panel, you
can view existing or define new relationships for the current application,
application group, or monitor resource.

You define relationships between dependent resources and supporting
resources. In the customization dialog you define relationships always from
the point of view of the dependent resource. So the resource that you have
currently selected will become the dependent resource, when you define a
new relationship.

Refer to “More about Policy Item RELATIONSHIPS” on page 132 for more
information on viewing or defining dependency relationships.

MESSAGES/USER DATA

This policy item allows you to specify automated actions, such as the
issuing of a command or reply, to be taken in response to a message issued
by the application.

You can also specify some of the application statuses as messages to trigger
any command of your choice.

How to proceed in specifying this policy item is described in “More about
Policy Item MESSAGES/USER DATA” on page 165.

STARTUP

This policy item allows you to add, remove or modify startup procedures
for the current application. Refer to “More about Policy Item STARTUP”
on page 175 for more information how to specify the startup automation
policy.

SHUTDOWN

You must specify NORMAL shutdown commands for any application that
you want to automate with SA z/OS. You may also specify INIT, IMMED,
FORCE, and FINAL shutdown commands.

Specifying shutdown commands or replies for the application allows the
automation to perform application shutdowns. You can specify different
shutdown commands or replies to be issued for the different types of
shutdown situations. How to achieve this is described in “More about
Policy Item SHUTDOWN” on page 179.

THRESHOLDS

Error threshold definitions for an application define how many abend
errors can occur before a message is logged or restart processing is
stopped. If no error thresholds are set here, the application uses the
thresholds 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. For more information on thresholds, see “How
SA z/OS Uses Error Thresholds” on page 31.

MINOR RESOURCE FLAGS

This policy item allows you to define flags for minor resources for the
Application Policy Object

application. Minor resources are any resources specified to be subsidiary to this application, for example, messages and statuses.

The main purposes in defining minor resources are to let you:

- Trigger automation for minor resources
- Override the automation settings of major resources for a single minor resource, or a select number of minor resources

See the description of the policy item MINOR RESOURCE FLAGS in "MVS Component Policy Object" on page 262 for information on how to manage minor resources.

The automation flags associated with minor resources are sometimes called extended automation flags. See "Automation Flags and Minor Resource Flags" on page 29 for information about them.

**SYSTEM ASSOCIATION**

This policy item allows you to define primary and secondary systems for the application. Primary and secondary categorizations help SA z/OS determine whether or not an application should be started by SA z/OS when SA z/OS is initialized on a system. These categories allow SA z/OS to perform automation, such as monitoring or shut down, on applications that can be moved by an external agent. For a detailed description on how to define system categories with this option refer to "More about Policy Item SYSTEM ASSOCIATION" on page 187.

**GENERATED RESOURCES**

This policy item lists all resources that were generated for the current application. Application resources are generated when a link between an application and a system occurs. Applications can never be linked directly to systems. This link happens indirectly by linking an application to an application group (either in an application’s WHERE USED policy item or in an application group’s APPLICATIONS policy item.)

The GENERATED RESOURCES panel is displayed which looks similar to the one shown in Figure 103 on page 144 however, the part Resources for THIS entry, showing the application group resource is missing.

**MEMBER OF**

This policy item lets you see the APG-Resources of which the current application is a member. Refer to "More about Policy Item MEMBER OF" on page 144 for more details.

**More about Policy Item APPLICATION INFO**

If you select the APPLICATION INFO policy item from the Policy Selection panel for Applications, the Application Information panel is displayed, as shown in Figure 107 on page 155.
Application Policy Object

Figure 107. Application Information Panel

This panel displays the following information:

Subsystem Name
Name Gives a unique, eleven character name to the Application you are defining. This name is used by SA z/OS automation. You may have multiple applications with the same subsystem name. However, only one of them can be linked to a system.

You can change the subsystem name by overwriting the existing name. If the application is linked, the name will be checked to be unique within all subsystem names of the policy database. If you want to change the subsystem name to a name that exists already, you need to unlink the application from all application groups first.

If no subsystem name is specified, the default is the value of the Entry name (if this name is not longer than 8 characters for applications of type IMAGE, and not longer than 11 characters for all other application types).

For applications of type IMAGE the subsystem name can only be an 8 character name.

Note:
Certain values are reserved for SA z/OS and cannot be used. Refer to the online help for a complete list.

Object Type
Refer to “Object Type” on page 147 for more details.

Application Type
Refer to “Application Type” on page 147 for more details.

Subtype
Refer to “Subtype” on page 148 for more details.

Job Type
Refer to “Job Type” on page 148 for more details.
Application Policy Object

**Job Name**
- Refer to "Job Name" on page 149 for more details.

**Transient Rerun**
- Refer to "Transient Rerun" on page 149 for more details.

**Scheduling Subsystem**
- Refer to "Scheduling Subsystem" on page 150 for more details.

**JCL Procedure Name**
- Refer to "JCL Procedure Name" on page 150 for more details.

**Captured Messages Limit**
- Specifies the maximum number of 'captured messages' which will be saved for the SA z/OS command DISPINFO.
  - If the field is left blank the value can be inherited from a class or the systems default definition. If neither of them are specified a default value of 0 will be used.

**MVS Automatic Restart Management Element Name**
- An application defined to MVS Automatic Restart Manager must have an element name that is unique within each sysplex where the application runs. Each element name consists of up to 16 alphanumeric and national characters including @,#,,$ and _ (underscore) after clone resolution, provided the first character is not numeric. Clone resolution occurs at automation control file load.
  - You may include system symbols and system automation symbols. For more details, see "Assigning System Automation Symbols (AOCCLONE)" on page 42.
  - The element name can be:

  **Constant**
  - The element name is the same, regardless of which system the application is running on. This is required for applications that are restartable on multiple systems in the sysplex. An application with a constant Automatic Restart Manager element name may at the same time have a cloned job name.

  **Note:**
  - An application with a cloned job name and a constant Automatic Restart Manager element name can be started on only one system within a sysplex.

  For applications of type IMAGE the value is forced to blank.

  **Cloned**
  - The application has a different element name on each system. The element name is derived from some attribute that is unique to the system on which the application runs. Applications with cloned element names must be defined to MVS Automatic Restart Manager in such a way that MVS will only attempt to restart them on the same system and not move them to other systems in the sysplex.
Applications with cloned element names may have constant job names. This can be useful if you want to have the same application running on multiple systems with Automatic Restart Manager recovery enabled.

For example, you have a USERPROC that runs on all your systems and you want Automatic Restart Manager recovery for it. Automatic restart management requires that USERPROC have a different element name on each system. Defining USERPROC to SA z/OS with a constant job name, but with a cloned element name (and perhaps some parameters to define the element name) allows you to define a single SA z/OS application with a different element name on each system.

**WLM Resource Names**

These are the names of the WLM Resources that are associated with the application. When the application is in either the UP or ENDED automation status, the resource will be set ON. When the application is in any other status, the resource will be set OFF.

You may include system symbols and system automation symbols, see “Assigning System Automation Symbols (AOCCLONE)” on page 42. A maximum of 3 different WLM resource names may be specified.

For applications of type IMAGE the value is forced to blank.

**Owner**

This specifies information for the operator about who to contact in case of error.

### More about Policy Item AUTOMATION INFO

Panel Application Automation Definition is displayed (as shown in Figure 108), if you select the policy item AUTOMATION INFO from panel Policy Selection for the Application entry type.

<table>
<thead>
<tr>
<th>COMMANDS HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command ==&gt; Application Automation Definition</td>
</tr>
</tbody>
</table>

**Entry Type**: Application  **PolicyDB Name**: DATABASE_NAME
**Entry Name**: APPL1  **Enterprise Name**: YOUR_ENTERPRISE

- **Command Prefix**
- **Message Prefix**
- **Sysname**
- **Monitor Routine**
- **Periodic Interval**

**Enter monitor information**:  
**Start on IPL**
**Start on Recycle**
**Start Timeout**
**Start Cycles**
**Restart Option**
**External Startup**

**Enter shutdown information**:  
**External Shutdown**
**Shutdown Delay**
**Cleanup Delay**

*Figure 108. Application Automation Definition Panel*

This panel contains the following fields:
Application Policy Object

Command Prefix
This is the console prefix character for the application. Many applications, such as JES2, JES3, and NetView support the use of a command prefix character which identifies a command as belonging to that application. If such support is available for this application, enter the corresponding command character.

Any command that would normally be issued at a z/OS console should be prefixed with MVS. NetView commands to control NetView automation do not require the MVS prefix.

You may include system symbols and system automation symbols, see "Assigning System Automation Symbols (AOCCLONE)" on page 42.

Note:
The single quotation character (') is reserved for SA z/OS and should not be used.

Message Prefix
Specifies one or more message prefixes for this application. You only have to specify these message prefix(es) if the application issues messages that do not contain the MVS job name. These message prefixes must be unique for this application.

Sysname
The system name for the subsystem. This is only required for JES subsystems where the Sysname for the subsystem is not the same as the MVS Sysname. If Sysname is not specified, it will default to the name specified in the AUTOMATION SETUP policy item of the System policy object.

Monitor Routine
The name of the application monitor routine used to monitor this application’s status. If you leave this field blank, the default application monitor routine is the routine that you may globally specify for a system for all applications that run on this system. You achieve this on the system’s Environment Setup panel (see “More about Policy Item AUTOMATION SETUP” on page 117). If an application belongs to an application class, inheritance from the class overrides the Environment Setup default. For valid application monitor routines provided by SA z/OS refer to “More about Policy Item AUTOMATION SETUP” on page 117. You can, however, specify a user-written monitor routine, or enter NONE if you do not want this application to be monitored.

The following specifications are valid:

INGPJM1ON
Monitors status via ASCB checking.

AOFPADM1ON
Monitors status via MVS D A commands. This should be avoided.

AOFPAT1ON
Monitors status of a task in the NetView environment.

AOFPUX1ON
Routine used to monitor USS applications.

AOFPAM1ON
Routine used to monitor a PPI receiver.
Application Policy Object

- **AOFPSM**
  Dedicated routine used to monitor SA z/OS Processor Operations.

- **INGMTSYS**
  Dedicated routine used to monitor IMAGE applications for BCPII usage.

- **ISQMTSYS**
  Dedicated routine used to monitor SA z/OS processor operations target system resources.

- **NONE**
  No monitoring routine is to be used.

- **username**
  A user-defined and created monitoring routine.

---

**Note:**
If a user-defined routine is not found by automation, the SA z/OS routine INGPJMON is used except for applications of type IMAGE. In this case INGMTSYS is the default.

If this application is linked to an application CLASS entry which has the Monitor Routine specified, the CLASS policy takes precedence over the Environment Setup policy.

### Periodic Interval
This is the amount of time between application subsystem monitoring cycles. Monitoring may be suspended by specifying NONE.

### Start on IPL
Specifies how SA z/OS determines the initial status of the application on the first occasion that SA z/OS is started after a z/OS IPL or an automation status file reallocation.

Reallocation of the automation status file has the same effect as a z/OS IPL because SA z/OS stores the last IPL date and time in the automation status file. When the automation status file is deleted, SA z/OS assumes that the next startup is the first startup following a z/OS IPL.

You can specify four values for Start on IPL:

- **YES**
  The application is always startable when the start dependency is fulfilled. The application status is set to DOWN during initialization. It is recommended that you specify YES for critical applications such as JES, VTAM, and TSO. Replying NOSTART to message AOF603D (specify initialization options) overrides this setting.

- **NO**
  The application is always non-startable. The application status is set to CTLDOWN during initialization, and you must use the SETSTATE command to instruct SA z/OS to restart it.

- **blank**
  The application is startable, depending on the status recorded for it in the automation status file. If the status is STOPPING, STOPPED, BREAKING, BROKEN, or CTLDOWN, it is not changed and the application is not started. Any other status is changed to DOWN during initialization. If the INITSTART automation flag is yes, the
application is started when its start dependency is fulfilled. This default setting is indicated as MAYBE on the SA z/OS display for the DISPINFO command.

**NONE**
This value is the same as blank except that this application, if it is an instance, does not inherit the value from its related application class.

**Start on Recycle**
The Start on Recycle option specifies how SA z/OS determines the initial status of the application when SA z/OS is re-initialized on a system that has not had an MVS IPL since SA z/OS was shut down.

You can specify four values for Start on Recycle:

**YES**
The application is always startable when the start dependency is fulfilled. The application status is set to DOWN during initialization. It is recommended that you specify YES for critical applications such as JES, VTAM, and TSO. Replying NOSTART to message AOF603D (specify initialization options) overrides this setting.

**NO**
The application is always nonstartable. The application status is set to CTLDOWN during initialization, and you must use the SETSTATE command to instruct SA z/OS to restart it.

**blank**
The application is startable, depending on the status recorded for it in the automation status file. If the status is STOPPING, STOPPED, BREAKING, BROKEN, or CTLDOWN, it is not changed, and the application is not started. Any other status is changed to DOWN during initialization. If the INITSTART automation flag is yes, the application is started when its start dependency is fulfilled.

**NONE**
This value is the same as blank except that this application does not inherit the value from its related application class.

**Start Timeout**
The start timeout interval that begins when SA z/OS issues a start command for an application. If the application fails to reach UP status in this time, an application monitor check is performed and the application’s status is set to STARTED2 if the application has been started or to INACTIVE if the application is not found. Message AOF571 is issued with a description of the situation. The default value for Start Timeout is 2 minutes.

**Start Cycles**
Specifies the number of times to cycle the start timeout period before posting the subsystem as a problem. The default value for Start Cycles is 1.

**Restart Option**
The circumstances under which the application should be restarted. The options are:

**Always**
The application is restarted if it was terminated without using the INGREQ command. Note that, if the advanced automation option AOFRESTARTALWAYS is set to 0, the application restart thresholds are checked. The application is restarted only if it has not exceeded
the critical threshold. By default the application is restarted every
time it is stopped outside the control of SA z/OS.

**Abendonly**
The application is eligible for restart only if it abnormally ends.
The application is restarted only if its critical error threshold has
not been exceeded.

**Never** The application is never eligible for restart by SA z/OS.
You can use this option to force a specific application state. For example, if
an application is defined with the restart option set to Always and an
operator tries to shut down the application without using the INGREQ
command, the application will be restarted when it has completed its
shutdown. The AOFRESTARTALWAYS advanced automation option can be
used to modify this behavior.

**External Startup**
Specifies whether the application is started externally or via a specified
startup procedure.

**INITIAL**
The application will only initially be started externally.

**ALWAYS**
The application will always be started externally.

**NEVER**
The application will never be started externally.

**blank** Indicates that the value can be inherited from an application
CLASS entry.

For applications of type IMAGE where the monitor routine is INGMTSYS
the value is forced to ALWAYS.

**External Shutdown**
Specifies whether the application is stopped externally or via a specified
shutdown procedure.

**FINAL**
The application will be stopped externally if the external
supporting resource (parent) is also being stopped.

**ALWAYS**
The application will always be stopped externally.

**NEVER**
The application will never be stopped externally.

**blank** Indicates that the value can be inherited from an application
CLASS entry

For applications of type IMAGE where the monitor routine is INGMTSYS
the value is forced to ALWAYS.

**Shutdown Delay**
The time delay between attempts to perform automated shutdown of an
application. Automation can perform multiple passes of shutdown
attempts. This value relates to shutdown commands specified for the
application. The MDFYSHUT command can be issued on shutdown passes
to change the timing between consecutive passes.
Application Policy Object

Cleanup Delay

The delay between application monitoring checks performed by SA z/OS after the final termination message (TERMMSG FINAL=YES) for an application is processed. The termination procedure is as follows:

1. When SA z/OS receives the final termination message for an application, it waits for the period defined in the AOFPAUSE global variable before performing its first application monitoring check. (See System Automation for z/OS Customizing and Programming for more information on AOFPAUSE.)

2. If the application monitoring check shows that the application is still active, it will wait for the period specified in Cleanup Delay and try again. This will be repeated if necessary.

3. If, after two checks, an application monitor check still finds the application to be active, the application will be posted to ZOMBIE status and periodically rechecked at increasing intervals.

The default value for the Cleanup Delay is 12 seconds.

More about Policy Item AUTOMATION FLAGS

The Flag Automation Specification panel [Figure 109] is displayed if you select policy item AUTOMATION FLAGS from the Policy Selection panel for applications. For more details about automation flags, refer to “Automation Flags” on page 34.

<table>
<thead>
<tr>
<th>Commands</th>
<th>Actions</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flag Automation Specification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Command ===＞</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Entry Type : Application</td>
<td>PolicyDB Name : DATABASE_NAME</td>
<td></td>
</tr>
<tr>
<td>Entry Name : APPL1</td>
<td>Enterprise Name : YOUR_ENTERPRISE</td>
<td></td>
</tr>
<tr>
<td>Resource: APPL1</td>
<td>More: +</td>
<td></td>
</tr>
<tr>
<td>Enter level of automation desired.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Automation Flags: Y = Yes</td>
<td>N = No</td>
<td>E = Exits</td>
</tr>
<tr>
<td>Assist Flags: D = Display</td>
<td>L = Log</td>
<td>N = None</td>
</tr>
<tr>
<td>Actions</td>
<td>Flag</td>
<td>Auto</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Automation . .</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Recovery . .</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Start. . . .</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ShutDown . .</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Initstart. .</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Restart . .</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Enter or Display times to disable automation . . NO Yes No

Figure 109. Flag Automation Specification Panel

This panel allows you to specify:

- The types of automation mode.
- Whether any automated operator functions will operate in assist mode.
- Whether user-provided exit routine(s) are to be called to determine whether or not automation is to continue.
- When automation occurs.

There are six types of automation flags supported by SA z/OS:

Automation

Is the general automation flag that must be yes for the remaining flags to be used. If it is 'NO', there is no automation for the resource.
Recovery
Indicates whether recovery of the resource (actions on messages that are
neither for the initialization nor the ending of the resource) is allowed.

Start
Indicates whether actions or responses to initialization for the resource are
allowed.

ShutDown
Indicates whether actions on or responses to ending messages are allowed.

InitStart
Indicates whether a resource can be started when its state is DOWN
(controls initial starting of the resource).

Restart
Indicates whether the restarting of a resource that abnormally ended or
was automatically shutdown is allowed.

For each flag you may specify:

Action
The Action field is an input field through which you can enter either the
full action name or fast key symbol (as denoted in the ACTIONS action bar
choice) that you want to perform, or enter ‘S’ and select the action from
the ACTIONS action bar.

The action that can be performed is SELECT (S) - selects the flag to define
or delete Exits.

Auto
Controls whether particular automated responses are allowed:

YES Indicates that automation is active.

NO Indicates that automation is not active.

EXITS Indicates that there are one or more user-provided exits routines
defined.

Note:
If no value is specified the ASSIST and EXITS specifications are not
built into the ACF.

Assist
Defines the operator assist mode for each of the specific automation states:

DISPLAY
Turns on assist mode for the chosen type of automation (for
example, START or RECOVERY). The normal actions that should
be taken under the chosen type of automation do not occur, but
messages describing what would have happened appear in the log
and Status Display Facility (SDF) is updated.

LOG Operates in the same way as DISPLAY, except that it only writes to
the log (no SDF display).

NONE Indicates that Assist mode is not operational with this choice; all
commands are issued normally.

NONE is the default.
Note:
Assist is not used on the general automation flag. An entry in this field is moved over to all the other specific flags.

The **Exits** field displays the number of exits defined.

If you want to enable automation but specify times when it should be disabled, you can enter YES in the final input field. Another panel will be displayed where you can specify one or more time ranges when automation will be disabled.

**More about Policy Item TRIGGER**

The Trigger for Application panel [Figure 110](#) is displayed if you select policy item TRIGGER from the **Policy Selection** panel for applications. For more details about triggers, refer to “Event and Trigger Support” on page 39.

**More about Policy Item SERVICE PERIOD**

If you select an application’s **SERVICE PERIOD** policy item, you reach the Service Period for Application panel. [Figure 111](#) shows an example.
More about Policy Item RELATIONSHIPS

The Relationship Selection List panel (Figure 99 on page 132) is displayed if you select policy item RELATIONSHIPS from the Policy Selection panel for application groups. It shows information about existing relationships and allows you to define new relationships.

The ACTION pull-down choice offers actions to edit and delete selected relationships. The Delete command immediately deletes the relationship. The Edit command leads to the Define Relationship panel (as shown in Figure 112).

```
<table>
<thead>
<tr>
<th>COMMANDS</th>
<th>ACTIONS</th>
<th>VIEW</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFGXREL</td>
<td>Relationship Selection List</td>
<td>Command ===</td>
<td>------</td>
</tr>
</tbody>
</table>

Entry Type: Application  PolicyDB Name: DATABASE_NAME
Entry Name: APPL1        Enterprise Name: YOUR_ENTERPRISE

Action #  Type  Supporting Resource  Auto  Chain
          1  HASPARENT  VTAM/APL/=  MAKEAVAILABLE  S/APG/=  ACTIVE  WEAK
```

Figure 112. Relationship Selection List Panel

This panel displays information about existing relationships. In all of the relationships for the current resource, this resource is the dependent resource. The columns have the same meaning as described in “Defining Relationships” on page 132.

Entering the NEW command from this panel to define new relationships takes you to the Define Relationship panel described in “Defining Relationships” on page 132.

More about Policy Item MESSAGES/USER DATA

If you select policy item MESSAGES/USER DATA from the Policy Selection panel for Applications (Figure 106 on page 151), the Message Processing panel shown in Figure 113 on page 166 is displayed.
ODO

<table>
<thead>
<tr>
<th>ACTIONS HELP</th>
<th>Message Processing</th>
<th>Command ===</th>
<th>Message ID</th>
<th>Comm Rep Code User Auto Ovr</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>SCROLL===</td>
<td>PAGE</td>
<td>Action</td>
</tr>
<tr>
<td>Entry Type</td>
<td>Application</td>
<td>PolicyDB Name</td>
<td>DATABASE_NAME</td>
<td></td>
</tr>
<tr>
<td>Entry Name</td>
<td>APPL1</td>
<td>Enterprise Name</td>
<td>YOUR_ENTERPRISE</td>
<td></td>
</tr>
</tbody>
</table>

Define message IDs and their automation actions.
CMD = Command REP = Reply CODE = CODE USER = User Data
AUTO = AT Actions OVR = AT Override

<table>
<thead>
<tr>
<th>Action</th>
<th>Message ID</th>
<th>Command</th>
<th>Response</th>
<th>Code</th>
<th>User</th>
<th>Auto</th>
<th>Ovr</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmd</td>
<td>AUTODOWN</td>
<td>Status commands</td>
<td>2</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ACORESTART</td>
<td>Status commands</td>
<td>2</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CTUTDOWN</td>
<td>Status commands</td>
<td>2</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DFS07691</td>
<td>SENDING A MAIL</td>
<td>4</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DFS08441</td>
<td>SENDING A MAIL</td>
<td>1</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DFS554A</td>
<td>SENDING A MAIL</td>
<td>5</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DFS690A</td>
<td>SENDING A MAIL</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Waiting for IMS CLT-Region</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 113. Message Processing Panel

On the Message Processing panel, you specify how SA z/OS should react if the application issues the specified message. You can type one of the following types of message response into the Action column. Depending on what you type here, SA z/OS displays a follow-on panel where you further specify SA z/OS’s automated reaction to the message.

For each of the automated responses, an entry in the NetView automation table is required that specifies which routine SA z/OS should be used to process your specified action. SA z/OS is able automatically to build the NetView AT entries for all CMD, REPLY, AUTO, and OVR actions as well as for certain USER and CODE actions.

Information about how to manage the NetView automation table is given in System Automation for z/OS Customizing and Programming.

The following actions are available:

CMD For issuing commands as a response to a message, see “CMD Processing” on page 168.

REP For issuing replies, see “Reply Processing” on page 169.

CODE If you want check for certain codes within the message. If the codes are contained in the message, you can enter a value that is returned to the calling CLIST, see “Code Processing” on page 170.

USER You can use the associated panel to let SA z/OS perform user-specified actions if the current message requires more specific automation than provided by the generic routines (see “User Defined Data” on page 172).

Using the ACFFQRY common routine from within a CLIST, you can retrieve the keywords entered into this panel and process them as needed. Syntax rules for specifying user data are available in the online help. For
information on how to use ACFFQRY in a CLIST refer to ACFFQRY in System Automation for z/OS Programmer’s Reference

AUTO
You can use the associated panel to specify messages that indicate the status of a resource, see “AUTO Processing” on page 173. Once messages have been defined, they are automatically added to the NetView automation table.

OVR
This allows you to override a default NetView automation table entry generated for a message, see “Overriding Message Automation” on page 174.

Notes:
1. AUTO cannot be defined in combination with OVR.
2. The AUTO action allows you to define Ignore or Suppress, which will prevent the building of an AT entry for the message.

Reserved Message IDs
With the following message IDs and also with any of the SA z/OS defined keywords you can invoke a special SA z/OS automation:

- AMRFSHORT
- AMRFFULL
- AMRFCLEAR
- CAPMSGs
- VTAMUP
- NODEUP
- ACORESTART (which is associated with action CMD)
- WTORS
- MVSDUMPTAKEN
- MVSDUMPRESET
- MVSDUMP
- SMFDUMP
- LOGREC
- SYSLOG
- Any of the system operations statuses

You can also specify a short description of the action that the response performs.

Note:

For product automation, the following reserved message IDs exist:

- for CICS:
- ABCODETRAN (.xxxxx, where xxxx is the optional transaction name)
  Additional transaction keyword dropped
- HEALTHCHK
- LISTSHUT
- RCVRSOS
- RCVRTRAN (.xxxxx, where xxxx is the optional transaction name)
  Additional transaction keyword dropped

* for IMS:
  - ABCODETRAN (.xxxxx, where xxxx is the optional transaction name)
    Additional transaction keyword dropped
  - ABCODEPROG (.xxxxx, where xxxx is the program name) Additional
    program keyword dropped

For a list of keywords for which no AT entries are built, see Appendix C, “Message
Definitions That Do Not Create AT Entries,” on page 381.

For more information on how to use these various methods of message processing
refer to System Automation for z/OS Customizing and Programming.

**CMD Processing**

SA z/OS displays the CMD Processing panel if you entered the action code CMD
into the Action column from panel Message Processing.

```
Command ==> CMD Processing Row 1 to 2 of 20
SCROLL ==> PAGE

Entry Name : APPL1 Message ID : AUTODOWN

Enter commands to be executed when resource issues the selected message.
or define this message as status message.

Status . . . _______ ('?' for selection list)

Pass/Selection Automated Function/'*'
Command Text
FREQ
MSG 'SYSOP' ==> JES3ABEND FREQ ALERT =='

CRIT
MSG 'SYSOP' ==> JES3ABEND CRIT ALERT =='
```

Figure 114. CMD Processing Panel

Use this panel to enter commands that are executed when the application issues
the previously specified message. If you specify one of the SA z/OS provided
automation commands (described in System Automation for z/OS Operator’s
Commands), they should be queued using the NetView CMD LOW command. This
ensures that TGLOBALs in use by the calling automation procedure are not
changed by the calling command.

For a list of TGLOBALs that can be used to be substituted into each command,
prior to execution, refer to ACFCMD in System Automation for z/OS Programmer’s
Reference.
The CMD Processing panel has a Status field that defines a resource status indicated by the Message ID. Valid values are displayed in a selection list if either a question mark or unknown status information is entered.

Any value contained in the selection list can be directly defined in the status field. Alternatively, this field allows you to mark the message to be captured, ignored, or suppressed. If it is to be ignored or suppressed, no NetView Automation Table entry will be generated. No data is accepted in this field if an override (OVR) action exists.

The value entered in the Pass/Selection field defines when the command or reply should be issued. The use of this field is as follows:

**Pass** Enter a number from 1 through 99 or an asterisk (*) in this field in order to specify a Pass value. During the build of the automation control file, this value will be prefixed with ‘PASS’ to create the selection criteria for the CMD or REP entry.

**Selection**
An expression other than a Pass is considered a Selection value and will be unaltered during the build process. A Selection value can be blank or any name that does not begin with a number or contain imbedded blanks, commas, quotes, or parentheses.

**Automated Function/*s*/**
Enter the name of the automated function under which the command is to run. If the field is left blank, the function defaults to the one currently running the command list. SA z/OS automated functions are mapped to NetView automation operators. This is documented in Table 8 on page 274.

---

**Note:**
Entering a ‘*’ enables return code checking for the specified command. This means, processing continues if the command returns a return code of 0, in all other cases the processing is stopped immediately. If running in ASSIST mode, return code checking is disabled.

**Command Text**
Enter the command or command list to be issued. Any system command must be prefaced with MVS.

Pass or Selection values can be repeated for multiple text entries. Each set of entries with the same pass or selection form a Command Group.

In the NetView automation table the ISSUECMD routine is normally generated to process this kind of message. For more details, see ISSUECMD in System Automation for z/OS Programmer’s Reference.

**Reply Processing**
SA z/OS displays the Reply Processing panel if you entered the action code REP into the Action column from panel Message Processing.
Use this panel to enter the replies to be issued when this resource issues the previously specified message.

The REP entry panel has the Status field, as described in "CMD Processing" on page 168.

The use of the Pass or Selection field is similar to that described in "CMD Processing" on page 168. Pass or Selection values can be repeated for multiple text entries. Each set of entries with the same pass or selection forms a Reply Group.

In the NetView automation table the ISSUEREP routine is normally generated to process this kind of message. For more details, see ISSUEREP in System Automation for z/OS Programmer’s Reference.

Additionally, there are the following fields:

**Retry Count**

Enter a number in the range from 1 to 99 in this field to define the retry attempts to be made before the common routine decides that a reply cannot be issued. A retry occurs when the outstanding reply number is not available. There is a two second time delay between each retry. The default retry count is 0.

**Reply Text**

Type into this field the actual reply to the message. Enter it in the same format as you would do from an MVS console (without the R and the reply number). To specify a null reply, type NULL in this field.

**Code Processing**

SA z/OS displays the following panel if you entered the action code CODE in the Action column from panel Message Processing.
Use this panel to enter the value to be passed to the calling CLIST when this resource issues the previously specified message and the codes you specify in this panel are contained in the message. Your CLIST can call the common routine CDEMATCH to compare data with the information placed in the CODE fields. You can use the override action to define the associated AT entry.

For non-predefined message automation use the CDEMATCH routine in the NetView automation table to process this kind of message. For more details, see CDEMATCH in System Automation for z/OS Programmer’s Reference.

The entry fields in the Code Processing panel have the following purpose:

CODE1, 2, 3
Data entered in these fields is compared with data from the user command list (CLIST). The data in the command list is typically pulled from the message. Therefore if the codes defined here match up with codes from the message text, a value is returned to the command list. You need not use all three CODE fields.

Value Returned
If all the CODE fields defined match up with those from the command list, the value placed in this field is returned to the command list to be used for decision making.

For the message ID, if the code matching values are found then the corresponding value is returned.

Example:

<table>
<thead>
<tr>
<th>Code 1</th>
<th>Code 2</th>
<th>Code 3</th>
<th>Value Returned</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR*</td>
<td>$PJF</td>
<td></td>
<td>STOPPING</td>
</tr>
<tr>
<td>ABEND*</td>
<td>SA22</td>
<td></td>
<td>ABENDING</td>
</tr>
</tbody>
</table>

The user command list compares Code 1 and Code 2 to codes it pulled from the message. If the codes match, the corresponding value is returned to the command list. Therefore, if in the example, ABEND* and SA z/OS match with the command list codes, the value ABENDING is given to the command list (which could be used to update a status). More information about this panel is given in the online help.
WTO Buffer Shortage Recovery: The definitions can also be used for WTO buffer shortage recovery. To define these use message ID WTOBUF and define the following values:

**CODE1**
Specifies the name of the job which might or might not be canceled. The job name may be specified with the wildcard character "*" as the last character.

**CODE2**
Must be WTO, WTOR or * to indicate for which requests the job(s) might or might not be canceled. Use just * to specify WTO and WTOR requests.

**CODE3**
Must be blank.

**Value Returned**
Must be CANCEL or KEEP to indicate whether the job(s) might or might not be canceled.

**Example:**

<table>
<thead>
<tr>
<th>Code 1</th>
<th>Code 2</th>
<th>Code 3</th>
<th>Value Returned</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB1</td>
<td>WTO</td>
<td>KEEP</td>
<td></td>
</tr>
<tr>
<td>JOB2*</td>
<td>WTO</td>
<td>KEEP</td>
<td></td>
</tr>
<tr>
<td>JOB3*</td>
<td>*</td>
<td>CANCEL</td>
<td></td>
</tr>
<tr>
<td>JOB4*</td>
<td>*</td>
<td>KEEP</td>
<td></td>
</tr>
<tr>
<td>*</td>
<td>*</td>
<td>KEEP</td>
<td></td>
</tr>
</tbody>
</table>

To set up the default behavior for all jobs not explicitly defined, a specification of CODE1=* and CODE2=* is needed. To indicate that all other jobs might be cancelled specify CANCEL in the VALUE RETURNED field, otherwise specify KEEP.

The job name *MASTER* cannot be entered in the CODE1 field. If your default behavior is setup to cancel all jobs not explicitly defined, a cancel command will be issued against *MASTER* if it is the job name being checked. As *MASTER* is non-cancelable, the cancel command will be refused. The effective result will be as if you had an explicit entry for *MASTER* with a VALUE RETURNED of KEEP.

Each line on the panel contains 4 fields. There is a limit on the total number of non-blank fields from the sixth line onwards. This limit is 254.

**User Defined Data**
SA z/OS displays the User Defined Data panel, as shown in Figure 117 on page 173 if you entered the action code USER into the Action column on the Message Processing panel.
You can use this panel to let SA z/OS perform user specified actions in case the current message requires more specific automation than provided by the generic routines. Using the ACFFQRY common routine from within a CLIST, you can retrieve the keywords entered into this panel and process them as needed. Syntax rules for how to specify user data is available in the online help. For information on how to use ACFFQRY in a CLIST refer to ACFFQRY in System Automation for z/OS Programmer’s Reference.

The following fields are available:

**Keyword**

A value is required and should represent the keyword that will be assigned the User Data.

**Data**

A value is required and can contain any characters. There are, however, rules regarding how the data value must be delimited if it contains any special characters. For more information about these rules, see the online help.

**AUTO Processing**

If you enter AUTO in the Action field on the Message Processing panel, this leads to the Message Type Selection, as shown in Figure 118 on page 174.
On this panel you can select a message type or status that is associated with the message ID.

**Overriding Message Automation**

If you enter OVR in the **Action** field for a message ID, the **Automation Processing** panel is displayed, as shown in Figure 119.

This panel allows you to either define your own or override predefined NetView Automation Table (AT) entries for a given **Entry Name** and **Message ID**. If you override the entry, an asterisk is shown in the OVR column of the **Message Processing** panel. It has the following fields:

**NetView AT source**

This field may be set to:

**DEFAULT**

The data displayed in the NetView AT Condition and Action fields are predefined by SA z/OS. This data will be used to build an AT entry unless it is overridden by the user.
USER The data displayed in the NetView AT Condition and Action fields are defined by the user. This data will be used to build an AT entry.

If user data exist you can toggle between user and default data by entering USER or DEFAULT.

NetView AT condition

This field displays the condition as it is used to build an AT entry. The condition field can be overridden.

This data will be enclosed in IF/THEN keywords when building an AT. No IF/THEN keywords should be used in this field. A leading IF and a trailing THEN will be removed.

NetView AT action x

These fields display the actions as they are used to build an AT entry. All action fields can be overridden.

A semicolon ‘;’ will be appended to the last action when building an AT. No trailing semicolon should be used in this field. If used, it will be removed.

To delete user data, leave the Automation Processing panel using the END key or PF3 while either the NetView AT source field is set to DEFAULT or the condition field is blank.

You can view the default values by entering DEFAULT in the NetView AT source field. If you then try to exit the panel, a warning message appear telling you that, by leaving the panel, you will delete your AT entry override. You can then either press the END or PF3 key to continue (and delete the entry), or press the ENTER key to keep the override.

Notes:

1. No syntax checks will be done for the AT entry within the customization dialog.
2. Lower case characters are converted to uppercase.
3. Labels defined for certain SA z/OS default AT entries will not be used if the AT entry was overridden.
4. If message automation is neither predefined by SA z/OS nor defined by the user, then no AT entry is generated for the given Entry Name.

More about Policy Item STARTUP

The STARTUP policy item is used to specify scheduling subsystem, JCL procedure name (both of which are also available on policy item APPLICATION INFO), the startup parameters and all the commands necessary to start a subsystem. These commands are issued when SA z/OS is going to start the subsystem.

The STARTUP policy item can be used to define several command sequences which are executed if an application requires certain start types depending upon the circumstances, such as a cold start, warm start or recovery start. By means of the INGREQ command, you can specify which command sequence should be taken.
Or, you can use this facility for performance improvements: If for example, as shown in Figure 120, application A depends on application B and B in turn depends on C and A also has a PREPAVAILABLE relation to C, then after C is started, you can have SA z/OS issue a sequence of prestart commands to A while the supporting resource B is starting.

![Diagram showing dependencies between A, B, and C]

Figure 120. Using STARTUP Policy

If you select policy item STARTUP from the Policy Selection panel for Applications (Figure 106 on page 151), you come to the Subsystem Startup Processing panel shown in Figure 121.

<table>
<thead>
<tr>
<th>COMMANDS</th>
<th>ACTIONS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subsystem Startup Processing</td>
<td>Row 1 to 3 of 3</td>
<td></td>
</tr>
<tr>
<td>Command</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Entry Type</td>
<td>Application</td>
<td>PolicyDB Name</td>
</tr>
<tr>
<td>Entry Name</td>
<td>APPL1</td>
<td>Enterprise Name</td>
</tr>
<tr>
<td>Scheduling Subsystem</td>
<td></td>
<td>(MSTR, JES Subsystem or blank)</td>
</tr>
<tr>
<td>JCL Procedure Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Startup Parameters</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To specify automated commands when starting this subsystem, select the appropriate start phase.

<table>
<thead>
<tr>
<th>Action</th>
<th>Phase</th>
<th>Description</th>
<th>Cmd</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PRESTART</td>
<td>Executed before startup is initiated</td>
<td></td>
</tr>
<tr>
<td></td>
<td>STARTUP</td>
<td>Executed to initiate the startup</td>
<td></td>
</tr>
<tr>
<td></td>
<td>POSTSTART</td>
<td>Executed after startup has completed</td>
<td></td>
</tr>
</tbody>
</table>

Figure 121. Subsystem Startup Processing Panel

For each application, you can specify a list of commands (startup procedures) which will be used before the application is started, to start the application and also after the application has started. All specified commands will be processed during the different startup phases of the application. The STARTUP policy item is used to specify all the necessary commands.

The advantage of this concept is that you can have SA z/OS perform certain actions for the preparation of an application. For example, before starting CICS, you can have the VTAM major node activated.

You can fill in the following fields:

**Scheduling Subsystem**

This specifies the name of the subsystem that is to schedule the current application, as follows:
MSTR  This indicates that the application is scheduled directly by the 
  z/OS master scheduler.

blank  This indicates that the application is scheduled by the primary JES.

other  This indicates that the application is scheduled by another 
  scheduling subsystem. The value entered is the name of that 
  subsystem. The subsystem specified must be defined as being type 
  JES2 or JES3 (see Application Type in the Define New Entry Panel for 
  Applications (Figure 105 on page 145).

This field is optional.

JCL Procedure Name

Specifies the name of the JCL procedure used to create this instance of the 
application. If specified, the default start command for the application will 
be:

MVS S procname, JOBNAME=jobname

where procname is the JCL procedure name and jobname is the job name 
specified above. If no JCL procedure name is specified, the default start 
command is:

MVS S jobname

You may include system symbols and system automation symbols. See 
"Assigning System Automation Symbols (AOCLONE)" on page 42 for 
details on using system automation symbols.

Parameters

Specifies the desired subsystem startup parameters. These values are 
added onto the MVS START command for the subsystem. For example, the 
desired start command for VTAM may be:

$ VTAM,,,(LIST=00)

Therefore, if this command is to be submitted from SA z/OS, the 
parameters field should contain:

,,,(LIST=00)

Whatever is entered in the parameters field is appended directly to the 
MVS START command for the subsystem. Therefore, the commas 
preceding (LIST=00) are necessary.

You may include system symbols and system automation symbols. See 
"Assigning System Automation Symbols (AOCLONE)" on page 42 for 
details on using system automation symbols.

You may also include run time variables, such as SUBSAPPL, SUBSJOB, 
SUBSPROC, SUBSSHUTDLY and others. For a complete list of run time 
variables, refer to SA z/OS documentation.

You can specify commands for the following startup phases:

PRESTART

Commands entered for this type are issued before the application is 
started. You can use these to perform some preparations, for example, let 
SA z/OS issue VTAM commands to activate additional connections.
If a PrepAvailable is defined for the resource, the PRESTART commands are executed when the PrepAvailable condition is satisfied, or as soon as the start request becomes winning. Therefore, if the PRESTART commands rely on another resource running, a PreAvailable relationship should be defined.

**STARTUP**

You should enter only one command. This will be issued to start the application.

If you select this option, you need to specify the complete startup command, otherwise, no automatic startup command is generated from the policy specified in the APPLICATION INFO policy item (in entry fields Job Name and JCL Procedure Name).

**POSTSTART**

Commands entered for this type are issued immediately after the UP message has been received.

The CMD column displays the number of commands that have been entered for a particular startup entry.

For each of these options (PRESTART, STARTUP, and POSTSTART) the Startup Command Processing panel is displayed (as shown in Figure 122), where you can enter the startup commands that are specific to the current application.

This panel shows the selected start phase and the external startup value specified on policy item APPLICATION INFO and it may be used to enter the commands that are particular for this start phase.

It contains the following fields:

**External Startup**

This shows whether the application is started externally or via the specified startup commands.

**Type**

Enter the condition (for example, COLD) for which the command should be executed.
Application Policy Object

NORM is a predefined type of SA z/OS.

Leaving this field blank means that the command is executed for any condition.

Automated Function/*

Enter the name of the automated function under which the command is to run. If the field is left blank, the function defaults to the one currently running the command list. SA z/OS automated functions are mapped to NetView automation operators. This is documented in Table 8 on page 274.

Note:

Entering an ‘*’ enables return code checking for the specified command. This means, processing continues if the command returns a return code of 0. In all other cases the processing is stopped immediately. If running in ASSIST mode, return code checking is disabled.

Command Text

Enter the command or command list to be issued. Any system command must be prefaced with MVS.

More about Policy Item SHUTDOWN

If you select policy item SHUTDOWN from the Policy Selection panel for Applications (Figure 106 on page 151), you come to the Subsystem Shutdown Processing panel shown in Figure 123.

Figure 123. Subsystem Shutdown Processing Panel

The SHUTDOWN option is used to specify all the commands necessary to shutdown a subsystem and follow-on activities.

Because shutdown procedures may involve replies to outstanding messages as well as commands, the Action field is used to indicate whether a command or reply is to be specified for a particular phase of shutdown. When either CMD or REP is entered, the user is brought to another panel where the actual commands and replies can be defined, as shown in Figure 124 on page 181, Figure 125 on page 182, and Figure 126 on page 184.
Application Policy Object

Similar to the startup processing of applications described in the previous section, the following commands or replies can be chosen from this panel:

**INIT**  Commands and replies entered under this phase are issued when the shutdown process is initiated (before the subsystem is brought down).

Possible entries could be commands to send out warning messages to users or quiescing activities.

If you specified a PREPUNAVAILABLE relationship, then the commands listed under INIT will be executed if the condition is satisfied.

Initial shutdown commands do any necessary processing for an application when shutdown begins. For example, you can use initial shutdown commands to issue a warning to any logged-on users that the application will be shutting down. Or, for the JES2 subsystem, you may wish to stop users on JES2 NJE lines from starting new transmissions or sessions.

The initial shutdown commands should not be used to actually shut the application down as unpredictable results may occur.

**NORM**  If the NORM option is selected on the INGREQ STOP command, all the commands and replies entered under this phase are issued. These entries should be the commands and replies that perform a normal shutdown of the subsystem.

**IMMED**  The commands and replies specified here are run when the IMMED option on INGREQ STOP is selected. These entries bring the subsystem down immediately.

**FORCE**  INGREQ STOP also has a FORCE option that runs the entries specified here if selected. The commands and replies would be the typical ones a user would enter to force a subsystem off the system without any delay.

**FINAL**  The commands specified here are run after the final termination message has been received.

**Note:**

For this shutdown phase only commands can be specified.

The CMD and REPLY fields cannot be filled in, but instead display the number of commands or replies that have been entered for a particular shutdown entry.

The CMD and REPLY fields cannot be filled in, but instead display the number of commands or replies that have been entered for a particular shutdown entry.

SA z/OS will issue an error message when the automation control file is loaded if you have defined any applications for which there are no shutdown commands.

You can use the multiple pass feature to issue one or more groups of shutdown commands or replies for all shutdown types except initial shutdown. Commands or replies with 1 in their corresponding pass fields are the entries that are issued during the first attempt at shutting down the subsystem. If after a specified amount of time (entered in the Shut Delay parameter of the Application Automation Definition panel) the subsystem is not yet down, another attempt (or pass) is tried. The next pass runs all entries that have 2 in their pass field. This continues until the subsystem is down. Each pass may consist of zero or more commands and zero
or one reply. Note that issuing multiple replies on a single pass is NOT advisable. Either code secondary replies on subsequent passes or chain them through the automation table and ISSUEREP.

All commands for a given pass are processed before any replies for that pass are processed. After each set of commands and replies is issued, SA z/OS will wait for the amount of time defined in the Shut Delay field on the Application Automation Definition panel. After that time, if a call to the application monitor routine shows that the application has not shut down, the commands and/or reply defined for the next pass will be issued. If a call to the application monitor routine shows that the application has shut down, any subsequent shutdown passes will not be processed. If the final termination message is not received within the time specified in AOFSHUTDELAY, SA z/OS assumes that the application is down.

If SA z/OS processes all the shutdown passes defined for an application and the application monitor routine still shows the application as active, a warning is sent to the operator and the application status is set to STUCK.

The next panel displayed depends on the shutdown processing that you specify for the shutdown phases (INIT, NORM, etc.) on the Subsystem Shutdown Processing panel. You can specify either command processing CMD (C) or reply processing REP (R) for all shutdown types (except for FINAL, where you can specify only CMD). How to proceed on these follow-on panels is described in the following sections:

- **“Command Shutdown Processing for INIT and FINAL”**
- **“Command Shutdown Processing for NORM, IMMED and FORCE” on page 182**
- **“Reply Shutdown Processing” on page 183**

### Command Shutdown Processing for INIT and FINAL

For INIT and FINAL shutdown commands, the Shutdown Command Processing panel is displayed, as shown in **Figure 124**

<table>
<thead>
<tr>
<th>COMMANDS HELP</th>
<th>Shutdown Command Processing</th>
<th>Row 1 to 1 of 20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command ===&gt;</td>
<td>Shutdown Command Processing</td>
<td>SCROLL==&gt; PAGE</td>
</tr>
<tr>
<td>Entry Type : Application</td>
<td>PolicyDB Name : DATABASE_NAME</td>
<td></td>
</tr>
<tr>
<td>Entry Name : APPL1</td>
<td>Enterprise Name : YOUR.ENTERPRISE</td>
<td></td>
</tr>
<tr>
<td>Subsystem : VTAM</td>
<td>Shutdown Phase: SHUTINIT</td>
<td>External Shutdown: NEVER</td>
</tr>
</tbody>
</table>

Enter commands to be executed when the selected shutdown phase is invoked for this subsystem. Type may be NORM, IMMED or FORCE.

Type Automated Function/"*

Command text

MVS SEND 'VTAM COMING DOWN SHORTLY'

---

**Figure 124. Shutdown Command Processing Panel for INIT and FINAL**

This panel shows the subsystem name and the external startup value specified on the Application Info policy. It may be used to enter the commands that are particular for the specified shutdown phase, as follows:
**External Shutdown**

This shows whether the application is stopped externally or via the specified shutdown commands.

The following fields can be filled in:

**Type** Enter the condition for which the command should be executed. NORM, IMMED and FORCE are allowed types and correspond to the shutdown phases SHUTNORM, SHUTIMMED and SHUTFORCE.

**Automated Function**

Enter the name of the automated function under which the command is to run. If the field is left blank, the function defaults to the one currently running the command list.

**Note:** Entering a ‘*’ enables return code checking for the specified command. This means that processing continues if the command returns a return code of 0. In all other cases processing is stopped immediately. If running in ASSIST mode, return code checking is disabled.

**Command Text**

Enter the command or command list to be issued. Any system command must be prefaced with MVS.

You cannot issue multiple passes of initial shutdown commands, neither should you invoke MDFYSHUT.

**Command Shutdown Processing for NORM, IMMED and FORCE**

For NORM, IMMED, and FORCE shutdown commands, the Shutdown Command Processing panel is displayed, as shown in Figure 125.

```
  COMMANDS  HELP
  ----------------------------------------------------------
  Command ===> Shutdown Command Processing  Row 1 to 2 of 20
  SCROLL ===> PAGE

Entry Type : Application                                     PolicyDB Name : DATABASE_NAME
Entry Name  : APPL1                                          Enterprise Name : YOUR_ENTERPRISE
Subsystem   : APPL1                                          External Shutdown: NEVER
Shutdown Phase: SHUTNORM

Enter commands to be executed when the selected shutdown phase is invoked for this subsystem.

Pass Automated Function/'*'
Command text
1
  MVS Z NET,QUICK

2
  MVS Z NET,CANCEL
```

*Figure 125. Shutdown Command Processing Panel for NORM, IMMED, and FORCE*
This panel shows the subsystem name and the external startup value specified on the Application Info policy. It may be used to enter the commands that are particular for the specified shutdown phase, as follows:

**External Shutdown**

This shows whether the application is stopped externally or via the specified shutdown commands.

The following fields can be filled in:

**Pass**

Any number between 1 and 99 can be entered in this field. Commands or replies with 1 in their corresponding Pass field are the entries that are issued during the first attempt at shutting down the subsystem.

If after a specified amount of time (entered in the Shut Delay parameter from Subsystem Definition panel) the subsystem is not yet down, another attempt (or pass) is tried. The next pass runs all entries that have 2 in their Pass field. This continues until the subsystem is down. If you do not specify a pass for a command, it will be issued on every pass.

For example, Figure 125 on page 182 defines the normal shutdown commands issued for the VTAM subsystem. This example defines two passes of shutdown commands. On the first pass, the command MVS Z NET,QUICK is issued. On the second pass, the command MVS Z NET,CANCEL is issued.

In another example, MWS P TS0 could be a command for pass 1 and MWS C TS0 could be the command for pass 2 (which means that if purging TSO fails, it is canceled).

**Automated Function**

Enter the name of the automated function under which the command is to run. If the field is left blank, the function defaults to the one currently running the command list.

---

**Note:**

Entering a "*" enables return code checking for the specified command. This means that processing continues if the command returns a return code of 0. In all other cases processing is stopped immediately. If running in ASSIST mode, return code checking is disabled.

**Command Text**

Enter the command or command list to be issued. Any system command must be prefaced with MVS.

If you want to use MDFYSHUT to change the shutdown pass processing, you should specify it as a command on one of the passes.

**Reply Shutdown Processing**

For NORM, IMMED, and FORCE shutdown replies, the Shutdown Reply Processing panel is displayed if you enter action code REP, as shown in Figure 126 on page 184.
Application Policy Object

<table>
<thead>
<tr>
<th>COMMANDS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shutdown Reply Processing</td>
<td>Row 1 to 6 of 20</td>
</tr>
<tr>
<td>Command ===</td>
<td>SCROLL===</td>
</tr>
</tbody>
</table>

Entry Type : Application  PolicyDB Name : DATABASE_NAME
Entry Name : APPL1  Enterprise Name : YOUR_ENTERPRISE
Subsys : APPL1  External Shutdown: NEVER
Shutdown Phase: SHUTNORM

Enter the replies to be issued when the selected shutdown phase is invoked for this subsystem.

<table>
<thead>
<tr>
<th>Pass</th>
<th>Retry</th>
<th>Reply Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9</td>
<td>SHUT</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>TA</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>TER ACQ</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>END</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 126. Shutdown Reply Processing Panel for NORM, IMMED and FORCE

This panel shows the subsystem name and the external startup value specified on the Application Info policy. It may be used to enter the replies that are particular for the specified shutdown phase, as follows:

External Shutdown
This shows whether the application is stopped externally or via the specified shutdown commands.

The following fields can be filled in:

Pass  Any number between 1 and 99 can be entered in this field. Commands or replies with 1 in their corresponding Pass field are the entries that are issued during the first attempt at shutting down the subsystem.

If after a specified amount of time (entered in the Shut Delay parameter from Subsystem Definition panel) the subsystem is not yet down, another attempt (or pass) is tried. The next pass runs all entries that have 2 in their Pass field. This continues until the subsystem is down. A pass number is required for each shutdown reply.

For example, MVS P TSO could be a command for pass 1 and MVS C TSO could be the command for pass 2 (which means that if purging TSO fails, it is canceled).

Retry Count
A number in the range 1 to 99 can be entered in this field to define the retry attempts made before the common routine decides that a reply cannot be issued. A retry occurs when the outstanding reply number is not available. There is a 2-second time delay between each retry. The default retry count is 0.

Reply Text
Enter the reply to be issued when the selected shutdown phase is invoked for this subsystem. For a null reply enter NULL as the reply.

For INIT, you cannot issue multiple passes of initial shutdown replies. Therefore the Shutdown Reply Processing panel only contains fields for the reply text, as shown in Figure 127 on page 185.
For FINAL, shutdown reply processing is not allowed.

More about Policy Item THRESHOLDS

The Thresholds Definition panel is displayed if you select policy item THRESHOLDS on the Policy Selection panel for applications, as shown in Figure 128.

The information provided in the THRESHOLDS dialogs is used to determine when an operator should be notified of an error situation. An attempt at recovering an error (such as an abend of a subsystem) is handled by automation, but an operator may want to know if this error is occurring often.

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.
For more information on setting thresholds refer to "How SA z/OS Uses Error Thresholds" on page 31.

More about Policy Item MINOR RESOURCE FLAGS

The Minor Resource Selection panel is displayed if you select policy item MINOR RESOURCE FLAGS on the Policy Selection panel for applications, as shown in Figure 129.

<table>
<thead>
<tr>
<th>Command</th>
<th>Minor Resource Selection</th>
<th>SCROLL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entry Type: Application</td>
<td>PolicyDB Name: DATABASE_NAME</td>
<td></td>
</tr>
<tr>
<td>Entry Name: APPL1</td>
<td>Enterprise Name: YOUR_ENTERPRISE</td>
<td></td>
</tr>
<tr>
<td>Major Resource: APPL1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Select minor resource option to be altered.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 129. Minor Resource Selection Panel

This panel allows you to manage your minor resource definitions. From this panel you can do the following:

**Add** Add a minor resource by entering a name in one of the null entries at the bottom of the list and entering an S in front of the entry to select it and edit the automation flags. The following rules apply to the minor resource names:

- The name cannot contain any of the following characters: comma, single quotation mark, double quotation mark, left or right parenthesis, equal sign or a blank.
- The name cannot start or end with a period.
- The name cannot contain consecutive periods.

**Rename** Rename a minor resource by typing the new name over the current name.

**Delete** Delete a minor resource by blanking out the name. All automation policy for that minor resource will be deleted.

**Edit** Edit the automation flags for the minor resource by entering an S in the input field before the minor resource name. Note that you can select multiple entries in which case the automation flag panel will be displayed once for each selection, without having to return to this panel between selections.
Note:
If the update to any minor resource results in all of the automation flags being deleted then those minor resources will no longer be found in the ACF created during the next ‘build’. This can result from one of the following scenarios:

- Creating a new minor resource name by entering the name in one of the null entries but not entering $ to select the entry to define the automation flags.
- Editing an entry and resetting all the automation flags to blank.

More about Policy Item SYSTEM ASSOCIATION

Note:
The following description is only valid if the automation control file (ACF) created is loaded on an SA OS/390 1.3 system. For later SA z/OS releases, these definitions are no longer used for automation. You may use the definitions for your own purpose.

If you select an application group’s policy item SYSTEM ASSOCIATION, SA z/OS displays the System Association panel.

SA z/OS defines the concept of primary and secondary systems for applications for your convenience.

The System Association panel displays the current system association information for the selected application. System associations help determine whether or not SA z/OS is going to start an application that is linked to the system. There are two types of associations that can exist between a system and an application:

Primary
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.

Secondary
A system is a secondary system for an application if it is defined to
Application Policy Object

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.

A system can be a primary system for certain applications and a secondary system for others. Primary or secondary association for a system depends on the relationship between the applications and the system.

If you do not wish an application to be defined to automation on a system, you should remove the linkages between either the system and the application group containing the application or between the application group and the application.

SA z/OS always starts applications on primary systems, and never starts them on secondary systems, but does not provide any other mechanism for coordinating applications between primary and secondary systems. z/OS provides the intersystem coordination for applications which are recovered by Automatic Restart Manager. If you wish to use primary and secondary associations with an application that does not use Automatic Restart Manager or which requires cross sysplex recovery, you must provide your own coordination mechanism.

If primary or secondary associations are set, then the statuses MOVED and FALLBACK are applicable. The MOVED and FALLBACK statuses are defined to deal with the concept of primary and secondary systems. These statuses are different from an AUTODOWN or CTDOWN status associations:

- **MOVED** - The application is not running on this (primary) system. It should be active on another system.
- **FALLBACK** - The application is not running on this (secondary) system. It should be active on another system.

Refer to "Automation Flags" on page 34 for more information on automation statuses.

**Using MOVED and FALLBACK**
SA z/OS provides two automation statuses, MOVED and FALLBACK, to handle applications which normally run on one system (the primary) but which can run on an alternative system (the secondary) if the primary is unavailable.

SA z/OS does not provide any logic to actually perform a fallback from a primary to a secondary system. If you want cross-system fallback, you have to provide the coordinating logic yourself.

If you want the application moved automatically you will have to write your own code or define a MOVE-group. You can put applications into MOVED or FALLBACK status either through SETSTATE or AOCUPDT.

**More about Policy Item GENERATED RESOURCES**
The Generated Resources panel appears if you select policy item GENERATED RESOURCES from the Policy Selection panel for Applications, as shown in Figure 131 on page 189. This panel shows all resources that are generated by SA z/OS for the application. This panel is for information only. No actions are allowed.
Application Resources are generated when a link between an Application and a System occurs. Applications can never be linked directly to Systems.

This link happens indirectly by linking an Application to an Application Group (either in Application’s WHERE USED policy or in an Application Group’s APPLICATIONS policy.)

COMMANDS HELP
----------------------------------------
Command ===>
Row to 8 of 8
Generated Resources SCROLL===>
PolicyDB Name : DATABASE_NAME PAGE
Entry Type : Application
Entry Name : NET
Enterprise Name : YOUR_ENTERPRISE

Resource Name Generated by link to
Resources for THIS entry:
NET/APL/SYSTEM1 SYSTEM1
NET/APL/SYSTEM2 SYSTEM2

Figure 131. Generated Resources Panel

This panel displays the following information:

Resources for THIS entry
Shows the resources generated for the application.

Generated by link to
This says who the owner of a Resource is. In this case; owners are Application Groups because the current entry is an APL

A link to the owner creates the Resource, unlinking from the owner (or deletion of the owner) deletes the Resource..

More about Policy Item MEMBER OF for Applications
The Member of panel is displayed if you select policy item MEMBER OF on the Policy Selection panel for applications.

COMMANDS HELP
----------------------------------------
Command ===>
Member of SCROLL===>
PolicyDB Name : DATABASE_NAME PAGE
Entry Type : Application
Entry Name : AOR1
Enterprise Name : YOUR_ENTERPRISE

Own Resource... is a member of Preference Nature
AOR1/APL/SYS1 AOR_G/APG 700 MOVE/SERVER
AOR1/APL/SYS2 AOR_G/APG 700 MOVE/SERVER

Figure 132. Member of Panel for Applications

This panel lists all Resources, where this ApplicationGroup is a member "resource-wise"; that is, it lists all relations from Resources that have been derived from the current ApplicationGroup to other Resources. This panel is for information only. No data can be changed.

A "Member of" relation can be added or removed in the RESOURCES policy of the entry in the list’s "member of” column (that is in the RESOURCES policy of the APG Entry Name, from which the "member of” Resource was derived).
In this example, this panel is invoked for application AOR1 which is a member of a Sysplex Application Group. In such a case, you will see all resources generated for the sysplex.

**Policy Items for JES2 Subsystems**

Figure 133 is an example of a Policy Selection panel for a JES2 subsystem.

<table>
<thead>
<tr>
<th>ACTIONS</th>
<th>Policy Selection</th>
<th>Command</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARPGEPOL</td>
<td>Row 14 to 24 of 24</td>
<td>SCROLL</td>
<td>PAGE</td>
</tr>
</tbody>
</table>

Entry Type: Application  PolicyDB Name: DATABASE_NAME
Entry Name: JES2          Enterprise Name: YOUR_ENTERPRISE

Action  Policy Name        Policy Description
--------- ------------------- -------------------------------------
          JES2 SPOOLFULL      Define JES2 spool full procedures
          JES2 SPOOLSHORT      Define JES2 spool short procedures
          JES2 DRAIN           Define JES2 drain procedures

Figure 133. Policy Selection Panel for JES2 Subsystems

See “Policy Items for Applications” on page 151 for descriptions of most policy items on this panel. The additional policy items are:

**JES2 SPOOLFULL**

Provides for recovery from a JES spool full condition. Figure 134 on page 191 shows the recovery parameters that can be specified for JES2 when this policy item is selected.

**JES2 SPOOLSHORT**

Provides for recovery from JES spool shortage condition. The recovery parameters available for JES2 spool shortage recovery are the same as for spool full recovery. Refer to “More about Policy Item JES2 SPOOLSHORT” on page 192.

**JES2 DRAIN**

Allows you to specify which JES resources are to be drained and how they are to be drained in the event that a JES2 shutdown fails.

**More about Policy Item JES2 SPOOLFULL**

If you select policy item SPOOLFULL for a JES2 subsystem, SA z/OS displays the SPOOLFULL Processing panel.
The following parameters can be entered:

**Retry Time**

The time entered in this field defines the time between attempts at spool recovery. If the first set of recovery commands has been issued and the spool is not recovered after this amount of time, the second set of defined commands is issued. Time range 00:00:00 to 23:59:59. The default is 5 minutes.

**Buffer**

The percentage amount of spool that is to be recovered under the threshold (TGWARN for SPOOLSHORT, 100% for SPOOLFULL). The value must be positive and less than 75% of the threshold value. If invalid or not specified a default of 5 is used. For example, if TGWARN is set to 80 and Buffer is set to 10, SPOOLSHORT recovery will continue until spool utilization is less than 70%.

**Reset time**

This is the amount of time that must elapse after the spool has been driven below the recovery target before the spool incident is considered to be closed. If the spool exceeds the trigger threshold again before this interval is up, the recovery will begin again with the next pass, rather than restarting from pass 1.

The default is 15 minutes.

**CPUID 1st - 32nd**

An ordered list of system IDs belonging to the JES2 subsystems sharing the JES spool in a MAS definition. During the recovery the JES2 command D MEMBER,STATUS=ACTIVE,NAME is issued and the first machine in the list that is active on the display response is assumed to be responsible for handling the recovery.

**Note:**

You must keep the CPUID lists for JES subsystems sharing the spool space in synchronization.
If SA z/OS cannot find an active machine in the CPUID list, it will assume responsibility for the recovery itself. This can lead to multiple machines issuing recovery commands if your lists are incomplete.

**Edit Spoolfull Pass Commands**

Entering YES in this field allows you to edit the recovery commands. Figure 135 shows the **SPOOLFULL Processing** panel on which you can enter commands to be issued during SPOOLFULL processing.

![Figure 135. SPOOLFULL Processing Panel for JES2](image)

Commands are issued by passes, so a command to be issued on every pass should be specified without pass specification. Two substitution variables (EHKVAR1 and EHKVAR2) are available. EHKVAR1 contains the current spool utilization, and EHKVAR2 contains the recovery target.

**More about Policy Item JES2 SPOOLSHORT**

If you select policy item SPOOLSHORT for a JES2 subsystem, SA z/OS displays the SPOOLSHORT Processing panel, as shown in Figure 136 on page 193

---

**Application Policy Object**

192 System Automation for z/OS: System Automation for z/OS V2R3.0 Defining Automation Policy
Figure 136. SPOOLSHORT Processing Panel

The parameters that can be entered on this panel are similar to those described in “More about Policy Item JES2 SPOOLFULL” on page 190, and the Spoolshort Processing panel is similar to the Spoolfull Processing panel.

More about Policy Item JES2 DRAIN

If you select policy item DRAIN for a JES2 subsystem, SA z/OS displays the JES2 DRAIN Processing panel, as shown in Figure 137.

Figure 137. JES2 DRAIN Processing Panel

This panel is used to identify which JES resources are to be drained and how they are to be drained if a JES shutdown is unsuccessful:

Initial drain

Choose the resources that are to be stopped with a JES $P command. An initial drain occurs either because of a JES shutdown or an operator issued AOFDRAIN. YES indicates that a stop command should be issued for the resource while a NO (the default) does not drain the resource type.
Force drain

Define the drain command to be entered for each resource if JES has not successfully shutdown. Possible JES commands could be Restart ($E), Cancel ($C) or any other appropriate command. If NO is entered (the default), this resource type does not have any commands issued for it.

Policy Items for JES3 Subsystems

Figure 138 shows a Policy Selection panel for a JES3 subsystem.

<table>
<thead>
<tr>
<th>ACTION</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command: AOGEPOL</td>
<td>Policy Selection: Row 14 to 25 of 25</td>
</tr>
</tbody>
</table>

| Entry Type: Application | PolicyDB Name: DATABASE_NAME |
| Entry Name: JES3 | Enterprise Name: YOUR_ENTERPRISE |

<table>
<thead>
<tr>
<th>Action</th>
<th>Policy Name</th>
<th>Policy Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>JES3 SPOOLFULL</td>
<td>Define JES3 spool full procedures</td>
</tr>
<tr>
<td></td>
<td>JES3 SPOOLSHORT</td>
<td>Define JES3 spool short procedures</td>
</tr>
<tr>
<td></td>
<td>JES3 MONITOR</td>
<td>Define JES3 monitoring (SA 2.2 and earlier)</td>
</tr>
<tr>
<td></td>
<td>JES3 ABENDS</td>
<td>Define JES3 abend thresholds</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Figure 138. Policy Selection Panel for JES3 Subsystems

See “Policy Items for Applications” on page 151 for descriptions of most policy items on this panel. The additional policy items are:

JES3 SPOOLFULL

Allows you to specify recovery from JES spool full condition. For more information refer to “More about Policy Item JES3 SPOOLFULL.”

JES3 SPOOLSHORT

Allows you to specify recovery from JES spool short condition. For more information refer to “More about Policy Item JES3 SPOOLSHORT” on page 195.

JES3 MONITOR

Allows you to define JES3 monitor settings and commands. For more information refer to “More about Policy Item JES3 MONITOR” on page 196.

Note:

Use this policy only to define the JES3 monitor data for SA OS/390 releases prior to SA z/OS version 2.3. For newer SA z/OS releases use monitor resources (entry type MTR).

JES3 ABENDS

Set thresholds for JES3 abends. For more information on setting thresholds refer to “More about Policy Item JES3 ABENDS” on page 198.

More about Policy Item JES3 SPOOLFULL

If you select policy item SPOOLFULL for a JES3 subsystem, SA z/OS displays the SPOOLFULL Processing panel, as shown in Figure 139 on page 195.
Commands are issued by passes, so a command to be issued on every pass should be specified with an asterisk * for the pass. Two substitution variables (EHKVAR1 and EHKVAR2) are available. EHKVAR1 contains the current spool utilization, and EHKVAR2 contains the recovery target.

More about Policy Item JES3 SPOOLSHORT

If you select policy item SPOOLSHORT for a JES3 subsystem, SA z/OS displays the SPOOLSHORT Processing panel, as shown in Figure 140.

Commands are issued by passes, so a command to be issued on every pass should be specified with an asterisk * for the pass. Two substitution variables (EHKVAR1 and EHKVAR2) are available. EHKVAR1 contains the current spool utilization, and EHKVAR2 contains the recovery target.
More about Policy Item JES3 MONITOR

Note:
Use this policy only to define the JES3 monitor data for SA OS/390 releases prior to SA z/OS version 2.3. For newer SA z/OS releases use monitor resources (entry type MTR). See "Monitor Resource Policy Object" on page 237 and System Automation for z/OS Customizing and Programming for more details.

If you select policy item MONITOR for a JES3 subsystem, SA z/OS displays the JES3 Monitor Definition panel, as shown in Figure 141.

Figure 141. JES3 Monitor Definition Panel

Information used by spool recovery and certain JES3 monitoring routines is defined in this panel. If none of the entries are filled in then monitoring by JES3 is not performed. The data entered here is only valid if the processor is the JES3 global processor.

The following fields can be entered:

Total MDS
A threshold value may be set indicating a limit on the total number of jobs in setup. When an MDS monitor cycle occurs, an action will be taken if the number of jobs in setup is equal to or greater than this threshold value.

MDS queues
For each of the 10 JES3 MDS queues identified, a threshold may be set indicating a limit as to the number of jobs that particular queue may contain. If, for example, the ERROR threshold is set to 5, if 5 or more jobs are pending on the MDS Error queue when an MDS monitor cycle occurs, an action will be attempted.
The actions taken when an MDS threshold is reached are defined in the MESSAGES policy of the APL dialog. In place of the message ID, the following keywords should be entered with CMD in the Action column:

**MDSCOUNTQ**  
Current number of jobs in (MDS) setup.

**MDSCOUNTF**  
MDS Fetch Queue

**MDSCOUNTV**  
MDS Verify Queue

**MDSCOUNTW**  
MDS Wait Volume Queue

**MDSCOUNTE**  
MDS Error Queue

**MDSCOUNTA**  
MDS Allocate Queue

**MDSCOUNTB**  
MDS Breakdown Queue

**MDSCOUNTU**  
MDS Unavailable Queue

**MDSCOUNTR**  
MDS Restart Queue

**MDSCOUNTSS**  
MDS System Select Queue

**MDSCOUNTSV**  
MDS System Verify Queue

**MDS monitoring interval**  
This field defines how often (in minutes) an *IS command is to be issued in order to monitor the MDS queues for any of the MDS thresholds being reached.

**C/I monitoring interval**  
This field defines how often (in minutes) an *IA,D=CI command is issued for converter/interpreter activity monitoring purposes.

**C/I active threshold**  
This field defines the length of time (in minutes) a job may be active in C/I before &ingprods. takes an action. If C/I monitoring shows that a job has been in C/I longer than this threshold, the commands associated with the CITIME keyword are issued. Similar to MDSCOUNTx, CITIME should be entered in the Message ID field of SUBSYSTEM MESSAGES.

**Spool inquiry interval**  
This field defines how often (in minutes) a *IQ S command is issued to monitor the spool space.

**Free space threshold**  
This field defines the amount of free spool space left that causes an action to be taken. If the value given by the *IQ S command is less than the threshold value, the commands entered in the SPOOLSHORT policy of the APL dialog are issued.
JSM DSP inquiry interval
This defines how often (in minutes) an *I A,D=JSM command is to be issued to monitor for JSM (JES3 spool maintenance) activity. If monitoring shows that JSM is not active, the commands entered using the NOJSM keyword in the MESSAGES policy for JES3 definition are issued.

Edit automatically issued JES3 commands
The default for this field is NO. Entering YES brings the user to another panel where JES3 commands can be entered that are repeatedly issued at a specified interval.

There are two fields to be filled in on the JES3 Commands panel:

Interval
This defines in minutes how often the command is to be issued.

Command
The JES3 command to be issued should be entered here beside its corresponding interval.

More about Policy Item JES3 ABENDS
If you select policy item ABENDS for a JES3 subsystem, SA z/OS displays the Thresholds Specification panel, as shown in Figure 142. For more details about thresholds, see "How SA z/OS Uses Error Thresholds" on page 31.

The following fields can be entered:

Number
This field is used to define the number of occurrences of an error that creates the particular error level. The range of allowable numbers is 1 to 10.

Interval
The number of errors entered in the previous field must occur in a certain time period in order to create a specific error level. This time period is placed in this field. The format of this field is in hours and minutes. The range for hours is 00 to 99 and minutes 00 to 59. The colon between hours and minutes is optional (can be entered as HH:MM or HHMM).
The values entered in the Number and Interval fields should be such that the threshold levels are reached in a logical manner (a critical threshold should not be reached before a frequent threshold).

Invalid values are flagged if:
1. The critical number is less than or equal to the frequent number, AND
2. The critical interval is greater than or equal to the frequent interval.

A similar check is done between the frequent and infrequent values.

**Policy Items for CICS Subsystems**

Figure 143 shows the Policy Selection panel for applications that have been created with application type CICS (see entry field Application Type in Figure 105 on page 145). This Policy Selection panel contains additional policy items for a CICS subsystem.

Some of these policy items are only applicable to a CICS subsystem defined as an instance. If a subsystem CLASS of type CICS is selected, only the additional policy item RESOURCE THRESHOLDS will be available.

```
<table>
<thead>
<tr>
<th>ACTIONS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFGEPOL Command</td>
<td>Policy Selection Row 14 to 25 of 25</td>
</tr>
</tbody>
</table>

Entry Type : Application PolicyDB Name : DATABASE NAME
Entry Name : CICS Enterprise Name : YOUR_ENTERPRISE

Action | Policy Name | Policy Description
---|---|---
... |
---|---|---
CICS CONTROL | Define CICS subsystem specific data
CICS CONNECTION | Select CICS link monitoring definitions
STATE ACTION TABLE | Select CICS state action table
RESOURCE THRESHOLDS | Define CICS resource thresholds
...

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

*Figure 143. Policy Selection Panel for CICS Subsystems*

The CICS specific policy items are:

**CICS CONTROL**

With this policy item you can define the extended subsystem information required for CICS Automation. You will reach a panel described in "More about Policy Item CICS CONTROL" on page 200 where you can fill in the required information.

**CICS CONNECTION**

If you select this policy item, the panel CICS link for Application will be displayed, showing a list of CICS link definitions previously defined as described in "Link Definitions" on page 244. Use this policy item to connect a CICS link monitoring definition to the current CICS subsystem. This item is described in section CONNECTION of *System Automation for z/OS CICS Automation Programmer’s Reference and Operator’s Guide*. By selecting a link here, SA z/OS creates an entry in the automation control file as follows:

```
CONNECTION subsys.connection_id
```
where connection_id is the entry name specified when creating the entry 40/CCN. You may select multiple connections.

**STATE ACTION TABLE**

If you select this policy item, SA z/OS will display panel **CICS State/Action for Application** with a list of CICS state/action tables which previously have been defined as described in “CICS State/Action Tables” on page 243. From this list you can link one state/action table to the current CICS subsystem.

**RESOURCE THRESHOLDS**

With this policy item you can manage your CICS resource threshold definitions for the specific CICS subsystem. You will reach the panel CICS Resource thresholds selection described in “More about Policy Item RESOURCE THRESHOLDS” on page 203 where you can fill in the required information.

**More about Policy Item CICS CONTROL**

With this policy item you can define the extended subsystem information required for CICS Automation, as shown in [Figure 144](#).

---

**Figure 144. CICS Control Specifications Panel**

Fill in the following fields:

**Subtype**

This parameter describes the controller type. It can be entered on the APPLICATION INFO policy item of the application, or the application class on the Subtype field. The value shown is either the one specified on the application or the one specified on the application class to which the application is linked.

**APPLid**

Use one of the following:

**APPLID**

If XRF=NO is coded on the SIT for this CICS subsystem, use this
keyword to specify the VTAM application identifier, provided that
the name of the VTAM APPL statement and the ACB name
specified in it are the same.

(VTAM APPL name,VTAM ACBNAME)
If XRF=NO is coded on the SIT for this CICS, but the name of the
VTAM APPL statement is different from the ACBNAME specified
in it, code them both in this entry. The VTAM ACBNAME defined
must be the same as that used by the CICS application.

(generic-applid,specific-applid)
If XRF=YES is coded on the SIT for this CICS subsystem,
genéric-applid is the name given to both the active and alternate
subsystem and specific-applid is the specific name for this
subsystem. The specific-applid defined must be the same as that
used by the CICS application.

Major node
The VTAM node for this CICS subsystem. This is used for recovery
procedures when the VTAM ACB is not active.

Keypoint req
Determines whether or not a CICS keypoint is required and whether or not
Indoubt UOWs can exist. If KEYPOINTREQ=YES is specified, and a
keypoint is missed, warning message EVE677 is issued during the
shutdown. The status will also be changed to stopping if SA z/OS
determines that an immediate restart of CICS is unlikely.

KEYPOINTREQ is applicable to CICS TS and pre-CICS TS systems. When
KEYPOINTREQ=YES is specified, and warm keypoint data is not available
or UOWs indoubt exists, then SA z/OS forces the next start type of CICS
to be AUTO irrespective of what has been specified in the SIT or by the
operator. If a CICS start type of AUTO is always specified, either by the
SIT or by the operator, then the KEYPOINTREQ parameter is not needed.

At the last shutdown of CICS, if no CICS warm keypoint was taken or if
indoubt UOW(s) existed, then if KEYPOINTREQ=YES is defined in the
automation control file, a message is issued if you have chosen a
non-AUTO start type (that is, INITIAL or COLD). You then have the
option of specifying override=UOW to force a COLD or INITIAL start.
Note however that if the application is being started via a schedule and
start type COLD or INITIAL was set (via say INGSET) and if the above
mentioned conditions exist, then automation will start the application with
starttype AUTO.

SMFid
A CICS Automation command processor creates SMF records that contain
subsystem names, job names, application identifiers, event types, and event
data. These records can be used for availability reporting. The SMF record
number used is 215. If you don’t want record number 215 used, then use
this keyword to specify a different record number.

CMASid
Identifies the CICSPlex/SM CMAS region for shutdown purposes.

Auto Operator

Note:
The field is for SA OS/390 1.3 only.
Identifies the keyword used in the control file entry described in section AUTOOOPS—Automation Operators in System Automation for z/OS CICS Automation Programmer's Reference and Operator's Guide. That defines the automation operator used for this CICS subsystem.

Start Timer

**Note:**
This field is for SA OS/390 1.3 only.

The time in which a startup must initiate before an alert is sent to the operator. This is expressed in minutes and seconds (mm:ss). The default is 1 minute.

Cold Timer

**Note:**
This field is for SA OS/390 1.3 only.

The time in which a cold start must complete before an alert is sent to the operator. This is expressed in minutes and seconds (mm:ss). The default is 8 minutes.

Warm Timer

**Note:**
This field is for SA OS/390 1.3 only.

The time in which a warm start must complete before an alert is sent to the operator. This is expressed in minutes and seconds (mm:ss). The default is 3 minutes.

Emerg.Timer

**Note:**
This field is for SA OS/390 1.3 only.

The time in which an emergency start must complete before an alert is sent to the operator. This is expressed in minutes and seconds (mm:ss). The default is 7 minutes.

The extended CICS subsystem automation that you define here is also described in section CICSCNTL of System Automation for z/OS CICS Automation Programmer's Reference and Operator's Guide.

**More about Policy Item CICS CONNECTION**

With this policy item you can select CICS link monitoring definitions required for CICS Automation, as shown in Figure 145 on page 203.
This panel allows you to manage the links between the current Application entry and the CICS link entries.

**More about Policy Item STATE ACTION TABLE**

With this policy item you can manage the link from the current application entry to the CICS State/Action entry, as shown in Figure 146. Each application entry may be linked to one CICS State/Action entry.

---

**Figure 145. CICS Link for Application Panel**

This panel allows you to define a CICS State/Action for the Application. When this policy is selected a list of all CICS State/Action entries is displayed. The CICS State/Action entry that is currently linked to this Application entry is marked as "SELECTED".

**More about Policy Item RESOURCE THRESHOLDS**

With this policy item you can define CICS resource thresholds.
From this panel you can do the following:

- Add a CICS resource threshold definition by entering a name on one blank line and entering an S in front of the entry to select it to edit the thresholds. For the rules that apply to the CICS resource threshold names, refer to the panel’s online help.

- Pressing enter after typing a name and the S, brings you to the Thresholds Definition panel shown in Figure 24 on page 33 where you can define thresholds.

- Rename CICS resource thresholds by typing the new name over the current name.

- Delete CICS resource thresholds by blanking out the name. All automation policy for that resource threshold will be deleted.

- Edit the thresholds for the CICS resource thresholds by entering an S in the input field before the resource name. Note that you can select multiple entries in which case the thresholds panel will be displayed once for each selection, without having to return to this panel between selections.

**Note:**

If the update to any CICS resource thresholds results in all of the thresholds being deleted then those resources will no longer be found in the ACF created during the next ‘build’. This can result from one of the following scenarios:

- Creating a new resource threshold name by entering the name in one of the null entries but not entering S to select the entry to define the thresholds.
- Editing an entry and resetting all the thresholds to blank.

**Policy Items for DB2 Subsystems**

Figure 148 on page 205 shows the Policy Selection panel for applications that have been created with application type DB2 (see entry field Application Type in Figure 105 on page 145). This Policy Selection panel contains additional policy items for a DB2 subsystem.
Some of these policy items are only applicable to a DB2 subsystem defined as an instance. If a subsystem CLASS of type DB2 is selected, only the additional policy item RESOURCE THRESHOLDS will be available.

The DB2 specific policy items are:

**DB2 CONTROL**

With this policy item you can specify additional data for a DB2 subsystem. For more information refer to “More about Policy Item DB2 CONTROL.”

**RESOURCE THRESHOLDS**

With this policy item you can manage your DB2 resource threshold definitions for the current DB2 subsystem. These will overwrite the default threshold definition - if any - for DB2 subsystems running on the system where the current DB2 subsystem will run. You will reach the panel *DB2 Resource thresholds selection* which is equivalent to the CICS *Resource thresholds selection* panel shown in Figure 147 on page 204.

More about Policy Item DB2 CONTROL

With this policy item you can specify the additional data for a DB2 subsystem:
## Application Policy Object

### Commands: HELP

<table>
<thead>
<tr>
<th>Command</th>
<th>DB2 Control Entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entry Type</td>
<td>Application</td>
</tr>
<tr>
<td>PolicyDB Name</td>
<td>DATABASE_NAME</td>
</tr>
<tr>
<td>Entry Name</td>
<td>DB2</td>
</tr>
<tr>
<td>Enterprise Name</td>
<td>YOUR_ENTERPRISE</td>
</tr>
<tr>
<td>Subsystem</td>
<td>DB2</td>
</tr>
<tr>
<td>Subtype</td>
<td>MSTR</td>
</tr>
<tr>
<td>Subtype from APPLICATION INFO policy</td>
<td>More: +</td>
</tr>
</tbody>
</table>

Enter or update the following fields:

- **DB2 subsystem id**: ...
- **Active log data set name**: ...
- **Log full threshold**: __ Percentage full
- **Shutdown if indoubt**: __ YES NO blank
- **Process iteration delays** (HH:MM:SS):
  - **TSO logoff delay**: ...
  - **STOP tablespace delay**: ...
  - **Connection Monitor delay**: ...
  - **Terminate threads delay**: __ Cycles __
  - **Active log alert**: __ Threshold __
  - **Log offload interval**: ...

**Figure 149. DB2 Control Entries Panel**

### Subtype

This parameter describes the controller type. It can be entered on the APPLICATION INFO policy item of the application, or the application class on the Subtype field. The value shown is either the one specified on the application or the one specified on the application class to which the application is linked.

### DB2 subsystem id

This entry must correspond to the DB2 subsystem ID as defined in the system PARMLIB.

### Active log data set name

Enter the DSN for the active log data set that is checked before threshold checking is performed. This value will be compared with the DSN which appears in the DSNJ002I message. The final HLQ (.DSnn) is not required. Specify the name without quotes.

### Log full threshold

Specify the percentage the active log data set is allowed to become full, before the command corresponding to the DSNJ110E message is issued. 90 percent is the default value.

### Shutdown if indoubt

Specify YES or NO to indicate if indoubt threads should be checked before shutting down DB2. If you want shutdown to proceed without checking for indoubt threads, then enter YES. If not, then you can use precheck on the NetView shutdown panel to check for indoubt threads.

### Process iteration delays

Enter the following values in the format HH:MM:SS or HH:MM:.SS. Except for cycles and thresholds, which must be a valid 2-digit number.

### TSO logoff delay

This is the amount of time that the TSO logoff request should allow for before cancelling a TSO session. This is used, for example, during Thread Termination. 1 minute is the default.
STOP tablespace delay
This is the amount of time to delay between STOP tablespace request
cycles. There are up to 3 cycles performed to ensure that the tablespace is
shutdown. 2 minutes is the default.

Connection Monitor delay
This is the amount of time to delay between connection monitoring cycles.
30 minutes is the default.

Terminate threads delay
This is the amount of time to delay between thread termination cycles. 2
minutes is the default.

Terminate threads cycles
This is the number of cycles to attempt at terminating threads. 6 attempts
is the default.

Active log alert
This is the amount of time to allow between DSNJ111E messages before
issuing an alert. 5 minutes is the default.

Active log alert threshold
This is the number of DSNJ111E alerts to allow for, prior to issuing any
associated commands from the ACF for the DSNJ111E message entry. 5
alerts is the default.

Log offload interval
This is the amount of time to allow between DSNJ115I messages before
issuing an alert plus any associated commands from the ACF for the
DSNJ115I message entry. 30 minutes is the default.

More about Policy Item DB2 RESOURCE THRESHOLDS
With this policy item you can define DB2 resource thresholds.

![Figure 150. DB2 Resource Thresholds Selection Panel]

From this panel you can do the following:
• Add a DB2 resource threshold definition by entering a name on one blank line
   and entering an S in front of the entry to select it to edit the thresholds. For the
   rules that apply to the DB2 resource threshold names, refer to the panel’s online
   help.
Application Policy Object

Pressing enter after typing a name and the S brings you to the Thresholds Definition panel shown in Figure 24 on page 33 where you can define thresholds.

- Rename DB2 resource thresholds by typing the new name over the current name.
- Delete DB2 resource thresholds by blanking out the name. All automation policy for that resource threshold will be deleted.
- Edit the thresholds for the DB2 resource thresholds by entering an S in the input field before the resource name. Note that you can select multiple entries in which case the thresholds panel will be displayed once for each selection, without having to return to this panel between selections.

Note:
If the update to any DB2 resource thresholds results in all of the thresholds being deleted then these resources will no longer be found in the ACF created during the next 'build'. This can result from one of the following scenarios:

- Creating a new resource threshold name by entering the name in one of the null entries but not entering S to select the entry to define the thresholds.
- Editing an entry and resetting all the thresholds to blank.

Policy Items for IMS Subsystems

Figure 151 shows the Policy Selection panel for applications that have been created with application type IMS (see entry field Application Type in Figure 105 on page 145). This Policy Selection panel contains additional policy items for an IMS subsystem.

Some of these policy items are only applicable to an IMS subsystem defined as an instance. If a subsystem CLASS of type IMS is selected, only the additional policy items IMS CONTROL and RESOURCE THRESHOLDS will be available.

| ACTIONS HELP |
|------------------|------------------|
| Command ====
AOFGEPOl | Policy Selection | Row 14 to 24 of 24 |
| SCROLL===
PAGE |
| Entry Type: Application | PolicyDB Name: DATABASE_NAME |
| Entry Name: IMS | Enterprise Name: YOURENTERPRISE |
| Action | Policy Name | Policy Description |
| ... | ----------------- | -------------------- |
| ______ | IMS CONTROL | Define IMS subsystem specific data |
| ______ | STATE ACTION TABLE | Select IMS state action table |
| ______ | RESOURCE THRESHOLDS | Define IMS resource thresholds |
| ______ | ... | |
| ************************************************ | Bottom of data ************************************************** |

Figure 151. Policy Selection Panel for IMS Subsystems

The IMS specific policy items are:

**IMS CONTROL**

Use this policy item to define additional IMS information for the IMS CONTROL REGION subsystem. This is described in System Automation for z/OS IMS Automation Programmer’s Reference and Operator’s Guide in section
IMSCNTL. This policy item is only defined for an IMS control region. For more information refer to "More about Policy Item IMS CONTROL."

**RESOURCE THRESHOLDS**

With this policy item you can manage your resource threshold definitions for this IMS subsystem. It will override any other system default threshold definitions. You will reach the panel *IMS Resource thresholds selection* which is similar to the *CICS Resource thresholds selection* panel shown in Figure 147 on page 204 where you can fill in the required information.

**STATE ACTION TABLE**

If you select this policy item, SA z/OS will display a list of state/action tables which previously have been defined as described in "IMS State/Action Tables" on page 247. From this list you can link one state/action table to the current IMS subsystem or change or remove an existing link.

**More about Policy Item IMS CONTROL**

If you select policy item IMS CONTROL from the *Policy Selection panel* for IMS subsystems, and you have specified the subtype CTL, Figure 152 on page 210 where you describe the IMS/DC control region is displayed.
## Application Policy Object

**Figure 152. IMS Control Region Specifications Panel for subtype CTL**

The fields that you enter or update have the following meaning:

### Subtype

This parameter describes the controller type. It can be entered on the APPLICATION INFO policy item of the application, or the application class on the **Subtype** field. The value shown is either the one specified on the application or the one specified on the application class to which the application is linked.

```plaintext
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Major node1</td>
<td>Activate IMS VTAM major node</td>
</tr>
<tr>
<td>Major node2</td>
<td>Activate other VTAM major node</td>
</tr>
<tr>
<td>Other major1</td>
<td>Activate VTAM major node</td>
</tr>
<tr>
<td>Other major2</td>
<td>Activate other VTAM major node</td>
</tr>
<tr>
<td>Auto-Operator</td>
<td>NetView Auto-operator</td>
</tr>
<tr>
<td>Shutdown exit</td>
<td>User exit name</td>
</tr>
<tr>
<td>XRF alternate start</td>
<td>YES NO</td>
</tr>
<tr>
<td>XRF new alternate start</td>
<td>YES NO</td>
</tr>
</tbody>
</table>
```

### XRF details if XRF = YES

The following definitions are for SA13 only:

```plaintext
<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>XRF Synchronization</td>
<td>Y S N P T</td>
</tr>
<tr>
<td>Default XRF start type</td>
<td>XRF ACTIVE BACKUP</td>
</tr>
<tr>
<td>XRF shut delay</td>
<td>Seconds</td>
</tr>
<tr>
<td>XRF Watch wait</td>
<td>Seconds</td>
</tr>
<tr>
<td>XRF unlock delay</td>
<td>Minutes</td>
</tr>
<tr>
<td>Non automatic takeover</td>
<td>YES NO</td>
</tr>
<tr>
<td>Partner subsystem ID</td>
<td>IMS subsystem ID of partner IMS</td>
</tr>
<tr>
<td>Partner domainID</td>
<td>Netview domain ID of partner IMS</td>
</tr>
<tr>
<td>Partner subsystem</td>
<td>Subsystem name of backup control region</td>
</tr>
<tr>
<td>XRF Watch notify</td>
<td>Seconds</td>
</tr>
</tbody>
</table>
```

The fields that you enter or update have the following meaning:
APPLid
identifies the receiver/sender name when the program-to-program interface is initiated and used. Use one of the following:

Applid
Use this keyword if this subsystem is defined as non-XRF.

(generic-applid,specific-applid)
If this subsystem is defined as XRF-capable, then use
generic-applid as the name given to both the active and alternate subsystem and specific-applid as the specific name for this subsystem.

Appl1/Appl2 connect
If you want SA z/OS automation to check that the IMS APPLID is connectable prior to starting IMS, specify a value here for the IMS APPLID. APPLID1 or APPLID2 identifies the VTAM name. Use the specification APPLID1 for a non-XRF IMS subsystem. If the system is an XRF IMS, use both the APPLID1 and the APPLID2 specifications to define both APPLIDs that can be used by the IMS subsystem. APPLID1 will be checked when the HSBID is 1, APPLID2 will be checked when the HSBID is 2.

Default HSBID
For an XRF IMS control region, the DEFLTHSB specification describes the preferred HSBID to be used when restarting an IMS system in situations where it does not matter which HSBID is used (in terms of IMS recovery).

Startup parm1/2
If you have additional parameters that should be appended to the startup command to start IMS, then specify them here. For a non-XRF IMS subsystem, specify your local parameters using the PARMS1 parameter. For an XRF IMS subsystem, the parameters used will be based on the current HSBID. For example, if HSBID=1, then the parameters coded on the PARMS1 parameter will be used.

Shutgo
Specifying YES will allow shutdown of a control region to progress from a normal shutdown, then to an immediate shutdown, then to a forced shutdown (as needed) without any operator action.

IMS Subsystem ID
Specifies the subsystem ID (IMSID) specified in the IMS control region parameters or SYSGEN.

Default start type
This defines the default start type for the Start Functions panel, to be used for a control region startup. This start type will be used when SA z/OS determines that the subsystem should be running but is not. If the SA z/OS INGREQ command is issued to perform a start of a subsystem, the default start type will be used If a default is not specified, the start will fail.

Default force
This parameter is used during an escalation of shutdown from type IMMED to type FORCE. It retrieves the default shut options for a type FORCE shutdown. This parameter specifies the dump option associated with a type FORCE shutdown.

Applid connect
During IMS startup, if this is specified as YES, then the IMS control region
Application Policy Object

will not be started unless the APPLID is connectable. In any case, an XRF alternate subsystem will not be started regardless of what is specified if the APPLID is not connectable.

Broadcast delay
If the BROADCAST or ABORT SHUTDOWN option is selected on one of the shutdown panels, this parameter defines the time in minutes between the execution of the /BRO command and the start of the shutdown (SHUTSYS) process.

Force transfer delay
This is used to specify the escalation interval in minutes between the transfer from the shutdown type of IMMED to the shutdown type of FORCE. The value specified in this timer overrides the value specified in the shutdown verify delay.

Immed transfer delay
This is used to specify the escalation interval in minutes between the transfer from the shutdown type of NORM to the shutdown type of IMMED. The value specified in this timer overrides the value specified in the shutdown verify delay.

Region recovery delay
After the first indication of an IMS control region abend (receipt of the DFS629 message), the number of minutes before the final abend recovery routine is invoked.

Shutdown verify delay
After the IMS shutdown checkpoint command is issued, the number of minutes before the shutdown verification process is invoked.

Start monitoring delay
This parameter is used for IMS initialization. This timer defines the interval for IMS Automation to wait before proceeding with IMS subsystem monitoring. IMS Automation will wait this interval before scheduling the function to report on the progress of the startup of an IMS complex.

Start type (shutdown)
This parameter defines the start type that will be used when a control region is restarted as part of a shutdown request.

Start type (recovery)
This parameter defines the start type that will be used when a control region is restarted after a recoverable abend has occurred.

DB control
This parameter indicates whether the control region is a DB control region. (If YES, then XRF=NO must also be specified.)

XRF
This parameter is required for the control region only. It specifies whether the control region will be executing as an XRF complex.

XRF synchronization
This parameter is used for XRF systems only and controls the type and degree of region synchronization between the active and the alternate IMS subsystems. This parameter is effective when the parent IMS control region is the alternate. While the parent is the active IMS, the parameter is not used.

Region synchronization applies only to Batch Message Processing regions (SUBTYPE=BMP), Message Processing regions (SUBTYPE=TP) and Fast Path regions (SUBTYPE=FP).
Application Policy Object

Y When coded on a control region subsystem entry, this parameter permits synchronization of the dependent regions. This parameter must be coded on a control region for any of the dependent regions to be synchronized. When coded on a dependent region, this parameter determines that when the partner region is stopped or started on the active IMS subsystem, this region is also stopped or started on the alternate IMS subsystem to mirror the active.

S This parameter is not valid when SUBTYPE=CTL. When the partner region is started on the active IMS subsystem, this event is recorded. The region will be started on the alternate when a takeover occurs after takeover processing is complete.

N When coded on a control region subsystem entry, this parameter disables all synchronization of the dependent regions. When coded on a dependent region, the region will not participate in any type of synchronization.

P This parameter is not valid when SUBTYPE=CTL. This region will not participate in any type of synchronization; however, it will be started when the alternate IMS is started.

T This parameter is not valid when SUBTYPE=CTL. This region will be started after the IMS Takeover.

Default XRF start type
This parameter is used for XRF systems only and determines whether the control region is to be started as the Active or Backup. Automation uses the status file to store the HSBID of the control region that is the Active system, for use when restarting the IMS system. When this information does not exist in the status file, this parameter determines which subsystem will be started as the Active system.

XRF shut delay
This parameter is used for XRF systems only. During shutdown of an XRF IMS complex with an alternate that is executing, IMS Automation will attempt the shutdown of the alternate prior to the active. For an "Immediate" type of shutdown, shut delay is the time in seconds to wait after initiating alternate shutdown before continuing with the active shutdown process.

XRF Watch wait
This parameter is used for XRF systems only. During XRF takeover processing, this time is the number of seconds between each invocation of takeover watch or monitor processing.

XRF unlock delay
This parameter is used for XRF systems only. During an XRF takeover, once the UNLOCK command has been issued by the IMS Automation Surveillance processing this parameter specifies the number of minutes to suspend the surveillance CLIST. After the suspension, the CLIST is invoked again to check XRF takeover processing.

Non automatic takeover
This parameter is used for XRF systems only. If the XRF complex is defined so that a takeover will not automatically occur (specified in the DFSH5Bxx member of the IMS PROCLIB), a message will be issued to the MVS system console so the operator can respond to allow the takeover to occur or to stop the process. If the parameter specifies YES, then IMS Automation will respond with the affirmative reply to this message.
Partner subsystem ID
This parameter is used for XRF systems only and is required if XRF=YES. This is the subsystem ID for the partner IMS control region. This control region may be specified either in the local (same) or a remote domain.

Partner domainid
This parameter is used for XRF or FDR enabled systems only. For an XRF environment specify this parameter and XRF=YES. For an FDR enabled system specify this parameter and XRF=NO. This is the NetView domainID for the partner IMS control region in an XRF environment, or, the tracking FDR region in an FDR environment. This may be the local (same) or a remote domain.

Partner subsystem
This parameter is used for XRF or FDR enabled systems only. For a XRF environment specify this parameter and XRF=YES. For an FDR enabled system specify this parameter and XRF=NO. For the XRF CONTROL region this is the SA z/OS subsystem name of the alternate/backup control region. For the dependent regions this is the SA z/OS subsystem name of the corresponding XRF partner dependent regions. For an FDR enabled system, this is the SA z/OS subsystem name for the FDR region.

XRF Watch notify
This parameter is used for XRF systems only and is required if XRF=YES. This parameter can be used to tailor the XRF takeover surveillance processing. It is specified in seconds. If surveillance processing does not detect a change in the XRF status for the monitored IMS subsystem prior to the time specified in HERE, additional IMS Automation warning messages will be issued. These messages describe possible problems with the takeover. Once issued, the clock starts again and the messages are not issued until the time spent in surveillance once again exceeds the time specified in watch notify.

XRF Watch alert
This parameter is used for XRF systems only and is required if XRF=YES. This parameter can be used to tailor the XRF takeover surveillance processing. It is specified in seconds. Once surveillance processing exceeds the time specified in this parameter with no change in the XRF status, additional messages are issued indicating that operator intervention may be required to complete the takeover. Once issued, the messages will be issued again every XRF Watch wait time until an XRF status change is detected.

The following definitions apply only to SA OS/390 1.3:

**Major node1/2**

SA OS/390 1.3 users: If you want SA z/OS automation to check that the IMS VTAM Major Node is active (and make it active if it is not) prior to starting the IMS control region, then specify the correct major node here. The specification for an XRF IMS subsystem is similar to that for APPLIDs.

SA OS/390 Version 2 users: Specify the VTAM V NET,ACT,ID=xxx or V NET,INACT,ID=xxx commands for the required nodes in the Prestart and the Postshutdown command entries for each start type of the IMS control region policy. Alternatively, if you wish the startup/shutdown to wait until the node is active/inactive use the following commands:

- Active: PIPE CORRCMD V NET,ACT,ID=xxx | CONS
- Inactive: PIPE CORRCMD V NET,INACT,ID=xxx | CONS
Other major1/2

SA OS/390 1.3 users: If you want SA z/OS automation to check that any other VTAM Major Node is active (and make it active if it is not), then specify the correct node name here. The specification for an XRF IMS subsystem is similar to that for APPLIDs.

SA OS/390 Version 2 users: Specify the VTAM V NET,ACT,ID=xxx or V NET,INACT,ID=xxx commands for the required nodes in the Prestart and the Postshutdown command entries for each start type of the IMS control region policy. Alternatively, if you wish the startup/shutdown to wait until the node is active/inactive use the following commands:

- Active: PIPE CORRCMD V NET,ACT,ID=xxx | CONS
- Inactive: PIPE CORRCMD V NET,INACT,ID=xxx | CONS

Auto-Operator

This identifies the NetView Automated Operators that control this IMS system and its dependents. The value must be defined in an AUTOOP statement. Operators can be assigned to one or more IMS subsystems.

Shutdown exit

This parameter specifies that a user exit will be called. This user exit is invoked only at shutdown of the IMS control region. The user exit must be a CLIST and a member of the DSICLD concatenation of NetView.

XRF alternate start

This parameter is used for XRF systems only and is required if XRF=YES. When the control region is started and is the active in an XRF complex, this parameter specifies whether the alternate IMS control region should also be started immediately after the active becomes available.

When ALTSTART=YES is used, any service periods or triggers that are defined for the XRF backup or alternate subsystem will be ignored.

Note:

It is recommended that the same service periods are used for both XRF subsystems when ALTSTART=YES.

XRF new alternate start

This parameter is used for XRF systems only. After an XRF takeover, this parameter specifies whether a new alternate should be started to replace the old active IMS subsystem.

If the specified subtype is not CTL, the panel shown in Figure 153 on page 216 is displayed.
Application Policy Object

![Figure 153. IMS Dependent Region Specifications Panel](image)

Control region name

Is the SA z/OS subsystem name of the owning control region. This parameter is necessary only when a dependent region is defined as a dependent of more than one IMS control region. This parameter enables IMS Automation to determine how to display, start, and stop the region. You may include system symbols and system automation symbols. See "Assigning System Automation Symbols (AOCCLEONE)" on page 42 for details about using system automation symbols.

Default force

This parameter is used during an escalation of shutdown from type IMMED to type FORCE. It retrieves the default shut options for a type FORCE shutdown. This parameter specifies the dump option associated with a type FORCE shutdown.

Broadcast delay

If the BROADCAST or ABORT SHUTDOWN option is selected on one of the shutdown panels, this parameter defines the time in minutes between the execution of the /BRO command and the start of the shutdown (SHUTSYS) process.

XRF enabled

Specifies whether XRF synchronization should be enabled or not.

Partner subsystem

This parameter is used for XRF enabled systems only. For IMS dependent regions this is the SA z/OS subsystem name of the corresponding XRF partner dependent regions.

XRF synchronisation

This parameter is used for XRF systems only and controls the type and degree of region synchronization between the active and the alternate IMS subsystems. This parameter is effective when the parent IMS control region is the alternate. While the parent is the active IMS, the parameter is not used.

Y When the partner region is stopped or started on the active IMS subsystem this region is also stopped or started on the alternate IMS subsystem to mirror the active.
This region will not participate in any type of synchronization.

This region will not participate in any type of synchronization, however, it will be started when the alternate IMS is started.

When the partner region is started on the active IMS subsystem, this is recorded. The region will be started on the alternate when a takeover occurs after takeover processing is complete.

This region will be started after the IMS takeover.

### Policy Items for OPC Subsystems

Figure 154 shows the Policy Selection panel for applications that have been created with application type OPC (see entry field Application Type in Figure 105 on page 145). This Policy Selection panel contains additional policy items for an OPC subsystem.

These policy items are only applicable to an OPC subsystem defined as an instance. If a subsystem CLASS of type OPC is selected, no additional policy items will be available.

<table>
<thead>
<tr>
<th>ACTIONS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFGEPOL</td>
<td>Policy Selection</td>
</tr>
<tr>
<td>Command</td>
<td>SCROLL</td>
</tr>
</tbody>
</table>

Entry Type: Application
Entry Name: OPC
PolicyDB Name: DATABASE_NAME
Enterprise Name: YOUR_ENTERPRISE

Action | Policy Name | Policy Description  
--- | --- | ---
... | | 
----------------- | | 
OPC CONTROL | Define OPC Control entries 
----------------- | | 
... | | 

Figure 154. Policy Selection Panel for OPC Subsystems

The OPC specific policy item is:

**OPC CONTROL**

Use this policy item to specify OPC extensions to subsystem definitions. For more information, see "More about Policy Item OPC CONTROL."

**More about Policy Item OPC CONTROL**

If you select policy item OPC CONTROL on the Policy Selection panel for OPC subsystems, SA z/OS displays the following panel where you can describe additional information for OPC subsystems.
Fill in the following fields:

**Subtype**
This parameter describes the controller type. It can be entered on the APPLICATION INFO policy item of the application, or the application class on the Subtype field. The value shown is either the one specified on the application or the one specified on the application class to which the application is linked.

**Subsystem ID**
Enter the OPC subsystem ID.

**Controller ID**
Enter the OPC subsystem ID of the controller, controlling this tracker. This field is only used for tracker definitions. It allows for the specification of a remote controller. If the controller is not in the same sysplex as the tracker, then specify this field.

**API LU Name**
Enter the Logical Unit name of the controller, controlling this tracker. This field is only used for tracker definitions. It allows for the specification of a remote controller. If the controller is not in the same sysplex as the tracker, then specify this field.

For both the Controller ID and API LU Name fields, if the controller is in a remote sysplex or system that can be controlled by SA z/OS with the TARGET= parameter of the commands, then there is no need to specify these fields.

**Policy Items for USS Subsystems**

By using the USS CONTROL policy item you can define UNIX System Services specific data.
More about Policy Item USS CONTROL

If you select policy item USS CONTROL on the Policy Selection panel for USS subsystems, SA z/OS displays the following panel where you can enter additional information for UNIX subsystems.

Figure 156. Policy Selection Panel for USS Subsystems

COMMANDS HELP
---------------------------------------------
AOGUSS         US Control Specification
Command ===>  More: +

Entry Type : Application  PolicyDB Name : DATABASE_NAME
Entry Name : USSTEST      Enterprise Name : YOUR_ENTERPRISE

Subsystem name: USSTEST

User ID . . . . . . .  

Monitoring Command.

Enter or update one of the following fields:

Command/Path. . . .

File Name . . . . .

Monitored Port. . . .

Figure 157. US Control Specification Panel

Fill in the following fields:

User ID

Enter the UNIX user ID this resource belongs to.

Monitoring Command

Enter the UNIX System Services monitoring command that is used to get the status of the resource. This command must be a valid program in the
Application Policy Object

HFS. If the field is left blank, the SA z/OS supplied standard monitoring routine is used. The following return codes are possible:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The resource is available</td>
</tr>
<tr>
<td>4</td>
<td>The resource is starting</td>
</tr>
<tr>
<td>8</td>
<td>The resource is unavailable</td>
</tr>
<tr>
<td>12</td>
<td>An error occurred</td>
</tr>
</tbody>
</table>

If the first character is a ‘/’, the specified command is considered to be an absolute path. Otherwise the monitoring routine must reside in the same directory as the SA z/OS routines.

Command/Path
Enter the name of the command that is running without its arguments as shown by the UNIX command "ps" with the format specification "comm".

File Name
Enter the path of a file in the HFS. If the file exists and the user ID specified above is the owner of the file the resource is available, otherwise the resource is not available.

Monitored Port
Enter the TCP port number this resource is supposed to listen on the local host.

Defining VTAM Application Subsystems

VTAM applications may require recovery commands to be issued should either VTAM be restarted, or the VTAM application node be reactivated. These commands differ from subsystem to subsystem and can be specified in the Messages/User Data policy items as described in the following section.

Registering VTAM Application Subsystems with SA z/OS Recovery

To enable VTAM application recovery to take place, the subsystems must be registered with the SA z/OS recovery code. This is achieved by using the INGVTAM command which is described in System Automation for z/OS Programmer’s Reference. The following application policy items must be customized.

1. PRE-START policy

   The PRE-START policy must have at least a NORMal start item with the INGVTAM command to activate a list of major nodes. The following command can be used as an example:

   \[
   \text{INGVTAM \&SUBSAPPL REQ=ACTIVATE MAJNODE=(majnode1,majnode2,...)}
   \]

   Where \( \text{majnode} \) are VTAM application major nodes. Each major node will be varied active to VTAM when the subsystem prestart commands are issued. Note, it is expected that only one of the major nodes will contain the minor node that the VTAM application subsystem will use.

2. POST-SHUTDOWN policy

   The POST-SHUTDOWN policy is used to deregister the subsystem with SA z/OS VTAM application recovery. Use the INGVTAM \( \text{REQ=DEACTIVATE} \) command in the policy. For example:

   \[
   \text{INGVTAM \&SUBSAPPL REQ=DEACTIVATE}
   \]

   The DEACTIVATE request issues a vary net inactive for each major node registered by the \( \text{REQ=ACTIVATE} \). The vary is not done if the major node is
shared by other subsystems that have also registered the major node. When the last subsystem registered issues an INGVTAM REQ=DEACTIVATE, the major node will be varied inactive. The only exception to this is when the major node contains model resources with wildcards in the node definition. In this case the major node is never inactivated.

3. ACORESTART Messages/User Data policy
   The ACORESTART message policy must have the same definition as the PRE-START policy. This policy item is used to reregister the subsystem with SA z/OS VTAM application recovery.

4. NODEUP Messages/User Data policy
   Enter commands that are issued when the minor node for the subsystem is reactivated or message IST093I is issued. Typically these commands reopen the VTAM ACB that the subsystem uses to communicate with VTAM.

5. VTAMUP Messages/User Data policy
   Enter commands that are issued when the VTAM subsystem is restarted. Typically these commands reopen the VTAM ACB that the subsystem uses to communicate with VTAM.

6. Relationships policy
   Optionally enter a relationship for the subsystem to ensure that the prestart commands are only issued when VTAM is up. The required relationship is:
   \[\text{PrepAvailable(WhenAvailable), Passive, Weak} \rightarrow \text{VTAM/APL=}\]
   
   Where VTAM is the name of the VTAM subsystem and that is the supporting resource. Passive forces the relationship to wait until VTAM is UP. Weak specifies that only the supporting resource status is checked.

7. UP Messages/User Data policy
   Enter a command in the UP message for the VTAM subsystem. With this command, use the following parameter:
   \[\text{INGVTAM REQ=ACTIVATE}\]

   Note, that no subsystem is specified on the INGVTAM command. This will force the INGVTAM command to issue the VTAMUP message against every subsystem that has registered with SA z/OS VTAM application recovery.

---

**Events Policy Object**

Events basically are used for external dependencies where they represent a process that the automation cannot handle. For example, an operator needs to manually edit a file before the application can be started. SA z/OS remembers which events are SET (have occurred) and can be told when to forget them (by defining an UNSET condition).

**Creating a New Event**

To define a new event, enter 44 or EVT for events in the panel shown in Figure 2 on page 7.

This displays a panel similar to the one shown in Figure 158 on page 222. This panel may show a list of already existing events. Enter the command new on the command line.
A panel similar to the one shown in Figure 159 is displayed. Specify the event name, the UNSET condition, and a description.

The unset condition is preset to NONE. You can change this input field if desired. For more information on the unset condition, see “More about Policy Item CONDITION” on page 223.

**Policy Items for Events**

Figure 160 shows the policy items that are displayed when an event is selected from the Entry Name Selection panel for events.
CONDITION
With this policy item you can specify when SA z/OS should unset (forget) a previously set event.

More about Policy Item CONDITION
Selecting the CONDITION policy item will bring up a panel as shown in Figure 161 where you can define the unset condition of the current event.

![Figure 161. Unset Condition Panel for Events](image)

The unset condition determines when an event for an application is reset to not occurred (that is, unset). You can enter the following values:

**NONE**
Once set, the event will never be unset.

**START**
Once set, the event will be unset if the application is already in the respective state or if the observed status of the application turns into STARTING.

**UP**
Once set, the event will be unset if the application is already in the respective state or if the observed status of the application turns into one of the following:
- AVAILABLE
- DEGRADED

**DOWN**
Once set, the event will be unset if the application is already in the respective state or if the observed status of the application turns into one of the following:
- SOFTDOWN
- HARDDOWN
- SYSGONE
- STANDBY

Service Periods Policy Object
Service periods define the availability of applications. A service period is a set of time intervals (service windows) during which an application should be active or inactive.

![Figure 162 on page 224](image) shows the policy items that are displayed when an existing Service Periods entry is selected from the Entry Name Selection panel.

Chapter 5. Defining Automation Policy  223
The unique policy item for Service Periods is:

**SERVICE WINDOW**

With this policy item you define the time frame (or frames) when an application or an application group should be available or unavailable. You need to link a service period with a defined service window to Applications, Application Groups, or Monitor Resources.

More about Policy Item SERVICE WINDOW

Selecting the SERVICE WINDOW policy item will bring up a panel as shown in Figure 163 on page 225 where you can define the time duration of the service period.
A service period is a schedule of times (referred to as service windows) when a resource is expected to be active or inactive. Times that you enter into a line preceded by letter U will be used as From/To times for the periods when the resource should be available (UP). A line preceded by letter D will contain From/To times for DOWN periods. Start and Stop times are specified in 24 hour notation, 00:00-24:00, and in any order. For example, to keep an application continuously available from Monday through Tuesday around the clock, specify the times as shown:

<table>
<thead>
<tr>
<th>Day</th>
<th>From</th>
<th>To</th>
<th>From</th>
<th>To</th>
<th>From</th>
<th>To</th>
<th>From</th>
<th>To</th>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday</td>
<td>00:00</td>
<td>24:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tuesday</td>
<td>00:00</td>
<td>24:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wednesday</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thursday</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Friday</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Saturday</td>
<td>08:00</td>
<td>14:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sunday</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weekday</td>
<td>07:00</td>
<td>18:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weekend</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Daily</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The schedule is defined by specifying one or more (up to five) service windows for:
- **each day of the week**: Monday through Sunday
- **weekday days**: the weekday schedule (Monday through Friday)
- **weekend days**: the weekend schedule (Saturday and Sunday)
- **Daily**: all seven days; the daily schedule
Service Periods Policy Object

For the time between the specified intervals, the automation manager decides what to do with the resources.

For each UP and DOWN schedule, in the P column, you may specify a priority:

<table>
<thead>
<tr>
<th>Letter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>for low priority; Orders resulting from a service period with low priority can be overruled by an operator's INGREQ command. This is the default.</td>
</tr>
<tr>
<td>H</td>
<td>for high priority; H schedules override L schedules and low priority operator requests.</td>
</tr>
</tbody>
</table>

This priority is checked in case two goals get into conflict. The hierarchy of requests and priorities is implemented as follows:

<table>
<thead>
<tr>
<th>Request from</th>
<th>wins over</th>
<th>request from</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service period high</td>
<td>=========&gt;</td>
<td>service period low, INGREQ low</td>
</tr>
<tr>
<td>INGREQ low</td>
<td>=========&gt;</td>
<td>service period low</td>
</tr>
<tr>
<td>INGREQ high</td>
<td>=========&gt;</td>
<td>service period high, service period low, INGREQ low</td>
</tr>
</tbody>
</table>

Note:

There are other methods of generating requests within SA z/OS. For a complete description of these methods and the implemented hierarchy of their priorities refer to the documentation of the SA z/OS concepts in System Automation for z/OS User's Guide.

When defining a service period keep in mind that the startup time is the time at which SA z/OS is to begin startup processing. It is not the time at which the resource will be available. The time until availability depends on how long the startup of an application or an application group takes.

According to the goal driven concept, a vote to start a resource is generated from a service period that you enter into an in a UP service window. A vote to stop a resource is generated from each service period that you enter into a DOWN service window. These votes have to compete with other votes that have been probably issued (for example, due to operator request). At the end of the service period, these votes are removed from SA z/OS. An end of an UP service window does not mean implicitly the start of a DOWN service window.

For more information about the hierarchy of service windows or the syntax of FROM and TO times, see the online help for the product.

If you later change a service period that is already active, and the service period is activated again, this may lead to conflicts and mismatches with overrides that have been made previously. So if you want to change a service period, check your existing NetView overrides that use this service period beforehand, so that no override is outside the new service period (see also the INGSCHED command, described in System Automation for z/OS Operator's Commands).

Triggers Policy Object

In combination with events, triggers are used to control the starting and stopping of applications on top of what is done with specified goals. You use events and triggers if external conditions must be fulfilled (such as a tape mount).
Figure 164 shows the policy items that are displayed when a trigger is selected from the Entry Name Selection panel.

<table>
<thead>
<tr>
<th>ACTIONS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFGEPOL</td>
<td>Policy Selection</td>
</tr>
<tr>
<td>Command</td>
<td>SCROLL</td>
</tr>
<tr>
<td>Entry Type : Trigger</td>
<td>PolicyDB Name : DATABASE_NAME</td>
</tr>
<tr>
<td>Entry Name : TRIGGER1</td>
<td>Enterprise Name : YOUR_ENTERPRISE</td>
</tr>
<tr>
<td>Action</td>
<td>Policy Name</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>Enter description</td>
</tr>
<tr>
<td>CONDITION</td>
<td>Define the Trigger condition</td>
</tr>
<tr>
<td>WHERE USED</td>
<td>List APLs and APGs using this entry</td>
</tr>
<tr>
<td>COPY</td>
<td>Copy data from an existing entry</td>
</tr>
</tbody>
</table>

The unique policy item for a trigger is:

**CONDITION**

Entering **S** in front of item **CONDITIONS** will bring up a panel as shown in Figure 165 on page 228 where you can define the startup and shutdown conditions for this trigger.

More about Policy Item **CONDITION**

With the **CONDITION** policy item you can:

1. Connect a service period to a trigger (1). Type **S** into the ACTION column in Figure 165 on page 228

   ![Figure 164. Policy Selection Panel for Triggers](image)

   **Note:**
   
   This feature is available only for compatibility with SA OS/390 1.3. Since SA OS/390 2.1, service periods are linked directly to application groups or applications.

2. Define up to ten startup conditions for the application that you want to connect to the trigger (2), see "Defining STARTUP and SHUTDOWN Conditions for a Trigger" on page 228

3. Define up to ten shutdown conditions for the application that you want to connect to the trigger (3), see "Defining STARTUP and SHUTDOWN Conditions for a Trigger" on page 228
**Note:**
Of course, you get an empty *Trigger Conditions* panel when you define conditions for a new trigger. The entries that you can see here are only visible after you have finished the tasks described in the subsequent sections.

- A maximum number of 20 condition names (service period and events) can be selected for each condition.
- The overall number of different events in a trigger is limited to 28.
- An event can be specified either in STARTUP conditions or in SHUTDOWN conditions of a particular trigger, however, not in both types of conditions simultaneously.

**Defining STARTUP and SHUTDOWN Conditions for a Trigger**

If you type an S in front of a STARTUP condition on the panel shown in Figure 165 (2), SA z/OS displays another panel like the one shown in Figure 166 on page 229.

<table>
<thead>
<tr>
<th>Action</th>
<th>Condition</th>
<th>Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>SERVICE PERIOD</td>
<td>SERVP1 1</td>
</tr>
<tr>
<td></td>
<td>STARTUP 1</td>
<td>SERVICE EVENT1 2</td>
</tr>
<tr>
<td></td>
<td>STARTUP 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>STARTUP 3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>STARTUP 4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>STARTUP 5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>STARTUP 6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>STARTUP 7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>STARTUP 8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>STARTUP 9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>STARTUP 10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SHUTDOWN 1</td>
<td>EVENT3 3</td>
</tr>
<tr>
<td></td>
<td>SHUTDOWN 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SHUTDOWN 3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SHUTDOWN 4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SHUTDOWN 5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SHUTDOWN 6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SHUTDOWN 7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SHUTDOWN 8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SHUTDOWN 9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SHUTDOWN 10</td>
<td></td>
</tr>
</tbody>
</table>

*Figure 165. Trigger Conditions Panel*
Figure 166 displays a list of events where you can select one or more events that should trigger the startup of an application. Typing an S will select an event and set its status to SELECTED. All selected events will be added to the currently selected startup condition.

- You can enter a description for the startup condition.
- Entering YES on the Service Period entry field connects the already selected service period to this condition. In this scenario, you can only connect service period SERVP1 to any of the conditions for TRIGGER1. Therefore, enter YES on the Service Period entry field only for STARTUP 1.

Note: This feature is available only for compatibility with SA OS/390 1.3. Since SA OS/390 2.1, service periods are linked directly to application groups or applications.

If you type an S in front of a SHUTDOWN condition on the panel shown in Figure 165, SA z/OS displays another panel similar to the one shown in Figure 166. The panel title in the panel displayed will now read SHUTDOWN Condition for Trigger and will contain a list of events which you can add to or remove from the currently selected shutdown condition.

The Service Period input field is inhibited when no service period has been selected for the trigger.

Processor Policy Object

Use the Processor Policy Object to define and control your processor hardware in the sysplex.

Creating a New Processor

How to create new policy objects in general is described in "How to Create New Policy Objects" on page 14. Figure 167 on page 230 shows the panel for defining a new Processor policy object.
**Processor Policy Object**

![Figure 167. Defining a New SA z/OS Processor Entry](image)

**Name**  The processor name must have 1 to 8 alphanumeric characters and must begin with an alphabetic character, #, $, or @. The name must be different from the names of all groups, subgroups, systems (including the processor operations names), and processors defined in the current policy database.

**Processor Type**  Specifies the processor machine type of the target hardware. Select a name from the list shown to specify the processor type.

Use the generic type name eServer to specify any zSeries or 390-CMOS processor including those shown in the list. The actual processor type is determined at SA z/OS processor operations run time when a target system associated with the target hardware is initialized.

Use the type PSM to specify a ProcOps Service Machine on VM as 'virtual hardware' for VM guest systems. The default is eServer. For more details about the PSM, see System Automation for z/OS Customizing and Programming.

**Mode**  Specifies the operating mode of the processor hardware as defined in the Reset Profile that is used for hardware activation. For zSeries, 390-CMOS in logical partition mode, and PSM processors specify LPAR.

---

**Policy Items for Processors**

The following figure shows the Policy Selection panel.

![Figure 168. Policy Selection Panel for Processor](image)
Processor Policy Object

The unique policy items for the Processor policy object are:

**PROCESSOR INFO**
Using this policy, you can specify the processor type and mode, the connection protocol and network information.

**LPARS AND SYSTEMS**
This policy is displayed for processors with mode LPAR.

**SYSTEMS**
This policy is displayed for processors with mode ESA. Use this policy to select a system.

**More about Policy Item PROCESSOR INFO**
A panel similar to **Figure 169** is displayed if you select the PROCESSOR INFO policy item from the Policy Selection panel for Processors.

---

**Figure 169. Processor Information Panel**

On the Processor Information panel you can specify the following definitions so that the automation knows where the system is located to send the commands to the appropriate support element.

**Processor Type**
Specifies the processor machine type of the target hardware. Select a name from the list to specify the processor type. Use the generic type name eServer to specify any zSeries or 390-CMOS processor including those shown in the list. The actual processor type is determined at SA z/OS processor operations runtime when a target system associated with the target hardware is initialized. This field is required.

**Mode**
Specifies the operating mode of the processor hardware as defined in the Reset Profile used for hardware activation. This field is required.

**Connection Protocol**
Indicates the protocol used for communication with the processor.

**INTERNAL**
Means connection via the operating system.

**NVC**
Means connection via SNA network.
**Processor Policy Object**

**SNMP**

Means connection via TCP/IP.

A processor with connection INTERNAL is not built into the processor operations control file. It is used for Parallel Sysplex automation in combination with applications of type IMAGE. For more details, see [System Automation for z/OS Customizing and Programming](#). With these definitions the automation knows where the system is located to send the commands to the appropriate support element.

**CPC Name / NAU**

For NVC and INTERNAL connections, this indicates the network addressable unit of the support element within the network. This field is required.

**Network Name**

For NVC and INTERNAL connections, this indicates the name of the SNA network in which the support element resides. This field is required. For SNMP connections, the central processor complex name of the support element is combined of Network Name and CPC Name.

**Community Name / Auth Token**

For SNMP and INTERNAL connections, this indicates the community name of the support element. For NVC connections, this indicates the authorization token. Both community name and authorization token are defined at the support element, and are used to validate command authorization. This field is required.

**Activation Reset Profile**

Indicates the Reset profile name to use when activating the target hardware. This profile name must match with the Reset profile name defined at the target hardware support element. This field is optional. The Reset profile name specified here will be used by ProcOps common command processing if the CNAME keyword is omitted from the ACTIVATE or LOAD command, and use of a Reset profile is appropriate.

If no Reset profile name is specified here, and the CNAME keyword is not used on the LOAD or ACTIVATE command, ProcOps omits the CNAME parameter from the command sent to the support element. In this case, the support element uses its default Reset profile.

**Path Poll Frequency**

Enter how often (in minutes) SA z/OS processor operations should poll the support element for status. Valid values are 0 to 99 minutes. Specify 0 to deactivate polling.

**Path Poll Retries**

Enter the number of attempts to retry that should occur before SA z/OS processor operations notifies the operator that the connection failed. Valid values are 0 to 99 retries.

**TCP/IP Address or Hostname for Support Element**

If SNMP connection is selected, then at least one address must be specified. If both are specified, the support element address has preference. HMC address alone may be considered if multiple processors should be controlled via one single connection.
More about Policy Item LPARS AND SYSTEMS

If you have created a Processor policy object with LPAR mode, the policy item LPARS AND SYSTEMS is available, and has the LPAR Definitions panel shown in Figure 170 associated with it.

![Figure 170. LPAR Definitions Panel](image)

You can specify the following:

**Action**

Action allows you to remove LPAR definitions and to invoke the system selection panel:

- DELETE (D) deletes the current LPAR definition. Confirmation on a separate panel is not required.
- SYSTEM (S) displays a selection list of all defined systems allowing a system to link to the current LPAR

**LPAR Name**

Specifies the name of the logical partition. Duplicate LPAR names without Target System Name are wiped out when the panel is left. However, it is possible to link different systems to one LPAR, so duplicate LPAR names are accepted when they have links to different systems. Linking one system to multiple LPARs is not accepted.

**Target Mode**

Specifies the mode of the system that is supposed to run in this LPAR. The Target Mode LINUX is used for hardware that supports LPARs for Linux only, where a reduced instruction set is provided and therefore cannot run MVS or VM. For a Linux system specify LINUX if the processor hardware is configured in 'LINUX Only' mode, otherwise specify ESA.

**Target System Name**

You can specify a system entry name, which establishes a "link" of this system to the partition. Any name is allowed here, it is not required that the system entry has already been defined. During build, such an undefined system is treated as not specified. If the system is already defined as SYS entry, it is checked that it is a valid selection according to the rules described in the following section for entry type SYS. Otherwise, an error message is displayed.
Processor Policy Object

Policy Items for PSM Processors

Figure 171 shows the Policy Selection panel for PSM processors.

<table>
<thead>
<tr>
<th>ACTIONS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>--------</td>
<td>--------------</td>
</tr>
<tr>
<td>AOFGEPOL</td>
<td>Policy Selection Entry created</td>
</tr>
<tr>
<td>Command ===</td>
<td>SCROLL=== PAGE</td>
</tr>
</tbody>
</table>

Entry Type : Processor PolicyDB Name : DATABASE_NAME
Entry Name : Proc1 Enterprise Name : YOUR_ENTERPRISE

Action Policy Name Policy Description
----------- ---------------
DESCRIPTION Enter description
--------------------
PSM INFORMATION Enter and display PSM processor information
GUESTS AND SYSTEMS Define VM guests and select systems
COPY Copy data from an existing entry

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

Figure 171. Policy Selection Panel for PSM Processors

The unique policy items for the Processor policy object are:

PSM INFORMATION
Using this policy, you can specify the processor type and mode, the
connection protocol and network information.

GUESTS AND SYSTEMS
This policy allows you to define VM guests and select systems. For more
details, see “More about Policy Item GUESTS AND SYSTEMS” on page 236.

More about Policy Item PSM INFORMATION
A panel similar to Figure 172 is displayed if you select the PSM INFORMATION
policy item from the Policy Selection panel for PSM Processors.

<table>
<thead>
<tr>
<th>COMMANDS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>---------</td>
<td>--------------</td>
</tr>
<tr>
<td>AOFGCPO1</td>
<td>ProcOps Service Machine Information</td>
</tr>
<tr>
<td>Command ===</td>
<td></td>
</tr>
</tbody>
</table>

Entry Type : Processor PolicyDB Name : DATABASE_NAME
Entry Name : PRLPR01 Enterprise Name : YOUR_ENTERPRISE

TCP/IP Address or Hostname...
MULTISYS.PSM.PR2.US

Message Server Port ..... 1234 (0 - 65535)
Command Server Port ..... 1234 (0 - 65535)
IP Stack ................. 1234
Security Resource Name .... X1234
Security Network Name ..... PRINEM
Community Name / Auth Token .. OK
Path Poll Frequency ..... 30 (0 to 99 minutes)
Path Poll Retries ......... 5 (0 to 99)
VM Host Target System Name .. PRL1
Service Machine Name ..... PRLPR01

Figure 172. ProcOps Service Machine Information Panel

On this panel you can update the following:

TCP/IP Address or Hostname
Indicates TCP/IP address or symbolic host name for communication with
the ProcOps Service Machine. This address is required.
Message Server Port
Indicates the port the ProcOps Service Machine Message Server uses for communication with the ProcOps receiver task.

Command Server Port
Indicates the port the ProcOps Service Machine Command Server uses for communication with the ProcOps send program.

IP Stack
Indicates the to be used IP stack. If not defined, the default IP stack is used.

Security Resource Name
This field corresponds to the CPC Name / NAU name for NVC and INTERNAL connections and is used for SAF checking. This field is required.

Security Network Name
This field corresponds to the Network Name for NVC and INTERNAL connections and is used for SAF checking. This field is required.

Community Name / Auth Token For TCPIP connections
This indicates the authorization token (ProcOps Service Machines are connected only via TCPIP). This field is required.

Path Poll Frequency
Indicates how frequently the support element should be queried for status. Frequency must be specified in whole minutes in a range from 0 to 99. A frequency of 0 means that polling is deactivated. This field is required.

Path Poll Retries
Indicates the number of times query should be retried before notifying the operator. Valid values are 0 to 99 retries. This field is required.

VM Host Target System Name
Indicates the name of the VM host system if it is also defined in ProcOps This field is optional. The number of PSMs per hosting VM system is restricted to 12.

Note:
The host command ISQXPSM can only be used to operate the PSM via the VM host if this information is provided.

ProcOps Service Machine Name
Indicates the name of the PSM as it is known to the VM host system.

Note:
The host command ISQXPSM can only be used to operate the PSM via the VM host if this information is provided.

For ProcOps Service Machines, a Mode and Connection Protocol field is not available because the Mode is always LPAR and the Connection Protocol is always TCPIP and neither value can be changed.
More about Policy Item GUESTS AND SYSTEMS
If you have created a Processor policy object of type PSM, the GUESTS AND SYSTEMS policy item is available, and has the GUEST Definitions panel shown in Figure 173 associated with it.

This panel allows you to define VM GUESTs and link systems to them. Within ProcOps, VM guest systems are treated like LPARs:

**Action**
Action allows you to remove VM GUEST definitions and to invoke the system selection panel:
- DELETE (D) deletes the current VM GUEST definition. Confirmation on a separate panel is not required.
- SYSTEM (S) displays a selection list of all defined systems allowing a system to link to the current VM GUEST

**GUEST Name**
Specifies the name of the VM GUEST. Duplicate VM GUEST names without Target System Name are wiped out when the panel is left. However, it is possible to link different systems to one VM GUEST, so duplicate VM GUEST names are accepted when they have links to different systems. Linking one system to multiple VM GUESTs is not accepted.

**Target Mode**
Specifies the mode of the system that is supposed to run on this VM GUEST. The Target Mode LINUX is used for hardware that supports VM GUESTs for Linux only, where a reduced instruction set is provided and therefore cannot run MVS or VM.

For a Linux system specify LINUX if the processor hardware is configured in 'LINUX Only' mode, otherwise specify ESA. Valid VM GUEST Target Modes are 370, ESA, and LINUX.

**Target System Name**
You can specify the name of the system that runs on this VM GUEST. Any
name is allowed here; it is not required that the system entry has already
been defined. During build, such an undefined system is treated as not
specified.

If the system is already defined as a SYS entry, it is checked that it is a
valid selection according to the rules described in the following section for
entry type SYS. Otherwise, an error message is displayed.

The number of guest systems per PSM is restricted to 60. When a new guest
system is defined, the number of guests that have already been defined for the
associated PSM is checked. If there are already 60 defined, any further guest
definition is rejected.

**Monitor Resource Policy Object**

The Monitor Resource (MTR) policy object allow you to obtain the health state of
an object in two different ways:

- Actively, by polling—that is executing a monitoring command periodically
- Passively, by processing events

Monitor resources are connected to application resources (APLs) or application
group resources (APGs). The health status of the monitored object is propagated to
the APLs or APGs and results in a health status there.

Selecting *Monitor Resources* (MTR) from the *Entry Type Selection* panel will bring
you to the *Entry Name Selection* panel that lists, if you have any, your current
monitor resources. From this panel you can either create new monitor resources, or
select an entry and action for one of the existing entries.

*Figure 174* shows the policy items that are displayed when an existing Monitor
Resources entry is selected from the *Entry Name Selection* panel.

<table>
<thead>
<tr>
<th>ACTIONS HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFGEPOL</td>
</tr>
<tr>
<td>Command</td>
</tr>
<tr>
<td>Entry Type</td>
</tr>
<tr>
<td>Entry Name</td>
</tr>
<tr>
<td>Action</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

*Figure 174. Policy Selection for Monitor Resource*

The unique policy items for the Monitor Resource policy object are:

**MONITOR INFO**

This option allows you to make definitions for the monitor resource. Refer
to "More about Policy Item MONITOR INFO" on page 238 for more
details.
Monitor Resource Policy Object

RELATIONSHIPS
This policy item has the same functionality for entry types Application, ApplicationGroup, and Monitor Resource. For each application, application group, or monitor resource you may specify several relationships. They will be used to automate the startup or stopping of those applications, application groups, and monitor resources. If you select this policy item, in the Relationship Selection List panel, you can view existing relationships or define new relationships for the current application, application group, or monitor resource.

You define relationships between dependent resources and supporting resources. In the customization dialog you define relationships always from the point of view of the dependent resource. So the resource that you have currently selected will become the dependent resource, when you define a new relationship.

Refer to "More about Policy Item RELATIONSHIPS" on page 132 for more information on viewing existing or defining new dependency relationships.

HEALTHSTATE
This option allows you to define the processing that takes place when the monitor resource detects a change of the health state of the monitored resource. Refer to "More about Policy Item HEALTHSTATE" on page 239 for more details.

SERVICE PERIOD
This option allows you to link a service period to the monitor resource. Its resources are then automatically started up or shut down in the time frame specified by the service period. Refer to "More about Policy Item SERVICE PERIOD" on page 241 for more details.

More about Policy Item MONITOR INFO
If you select the MONITOR INFO policy item from the Policy Selection panel for monitor resources, the Monitor Resource Information panel is displayed, as shown in Figure 175.

<table>
<thead>
<tr>
<th>COMMANDS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitor Resource Information</td>
<td>More: +</td>
</tr>
<tr>
<td>Command ==&gt;</td>
<td></td>
</tr>
<tr>
<td>Entry Type : Monitor Resource</td>
<td>PolicyDB Name : DATABASE_NAME</td>
</tr>
<tr>
<td>Entry Name : MONITOR1</td>
<td>Enterprise Name : YOUR_ENTERPRISE</td>
</tr>
<tr>
<td>Activate command...</td>
<td></td>
</tr>
<tr>
<td>Deactivate command..</td>
<td></td>
</tr>
<tr>
<td>Monitor command...</td>
<td></td>
</tr>
<tr>
<td>Monitoring Interval... (hh:mm or blank)</td>
<td></td>
</tr>
<tr>
<td>Captured Messages Limit... (0 to 999, or blank)</td>
<td></td>
</tr>
<tr>
<td>Owner...</td>
<td></td>
</tr>
</tbody>
</table>

Figure 175. Monitor Resource Information Panel

To create a new Monitor Resource entry, provide the following information:
Monitor Resource Policy Object

Activate Command
This field is optional and is only needed to establish the monitoring environment.

Deactivate Command
This field is optional and is only needed to establish the monitoring environment.

Monitor Command
This can be any NetView command. If a command and an interval is specified, the command will be executed periodically while the monitor is active.

If a command is specified and no interval is specified, the command will be executed once during the startup phase.

If no command is specified the monitor is a passive monitor. In this case, the health status must be obtained via the INGMON command (see System Automation for z/OS Programmer’s Reference for more details).

You may include system symbols and system automation symbols but they must be entered in uppercase letters (for example &AOCCLONE1). See “Assigning System Automation Symbols (AOCCLONE)” on page 42 for more details.

Monitoring Interval
Specifies an interval time for periodic execution of the monitor command. An Interval value of 0 will be forced to blank. If an Interval is specified a monitor command must also be specified.

Captured Messages Limit
Specifies the maximum number of captured messages that will be saved for the SA z/OS DISPMTR command. If the field is left blank the value can be inherited from the system default definition. If it is also not specified there, a default value of 0 will be used.

Owner
Specifies information for the operator about who to contact in case of error.

More about Policy Item HEALTHSTATE
If you select the HEALTHSTATE policy item from the Policy Selection panel for monitor resources, the Healthstate Processing panel is displayed, as shown in Figure 176 on page 240
You can use this panel to enter commands to be executed when the monitor resource detects a change of the healthstate of the monitored resource to one of the following values: NORMAL, WARNING, MINOR, CRITICAL, FATAL or blank (which means for any change).

**Healthstate**

Defines the health state that the monitored resource must change to in order to initiate the specified commands. Valid values are:

- NORMAL
- WARNING
- MINOR
- CRITICAL
- FATAL
- blank

Leaving this field blank means that the specified commands will be executed in any health state change.

**Automated Function**

Enter the name of the automated function under which the command is to run. If the field is left blank, the function defaults to the one currently running the command list.

**Command Text**

Enter the command or command list to be issued. Any system command must be prefaced with MVS.

Four substitution variables are available:

**EHKVAR1**
Contains the monitor name.

**Note:**
Entering a "*" enables return code checking for the specified command. This means that processing continues if the command returns a return code of 0; in all other cases the processing is stopped immediately.
EHKVAR2
Contains the current health status.

EHKVAR3
Contains the old health status.

EHKVAR4
Contains the message.

A health state can be specified on the call to ACFCMD, which will issue the corresponding command or Command Group defined in the CMD entry.

A health state of blank can be specified to ensure that the corresponding command is issued regardless of the SEL= parameter included on the ACFCMD call. Thus, for any given health state processing, the blank entries will be issued in addition to the entries coded for that particular healthstate.

More about Policy Item SERVICE PERIOD
If you select the SERVICE PERIOD policy item from the Policy Selection panel for monitor resources, the Service Period for Monitor Resource panel is displayed, as shown in Figure 177.

---

![Figure 177. Service Period for Monitor Resource Panel](image)

---

Here you can add service periods to or remove them from a monitor resource.

---

Product Automation Policy Object

Selecting Product Automation (PRD) from the Entry Type Selection panel will bring you to the Entry type selections for Product Automation panel shown in Figure 178 on page 242, where you can select the product for which you want to define automation policy. You can select from the following products:

- DB2, see “Defining Automation for DB2 Components” on page 242
- CICS, see “Defining Automation for CICS Components” on page 243
- IMS, see “Defining Automation for IMS Components” on page 247
- OPC, see “Defining Automation for OPC Components” on page 251
Defining Automation for DB2 Components

For DB2 components, the following entry type is available:

**(DEN) DB2 System defaults**

Use this option to define additional default information that is specifically required for DB2 automation within SA z/OS but not required for automation of other products or applications.

This entry type is described in more detail in the following section.

DB2 System Defaults

**Figure 179** shows the policy items that are displayed when you select a DB2 System Default object from the related **Entry Name Selection** panel or when you are creating a new DB2 System Default object.

The unique policy items for DB2 System Defaults are:

**DB2 SYSTEM DEFAULTS**

With this policy item you can specify whether for the system to which the current DB2 System Default is linked, data sharing should be active or not. You reach a panel similar to the one shown in **Figure 149 on page 206** where you can specify DB2 defaults for the linked system.
RESOURCE THRESHOLDS
With this policy item you can define default threshold definitions for DB2 subsystems running on the systems to which the current DB2 system default will be linked. You will reach the panel DB2 Resource thresholds selection which is similar to the CICS Resource thresholds selection panel shown in Figure 147 on page 204 where you can define the equivalent information for DB2 subsystems running on the linked systems.

Defining Automation for CICS Components
For CICS components, the following entry types are available:

(CSA) State/Action tables
With this option, you can define a new or edit an existing state/action table for CICS. This is described in more detail in "CICS State/Action Tables."

(CCN) Link definitions
Use this option to define the inter-region and inter-system communication (IRC and ISC) links so that they can be monitored by SA z/OS.

(CVP) Link Monitor periods
With this option, you can define service periods for the link monitoring definitions. Service periods can be related to a CICS link as defined with option (CCN) Link definition.

These options are described in more detail in the subsequent sections.

CICS State/Action Tables
Figure 180 shows the policy items that are displayed when you select a CICS state/action table entry from the related Entry Name Selection panel or when you are creating a new CICS state/action table entry. Starting from this panel, you can select state/action tables available to your CICS applications for specific problems.

```
<table>
<thead>
<tr>
<th>ACTIONS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFGEPOL</td>
<td>Policy Selection Row 1 to 5 of 5 SCROLL==&gt; PAGE</td>
</tr>
<tr>
<td>Command ==</td>
<td></td>
</tr>
<tr>
<td>Entry Type: CICS State/Action</td>
<td>PolicyDB Name : DATABASE_NAME</td>
</tr>
<tr>
<td>Entry Name: CICS_SA_1</td>
<td>Enterprise Name : YOUR_ENTERPRISE</td>
</tr>
<tr>
<td>Action</td>
<td>Policy Name</td>
</tr>
<tr>
<td></td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td></td>
<td>STATE/ACTION TABLE</td>
</tr>
<tr>
<td></td>
<td>WHERE USED</td>
</tr>
<tr>
<td></td>
<td>COPY</td>
</tr>
<tr>
<td></td>
<td>********************************************* Bottom of data *********************************************</td>
</tr>
</tbody>
</table>
```

Figure 180. Policy Selection Panel for CICS State/Action Tables

The unique policy item for CICS state/action tables is:

STATE/ACTION TABLE
Use this policy item to define the state/action tables that are available to your CICS applications. For more information refer to "More about Policy Item STATE/ACTION TABLE."

More about Policy Item STATE/ACTION TABLE: The CICS State/Action tables panel is displayed if you select policy item STATE/ACTION TABLE from the Policy Selection panel for CICS State/Action entries as shown in Figure 180. Use this
panel to define the state/action tables that are available to your CICS applications. You can link these entries to CICS applications, using the STATE ACTION TABLE policy item for CICS applications as described "Policy Items for CICS Subsystems" on page 199.

Fill in the following fields:

**Short on storage**
The State/Action table used for short on storage recovery.

**Storage violation**
The State/Action table used for storage violation recovery.

**Autoinstall problem**
The State/Action table used for autoinstall problem recovery.

**VTAM ACB problem**
The State/Action table used for VTAM ACB problem recovery.

**Link Definitions**
Figure 182 shows the policy items that are displayed when you create or modify a link definition for CICS from the related Entry Name Selection panel.

The unique policy items for CICS link definitions are:

**CICS CONNECTION**
This policy item lets you describe the interregion and intersystem communication (IRC and ISC) links so that they can be monitored by
SA z/OS CICS Automation. For more information on how to do this, see "More about Policy Item CICS CONNECTION." Also refer to System Automation for z/OS CICS Automation Programmer’s Reference and Operator’s Guide section CONNECTION.

MONITORING PERIOD
With this policy item you can select one monitoring period which has previously been defined with option (CVP) Link Monitor periods for the current CICS link definition (CICS connection).

For more information on link definitions, refer to System Automation for z/OS CICS Automation Programmer’s Reference and Operator’s Guide

More about Policy Item CICS CONNECTION: If you select policy item CICS CONNECTION from the Policy Selection panel for CICS links, SA z/OS displays the following panel where you describe the inter-region and inter-system communication (IRC and ISC) links to be monitored by SA z/OS CICS Automation.

<table>
<thead>
<tr>
<th>COMMANDS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADFPCCN</td>
<td>CICS connection links</td>
</tr>
</tbody>
</table>

Command ==> 

Entry Type : CICS link  PolicyDB Name : DATABASE NAME
Entry Name : LINK_DEF1  Enterprise Name : YOUR_ENTERPRISE

Enter or update the following fields:

- Connection id . . . . . . CICS sysid
- Type . . . . . . . . . . (IMS/CICS)
- Echo . . . . . . . . . . . (txn/pgm,seconds)
- Time zone offset . . . . 0 (hh:mm,WEST/EAST)
- Check interval . . . . . 05:00 (mm:ss)
- Max repair tries . . . . 3 (0-99)
- Release delay . . . . . . 05 (4-99)
- Acquire delay . . . . . . 05 (4-99)
- Critical link (SDF) . . NO YES NO

Figure 183. CICS Connection Links Panel

Fill in the following fields:

**Connection id**
The 4 character symbolic name by which this link is known to this CICS subsystem. This is defined on the CICS subsystem itself (SYSID).

**Type**
Indicates the type of subsystem this link is attached to, valid values are "CICS" or "IMS", this is used only when you want echoplexing to occur.

**Echo**
depends on the entry in the **Type** field:

- **CICS** The transaction name used to call the echoplex program on the target subsystem.
- **IMS** Any non-blank character.

The time parameter is used to define the time delay in seconds after which an echo response must have been received from the remote system before a message is issued. Valid values are from 1 to 99, default 3.
Product Automation Policy Object

Time zone offset
This parameter specifies the remote systems time offset relative to the UTC (Greenwich mean time). This is applied to the times specified in the DAY keyword to determine when monitoring is active for the remote system. Default value is 0.

Check interval
The time interval between 2 consecutive checks of a connection. Valid values are from 00:05 to 99:59, the default value is 05:00.

Max repair tries
Specifies the number of automatic actions that are to take place after the detection of a connection failure. Valid values are 0 to 9, default value is 3.

Release delay
Specifies the time delay in seconds needed by the system to release a connection. Valid values are 4 to 99, default value is 5.

Acquire delay
Specifies the time delay in seconds needed by the system to acquire a connection. Valid values are 4 to 99, default value is 5.

Critical link
Specifies whether or not a link failure for this link will change the severity shown on the SDF panels, default value is NO.

Link Monitor Periods
Figure 184 shows the policy items that are displayed when you select a monitoring period CICS from the related Entry Name Selection panel or if you are creating a new monitoring period.

<table>
<thead>
<tr>
<th>ACTIONS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFGEPOL</td>
<td>Policy Selection</td>
</tr>
<tr>
<td>Command</td>
<td>Row 1 to 5 of 5</td>
</tr>
<tr>
<td></td>
<td>SCROLL==&gt;&gt; PAGE</td>
</tr>
</tbody>
</table>

Entry Type: Monitoring period
Entry Name: MON_PERIOD_1
PolicyDB Name: DATABASE_NAME
Enterprise Name: YOUR_ENTERPRISE

Action
- Policy Name
- DESCRIPTION
- MONITORING PERIOD
- WHERE USED
- COPY

Policy Description
Enter description
Define the monitoring period
List APLs using this entry
Copy from existing entry

Figure 184. Policy Selection Panel for CICS Link Monitor Periods

The unique policy item for a CICS monitoring period is:

MONITORING PERIOD
With this policy item you can specify time periods when the current CICS connection should be monitored. You specify these time periods as described in “More about Policy Item MONITORING PERIOD.”

More about Policy Item MONITORING PERIOD: If you select policy item MONITORING PERIOD from the Policy Selection panel for CICS monitoring periods, SA z/OS brings up the following panel where you define the time frames for when the CICS connection link related to the (CCN) Link definition should be active.
For information about valid values and the syntax of the time frames, refer to the online help.

### Defining Automation for IMS Components

For IMS components, the following entry types are available:

**(ISA) State/Action tables**

With this option, you can define or modify a state/action table for IMS. This is described in more detail in [“IMS State/Action Tables.”](#)

**(ISF) IMS XRF status files**

An IMS XRF status file policy object is used to define the partner domain and status file in an XRF environment. These objects are linked to `System` objects. (XRF stands for extended recovery facilities).

**(IRN) IMS XRF Resources**

With this option, you can match the XRF RENAME to the IMS subsystems with which it is associated. This is described in more detail in [“IMS XRF Resources” on page 250.](#)

These are described in more detail in the subsequent sections.

### IMS State/Action Tables

*Figure 186 on page [248](#) shows the policy items that are displayed when you want to create a new state/action table for IMS or if you select an existing one from the related `Entry Name Selection` panel.*
The unique policy item for IMS state/action table is:

IMS STATE/ACTION
This policy item lets you define IMS state/action tables. How to do this is described in more detail in "More about Policy Item IMS STATE/ACTION."

More about Policy Item IMS STATE/ACTION: The IMS state action tables panel is displayed if you select policy item IMS STATE/ACTION from the Policy Selection panel for IMS State/Action shown in Figure 186. Use this panel to define the state/action tables that are available to your applications. For detailed information on state/action tables please refer to System Automation for z/OS IMS Automation Programmer's Reference and Operator's Guide.

To define a state/action table, fill in the following fields:

MSC link recovery
The state/action table used for automatic recovery from MSC link errors.

OLDS error recovery
The state/action table used to define procedures for automatic recovery from OLDS errors.

MPP/BMP recovery
The state/action table used to define automatic recovery procedures initiated when an MPP or BMP transaction/program has abended.
IMS XRF Status Files

Figure 188 shows the policy items that are displayed when you select an IMS XRF status file from the related Entry Name Selection panel or when you want to create a new IMS XRF status file.

The unique policy item for IMS XRF status files is:

**IMS STATUS FILE**

This policy item lets you define IMS status file names. How to do this is described in more detail in “More about Policy Item IMS STATUS FILE.”

**More about Policy Item IMS STATUS FILE:** The IMS Status file specification panel is displayed if you select policy item IMS STATUS FILE from the Policy Selection panel for IMS XRF Status File shown in Figure 188.

The status file specification describes the partner IMS status file that will be used in an XRF IMS environment. To support an XRF IMS environment, IMS Automation must access the status file of the partner IMS subsystems. This is only required in an XRF environment.

**Partner domain name**

This parameter is the NetView domain ID that is controlling the partner IMS subsystem in an XRF environment.
Partner status file

This parameter is the data set name of the status file used by the partner IMS in the partner’s domain. This data set is dynamically allocated during SA z/OS initialization.

For more information about the status file specification refer to the description of the STSFILE keyword in System Automation for z/OS IMS Automation Programmer’s Reference and Operator’s Guide.

IMS XRF Resources

Figure 190 shows the policy items that are displayed when you select an IMS XRF resource from the related Entry Name Selection panel.

<table>
<thead>
<tr>
<th>ACTIONS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>AOGEPOL</td>
<td>Policy Selection</td>
</tr>
<tr>
<td>Entry Type: IMS resource name</td>
<td>PolicyDB Name: DATABASE_NAME</td>
</tr>
<tr>
<td>Entry Name: IMS_XRF_1</td>
<td>Enterprise Name: YOUR_ENTERPRISE</td>
</tr>
<tr>
<td>Action</td>
<td>Policy Name</td>
</tr>
<tr>
<td></td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td></td>
<td>IMS RESOURCE NAMES</td>
</tr>
<tr>
<td></td>
<td>WHERE USED</td>
</tr>
<tr>
<td>COPY</td>
<td>Copy data from an existing entry</td>
</tr>
<tr>
<td>*******************************************************</td>
<td>*******************************************************</td>
</tr>
<tr>
<td>Bottom of data</td>
<td>Bottom of data</td>
</tr>
</tbody>
</table>

Figure 190. Policy Selection Panel for IMS XRF Resource

The unique policy item for IMS XRF resources is:

**IMS RESOURCE NAMES**

This policy item is used to match the XRF RSENAME to the IMS subsystems with which it is associated. These entries are linked to SYS entries. For more information, refer to "More about Policy Item IMS RESOURCE NAMES."

More about Policy Item IMS RESOURCE NAMES: The IMS Resource names panel is displayed if you select policy item IMS RESOURCE NAMES from the Policy Selection panel for IMS Resource names shown in Figure 190. Use this panel to define IMS resource names. Such an IMS resource name should be coded for each XRF IMS complex defined, to match the XRF RSENAME to the IMS subsystems with which it is associated. For more information on IMS resource names refer to System Automation for z/OS IMS Automation Programmer’s Reference and Operator’s Guide for the description of the IMSRESNM keyword.
To define IMS resource names, fill in the following fields:

**XRF Resource ID**
This is the value specified as the RSENAME for the XRF complex as defined in the DFSHSBxx member in the IMS PROCLIB.

**XRF partner name1**
This parameter is the SA z/OS subsystem name of one of the IMS control region subsystems that participate in an XRF complex.

**XRF partner name2**
This parameter is the partner IMS control region subsystem name that participates in this XRF complex.

The order in which the IMS subsystems are specified does not matter.

---

**Defining Automation for OPC Components**

For OPC components, the following entry types are available:

**(OEN) OPC System details**
With this option, you can define certain system wide defaults. Selecting this entry type will bring you to the *Entry Name Selection* panel for OPC system details where you can create or edit system defaults. You can link an OPC system detail to one or more System policy objects, however, you can link only one OPC system detail to a System.

**(OCS) Controller details**
With this option, you can link the OPC controller subsystem name with the NetView domain upon which it resides. Selecting this entry type will bring you to the *Entry Name Selection* panel for OPC controller details where you can create or edit controller details. You can link an OPC controller detail to one or more System policy objects, however, you can link only one OPC controller detail to a System.

**(OSR) Special resources**
Use this option to define special resources that are to be set for an OPC controller.

**(ODM) Workstation domains**
Use this option to relate an OPC automatic workstation to a NetView domain ID.

These are described in more detail in the subsequent sections.
Product Automation Policy Object

OPC System Details

Figure 192 shows the policy items that are displayed when you select an OPC system detail from the related Entry Name Selection panel or if you want to create a new OPC system detail.

<table>
<thead>
<tr>
<th>ACTIONS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFGEPOL</td>
<td>Policy Selection</td>
</tr>
<tr>
<td>Command ===</td>
<td>SCROLL===&gt; PAGE</td>
</tr>
</tbody>
</table>

- Entry Type: OPC System details
- PolicyDB Name: DATABASE_NAME
- Entry Name: OPC_SYS_DET1
- Enterprise Name: YOUR_ENTERPRISE

<table>
<thead>
<tr>
<th>Action</th>
<th>Policy Name</th>
<th>Policy Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESCRIPTION</td>
<td>Enter description</td>
<td></td>
</tr>
<tr>
<td>OPCAO ENVIRON</td>
<td>Define OPCAO environ entries</td>
<td></td>
</tr>
<tr>
<td>WHERE USED</td>
<td>List Systems linked to this entry</td>
<td></td>
</tr>
<tr>
<td>COPY</td>
<td>Copy data from an existing entry</td>
<td></td>
</tr>
</tbody>
</table>

---

Figure 192. Policy Selection Panel for OPC System Detail

The unique policy item for OPC system details is:

OPCAO ENVIRON

This policy item checks the status on START and STOP requests, and specifies the retention of critical messages. If you select this policy item, SA z/OS displays the OPC System details panel which is described in more detail in "More about Policy Item OPCAO ENVIRON."

More about Policy Item OPCAO ENVIRON: The OPCAO ENVIRON policy item checks status on START and STOP requests, and specifies retention of critical messages. If you select this policy item from the Policy Selection panel for OPC system details, SA z/OS displays the panel shown in Figure 193.

<table>
<thead>
<tr>
<th>COMMANDS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFFDEN</td>
<td>OPC System details</td>
</tr>
<tr>
<td>Command ===</td>
<td></td>
</tr>
</tbody>
</table>

- Entry Type: OPC System details
- PolicyDB Name: DATABASE_NAME
- Entry Name: OPC_SYS_DET1
- Enterprise Name: YOUR_ENTERPRISE

OPC system details entry name: OPC_SYS_DET1

Enter or update the following table names:

- Status check on requests... YES  (YES/NO)
- Critical msg retention... PERM  (hh:mm/PERM)
- Operation reset delay... NEVER  (hh:mm/NEVER)

---

Figure 193. OPC System Details Panel

The following fields can be filled in:

Status check

This entry controls checking of subsystem status on OPC Automation START, STOP, and RECYCLE requests. If not coded, OPC Automation will enforce the rule that START operations will be ended in error if the subsystem is already UP, and likewise for STOP operations if the subsystem is down. Code REQUEST=NO if you wish to have OPC Automation proceed as normal in this case. The default value is YES.
Critical msg retention

This entry specifies retention of OPC Automation critical messages automated with the DFCRIT command, and OPC error messages. The default is PERM, which means that messages will not be deleted on a timed basis.

Operation reset delay

OPRESET specifies how long the NetView interface to OPC may be unavailable before OPC Automation will not reset operations which ended in error while it was down. The NetView interface consists of both the NetView SA z/OS OPC Automation address space and its associated NetView SSI address space, over which requests flow from OPC to OPC Automation. When the interface is down, OPC Automation sets any operations destined for a NetView workstation to error status with an error code of UNTV. If no value is coded, or OPRESET=NEVER is coded, then no operations will be reset when the interface becomes available again. The default value is NEVER.

For more information refer to [System Automation for z/OS OPC Automation Programmer’s Reference and Operator’s Guide](#).

Controller Details

Figure 194 shows the policy items that are displayed when you are creating a new OPC controller detail or when you select an existing OPC controller detail from the related Entry Name Selection panel.

![Policy Selection Panel for OPC Controller Details](image)

The unique policy items for OPC controller details are:

**OPCA PCS**

With this policy item you specify either the NetView domain on which the OPC controller resides or that the local sysplex is to be searched for the active controller when required. It must also specify the subsystem name for the OPC controller. For more information, refer to “More about Policy Item OPCA PCS” on page 254.

**SPECIAL RESOURCES**

Use this policy item to select one or more special resources as defined with option (OSR) Special resources for the current controller. These selected special resources for OPC are used on the systems to which the current controller details entry is linked. (You can view the systems to which this entry is linked and link or unlink this entry to systems using the WHERE USED policy item).
More about Policy Item OPCA PCS: The panel shown in Figure 195 appears if you select policy item OPCA PCS from the Policy Selection panel for OPC controller details. The OPCA PCS automation definition defines the NetView domain on which the OPC Controller resides and specifies the subsystem name for the OPC Controller system.

Figure 195. OPC Controller Details Panel

Fill in the following fields:

NetView domain name
Specify the parameter as either a NetView domain ID (up to five characters) or the keyword SYSPLEX. If a domain ID is specified, it indicates that the OPC Controller always runs on the specified domain. The keyword SYSPLEX indicates that the OPC Controller may be run on any one of the systems in the local sysplex.

OPC controller subsystem
The MVS subsystem name of the OPC Controller as defined in the IEFSSNxx member.

Note: When defining the OPC Controller to SA z/OS, this name must be used as the subsystem name and job name.

OPC Special resources
The following entries are valid:

YES The OPC special resources will be set for any SA z/OS monitored subsystem specified in the OPC special resource definitions panel (see Figure 197 on page 255). Ensure that the OPC Special Resources are defined if using this option.

Note: If no OPC Special Resources were defined, and the OPC Special Resource SRSTAT=YES is defined in the OPC Controller details panel, then, when a BUILD is done for the ACF, the OPC Special resource definition will be reset to SRSTAT=NO as it would have detected that there were no OPC Special Resources defined in the ACF.

NO No Special Resources will be set. This is the default.
ALL Special Resources will be set for all SA z/OS Monitored subsystems. No Special Resource entries are required.

**OPC Special Resources**

Figure 196 shows the policy items that are displayed when you select an OPC special resource from the related *Entry Name Selection* panel.

![Figure 196. Policy Selection Panel for OPC Special Resources](image)

The unique policy item for an OPC special resource is:

**SPECIAL RESOURCE**

With this policy item you can specify resources as special resources for OPC. SA z/OS will display the *OPC special resource definitions* panel. For a detailed description of this panel, refer to “More about Policy Item SPECIAL RESOURCE.”

**More about Policy Item SPECIAL RESOURCE:** With this policy item SA z/OS is able to globally control the creation and setting of OPC Special resources based on the status of SA z/OS monitored subsystems. This allows OPC to resolve job scheduling dependencies based on the status of SA z/OS automated resources. A group or groups of special resources can be linked to a controller definition defined with option 31 (see “Defining Automation for OPC Components” on page 251).

![Figure 197. OPC Special Resource Definitions Panel](image)

Fill in your OPC Special resource names with up to 24 characters. Wildcard characters and patterns may be entered.
Workstation Domains

Figure 198 shows the policy items that are displayed when you select an OPC workstation domain from the related Entry Name Selection panel.

The unique policy item for an OPC workstation domain is:

OPCA DOMAINID

If you select this policy item, the Message Processing panel will come up where you specify how SA z/OS should react if OPC issues the specified message. For a detailed description of this panel and how to define message processing policy on it, refer to Figure 113 on page 166. The Message Processing panel for OPC shows minor differences in the top of the panel compared to the one shown in Figure 113 on page 166. These are shown here:

<table>
<thead>
<tr>
<th>Entry Type</th>
<th>Policy Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workstation domainID</td>
<td>PolicyDB Name</td>
<td>DATABASE_NAME</td>
</tr>
<tr>
<td>Entry Name</td>
<td>OPC WST DOM1</td>
<td>Enterprise Name</td>
</tr>
<tr>
<td>Action</td>
<td>Policy Name</td>
<td>Description</td>
</tr>
<tr>
<td>OPCA DOMAINID</td>
<td>Define the OPCA Domainid entries</td>
<td></td>
</tr>
<tr>
<td>WHERE USED</td>
<td>List Systems linked to this entry</td>
<td></td>
</tr>
<tr>
<td>COPY</td>
<td>Copy data from an existing entry</td>
<td></td>
</tr>
</tbody>
</table>

For more information refer to System Automation for z/OS OPC Automation Programmer’s Reference and Operator’s Guide under OPCA DOMAINID. Note that only a single entry of DOMAINID may be entered in the Message ID field; no other value will be accepted. The actual Workstation/Domain entries are made by entering CODE in the Action field.

Timers Policy Object

The Timers policy objects allow you to define sets of timers that run commands or automation procedures at scheduled times. Figure 199 on page 257 shows the policy items that are displayed when a new Timers entry is defined or when an existing Timers entry is selected from the Entry Name Selection panel.
The unique policy item for Timers is:

**TIMERS**
You define a timer in the *Timer specification* panel shown in [Figure 200](#).

SA z/OS displays the following panel if you select the TIMERS policy item from the *Policy Selection* panel for Timers.

---

**Figure 199. Policy Selection Panel for Timers**

You should note that, at run time, the commands that you define here are converted to the NetView CHRON format.

A timer definition consists of:

**Timer ID**
When a user timer is defined with the SAVE-option outside the control of SA z/OS, it is recommended that a unique timer ID is specified to avoid duplicate timers being created at automation control file (ACF) reload.

**Execution Time**
When SA z/OS uses the timer. You can schedule a timer for a specific day and time, after a certain day of the year, or repetitively at intervals.

**Issuer** Where the timed command or application program should run. You can specify that the command or application program should run under:
Timers Policy Object

- A NetView operator ID, in which case SA z/OS issues the timer-driven command only if that operator is logged on.
- The primary programmed operator interface task (PPT), in which case SA z/OS issues the timer-driven command even when an operator is not logged on.
- An automation operator name.

Save

Whether SA z/OS should save the timed event in the NetView SAVE/RESTORE database. You can enter one of the following values:
- YES
- NO

Clock

Specifies whether the command is to run if it were on a Greenwich Mean Time schedule or the local system time. You can enter one of the following values:
- GMT
- LOCAL

Catchup

Whether a saved timer should be caught up after a system outage.

Command text

The actual timer-driven command or automation procedure SA z/OS issues or calls.

Timeout Settings Policy Object

This policy object enables you to create sets of timeout definitions. Each timeout definition can specify timeout values for:
- System, NetView, or SA z/OS commands
- Cross-domain logon requests
- Your own timeout variables

If the time limit is exceeded, then it is assumed that an error condition exists and an appropriate action is taken.

Although only two specific fields, Wait and XDOM, are given, you can add other timeout names that may be unique to your environment. Figure 201 shows the policy items that are displayed when a new Timeout Setting entry is defined or when an existing Timeout Setting entry is selected from the Entry Name Selection panel.

Figure 201. Policy Selection Panel for Timeout Settings
The unique policy item for Timeout Settings is:

**TIMEOUT VALUES**

SA z/OS defines two timeout variables, XDOMTIME and WAITTIME. Values for these variables can be set in the Wait and XDOM Value fields of the Environment Definition: TIMEOUT panel. Figure 202 shows the definition panel for a Timeout Settings.

![Figure 202. Environment Definition: TIMEOUT Panel](image)

The following timeout values can be entered:

**Wait** Defines how long automation processing waits for a response after issuing a system, NetView, or SA z/OS system operations command. The wait parameter is used in the automation code to determine when a response error condition has occurred. Depending on which part of the code was being used at the time of the error and how critical the command is, an action is taken. The default time is 29 seconds. The range of values is 1 to 32000 seconds.

It is recommended that you choose a value less than 30 seconds to avoid the NetView message DSI594A being displayed in SA z/OS full screen function. The field, wait, creates the global variable WAITTIME.

**XDOM** Defines the maximum amount of time that message forwarding routines wait for a response from a cross-domain logon command. If this value is exceeded, the routines assume that the logon attempt failed. The default time is 120 seconds. The range of values is 1 to 32000 seconds. XDOM creates the global variable XDOMTIME.

**Installation supplied timeout names and values**

This section is used to define your own timeout values. Common global variables are created containing these values, and these variables can be used in your own automation procedures. The value of these variables must be in the range of 1 to 32000 seconds.
**Timeout Settings Policy Object**

---

**Note:**

You must ensure that the names of any global variables you create do not clash with SA z/OS external or internal global variable names. SA z/OS external global variables are documented in [System Automation for z/OS Customizing and Programming](#). In addition, you should not use names beginning with EH, AOF, SUBS, ING or ISQ.

For example, you can define the name TEST with a value of 30. This creates a global variable called TESTTIME containing the value 30. The variable TESTTIME can now be used in any automation procedure. You can enter an optional description of the timeout variable for reference.

---

**Note:**

The TESTTIME variable is initialized after SA z/OS has initialized its own environment from the automation control file. This means you should not use a variable such as TESTTIME in a routine which is invoked before SA z/OS is fully initialized.

---

**Tape Attendance Policy Object**

This policy object allows you to specify whether or not you want to monitor tape mounts on this system. If you do want to monitor tape mounts, you must enter warning and alert thresholds (in minutes and seconds) indicating how long a wait is allowed in mounting a tape before the response is considered unsatisfactory.

You can customize SA z/OS to respond to the IEF238D message while tape drives are attended by operators. This can be achieved by specifying a coded entry for the IEF238D message in the MESSAGES/USER DATA policy item of the MVS Component policy object. The coded entries should be entered as follows:

- **CODE1** should contain a pattern match for the job name requiring the response to the IEF238D message. You can specify a trailing asterisk (*) to allow for generic job name matching.
- **CODE2** should contain either D, W, C, DW, DC, WC or DWC where D is Device name, W is Wait and C is Cancel. The most likely combination to use would be DC (Device name or Cancel) and WC (Wait or Cancel).
- **CODE3** is not required for this message and should be left blank.
- **Value Returned** should contain your reply to IEF238D dependent on the results of the CODE1 and CODE2 matching process. Typically this will be NOREPLY, CANCEL or WAIT.

SA z/OS uses the CDEMATCH common routine using the IEF238D CODE1 and CODE2 values to issue the returned value as the reply. This is achieved via an entry in the AT for the IEF238D as shipped with the samples. For further information, with regard to [Responding with Code Matching](#), refer to chapter Adding a Message to Automation in [System Automation for z/OS Customizing and Programming](#). For more details on the format of the IEF238D, refer to the system message documentation for your particular operating system level and release.

This policy object also allows you to set a schedule for unattended tape drives. If a job requests a tape mount at a time defined as unattended, then the job will be canceled. [Figure 203 on page 261](#) shows the policy items that are displayed when a
new Tape Attendance entry is defined or when an existing Tape Attendance entry is selected from the Entry Name Selection panel.

The unique policy item for Tape Attendance is:

ATTEND TIME

This policy item allows you to specify when tapes are attended and whether or not you want to monitor tape mounts on this system. If you do want to monitor tape mounts, you must enter warning and alert thresholds (in minutes and seconds) indicating the maximum tape mount delay that is allowed before the response is considered unsatisfactory. Refer to Figure 204 for more information on this policy item.

Figure 204 shows the Environment Definition: TAPES panel.

The fields are:

Tapes Attended

Specify whether tape drives are attended or not. This is a global setting that indicates if there are any attended tapes attached to the system. If there are attended tapes then you can specify days and times at which they will not be attended.
Tape Attendance Policy Object

Monitor Tape Mounts
Specify whether tape mounts should be monitored on the SA z/OS workstation. If you enter YES in this field, SA z/OS will automatically create a Tape Mounts object for the NMC workstation. A Tape Mounts object is an icon that represents an unsatisfied tape mount request by a job. When SA z/OS detects a tape mount request, it creates an icon under the Tape_Mount group icon. The icon is green, initially, indicating a satisfactory condition. If the tape mount request is not satisfied before the Warning interval has expired, the icon changes color to white. If the tape mount request is still not satisfied after the Alert interval has expired, the icon changes color from white to red.

The icon is deleted automatically when the requested tape is mounted or the job requesting the tape mount is canceled.

Note:
The operator can specify whether the NMC workstation view that contains the tape mount icons is automatically refreshed. It is advisable to set this to refresh automatically so that tape mounts can be monitored without manual intervention.

The name of the tape mount object is of the form
SSSSS.VVVVVV_DDD(D)

where:
SSSSS is the NetView domain name where the mount request was issued. This may be a remote system.
VVVVVV is the tape volume name
DDD(D) is the device number that the tape should be mounted on. This device number is either 3 or 4 digits depending on the level of z/OS.

Warning Delay
The delay period, in minutes and seconds, after which the tape mount icon changes to the warning color of white if the tape mount has not been completed.

Alert Delay
The delay period, in minutes and seconds, between the time that the warning delay is reached and the time that the tape mount icon changes to the alert color of red, if the tape mount has not been completed.

If tapes are attended, enter days and times (24 hour clock) when unattended.
These fields are used to enter days and times when tape drives are unattended. A "*" character can be used in the Day field to specify all days of the week.

MVS Component Policy Object

This policy object allows you to apply automation to various z/OS data sets, resources and facilities. These z/OS data sets, resources, and facilities are referred to as MVS components.
You can set error thresholds and automation flags for the following MVS components:

- MVSDUMP (SYS1.DUMP00 - xx)
- SMFDUMP (SYS1.MANx)
- LOGREC (SYS1.LOGREC)
- SYSLOG (SYS1.SYSLOG.INDX)
- WTOBUF (z/OS Console Buffers)

In addition, you can customize the recovery procedure used for a WTOBUF overflow and automate responses to messages issued for MVS components. Figure 205 shows the policy items that are displayed when a new MVS Component entry is selected or when an existing MVS Component entry is selected from the Entry Name Selection panel.

**Figure 205. Policy Selection Panel for MVS Components**

The policy items for MVS Component policy objects are:

**MVSDUMP THRESHOLDS**
This option allows you to specify error thresholds for dump data sets. For more information see "More about the THRESHOLD Policy Items" on page 264.

**SMFDUMP THRESHOLDS**
This option allows you to specify error thresholds for System Management Facility (SMF) data sets. For more information see "More about the THRESHOLD Policy Items" on page 264.

**LOGREC THRESHOLDS**
This option allows you to specify error thresholds for LOGREC data sets. For more information see "More about the THRESHOLD Policy Items" on page 264.

**SYSLOG THRESHOLDS**
This option allows you to specify error thresholds for SYSLOG data sets. For more information see "More about the THRESHOLD Policy Items" on page 264.

**MESSAGES/USER DATA**
This policy item allows you to specify automated actions, such as the issuing of a command or reply, to be taken in response to a message issued
MVS Component Policy Object

by the application. You may specify commands if the application issues messages because something went wrong with the MVSDUMP, SMFDUMP, SYSLOG or LOGREC data sets. For more information see “More about Policy Item MESSAGES/USER DATA” on page 268.

MINOR RESOURCE FLAGS
This policy item allows you to define flags for minor resources for the system. Minor resources are any resources, such as messages or statuses, that are specified to be subsidiary to the major resource MVSESA.

Minor resources take, by default, the automation settings of their major resources. The main purposes in defining minor resources are to:

• Enable automation for minor resources
• Override the automation settings of major resources for a single minor resource, or a group of minor resources.

For information on this policy item see “More About Policy Item MINOR RESOURCE FLAGS” on page 269. For information on the relation of minor resources and automation flags refer to “Automation Flags and Minor Resource Flags” on page 39.

WTOBUF RECOVERY
This option allows you to customize recovery from write-to-operator (WTO) buffer shortage and full conditions at an operator console.

Note:
When SA z/OS receives a 100%-full WTOBUF message, it initiates the recovery process unless an operator has already performed the recovery. It is most likely that operators will handle 100%-full WTOBUF situations.

You can specify the number of backlogged console buffers allowed, and the time between WTOBUF recovery attempts. You can define a series of passes specifying commands to be issued in order to recover the WTOBUF. You can also specify parameters and additional commands to be issued after the buffer is cleared in order to reset the console to normal operation.

Refer to Figure 206 on page 265 for more information on this policy item.

More about the THRESHOLD Policy Items
For various threshold policy items, SA z/OS displays the appropriate panels as follows:

• for MVSDUMP THRESHOLDS: the Component Definition: MVSDUMP panel
• for SMFDUMP THRESHOLDS: the Component Definition: SMF DUMP panel
• for LOGREC THRESHOLDS: the MVS Component Definition: LOGREC panel
• for SYSLOG THRESHOLDS: the Component Definition: SYSLOG panel

The information provided in the THRESHOLDS dialogs is used to determine when an operator should be notified of an error situation. An attempt to recover an error (such as an abend of a subsystem) is handled by automation, but an operator may need to be informed if this error is occurring too often.

For more information about thresholds for MVS Components, refer to “Thresholds for MVS Components” on page 34.
More about the WTOBUF RECOVERY Policy Item

Note:

Use this policy only to define the WTO buffer shortage recovery process for SA OS/390 prior to version 2.1 (APAR level OW47903). For newer SA z/OS releases use the policy MESSAGES/USER DATA.

The following recovery parameters can be specified:

**NBUF**  This is the number of backlogged console buffers allowed. When this number is reached, recovery is initiated. The valid range of values is 0 to 9999. The default is 100.

**Retry**  This field represents the time between recovery attempts. The range is from 00:00:00 to 23:59:59. Each set of recovery actions is performed after waiting the time specified by the retry parameter until the WTOBUF relief message is received.

**RECOVERY**  This option specifies recovery operations. A panel similar to 

```
Figure 207 on page 266
```
is displayed if you select this option.
This panel defines the commands to be used for WTO buffer recovery and has the following options that can be filled in:

**Pass/Command Text**

This is a standard command specification and is used to define those commands to be issued when a WTO buffer shortage is detected. Recovering from the shortage may involve issuing several commands, so the PASS parameter can be used to define the command sequence.

A command to be issued on every pass should specify * as the pass number. The variables EHKVAR1 and EHKVAR2 are automatically filled with the console ID, or name if the ID is not specified, and the device number of the console in trouble. These variables can be used in any of the recovery commands, for example:

```mvs
MVS K S,RTME=001,DEL=RD,L=&EHKVAR1
```

If automation is on, the recovery process is as follows.

When the WTOBUF shortage or full message is received the MVS command 'D C,B' will be issued to determine which consoles have a buffer shortage. Recovery commands will be issued for those consoles with a buffer backlog greater than the number of buffers specified by the NBUF parameter. Automation decides which pass number to start the recovery process with based on the console roll mode as follows:

- If the roll mode = rd, the starting pass number = 2.
- If the roll mode = r and the roll time > 1, the starting pass number = 2.
- If the roll mode = r and the roll time <= 1, the starting pass number = 3.
- For all other combinations the starting pass number = 1.

After executing all the commands associated with a pass, the recovery process will be scheduled to issue the next pass after waiting the amount of time specified by the retry parameter. Automation will check to see if the WTOBUF relieved message has been received before each pass is issued.
NORMCONS

Choose this option to specify procedures for restoring normal console mode after a WTOBUF recovery.

A panel similar to Figure 208 is displayed.

When a WTOBUF shortage has been relieved by SA z/OS, consoles involved in the recovery will be restored to the normal operating mode. The normal console mode can be defined by completing the following options:

The normal console mode can be defined by completing the following options:

**Roll time**

This is the time (in seconds) between messages rolling on the console. The value can be 0, 1/4, 1/2, or 1 to 999.

**Roll mode**

The message deletion mode. The following modes can be specified:

- **AUTO** - Enter this to specify automatic message deletion mode. When the screen is full, all flagged messages are removed.
- **MAN** - Enter this to specify manual message deletion mode. All messages must be removed from the screen manually.
- **ROLL** - Roll mode takes effect if this is entered. At each specified interval (defined in Roll Time field) a specific number of messages (defined in RNUM parameter of CONSOLxx member in PARMLIB) are rolled off the screen.
- **ROLLD** - Roll-deletable mode takes effect if this is entered. The messages roll off in the same way as roll mode except that action messages are held at the top of the screen.
- **WRAP** - Wrap mode takes effect. When the screen is full, the next message overlays the message at the top of the screen, and subsequent messages continue overlaying older messages down the screen.

---

**Figure 208. Console Mode After WTOBUF Recovery Panel**

The following fields can be specified:

**Roll time**

Seconds between message rolls

**Roll mode**

Message deletion mode:
- Automatic (AUTO), Manual (MAN), Roll (ROLL), Roll all but action messages (ROLLD, roll-delete), Wrap (WRAP)

**Console size**

Minimum size of console used to run SA z/OS

**Console Areas:** A B C D E F G Total Area Allocated

Size

---

Additional Commands to Restore Normal Console Operation.
MVS Component Policy Object

Console Size
The total number of lines on the console that may be used for display areas.

Console Areas
The seven letters (A-G) can represent seven different display areas on the console. The number of lines entered under the first letter defines a display area at the bottom of the screen. The next letters define areas moving towards the top of the screen. The total number of lines for the display areas should not exceed the number entered in Console Size.

Additional Commands to Restore Normal Console Operation.
Additional commands to restore the console to normal operating mode. The following commands will be issued if no values are specified:
’MVS K E,D,L=&EHKVAR1
’MVS K A,NONE,L=&EHKVAR1

More about Policy Item MESSAGES/USER DATA
If you select policy item MESSAGES/USER DATA from the Policy Selection panel for MVS Components (Figure 205 on page 263), you come to the Message Processing panel shown in Figure 209.

<table>
<thead>
<tr>
<th>ACTIONS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>===&gt;</td>
</tr>
<tr>
<td>Row 1 to 4 of 23</td>
<td></td>
</tr>
<tr>
<td>Entry Type : MVS Component</td>
<td></td>
</tr>
<tr>
<td>PolicyDB Name : DATABASE_NAME</td>
<td></td>
</tr>
<tr>
<td>Entry Name : MVS_COMPONENTS</td>
<td></td>
</tr>
<tr>
<td>Enterprise Name : YOUR_ENTERPRISE</td>
<td></td>
</tr>
<tr>
<td>Define message IDs and their automation actions.</td>
<td></td>
</tr>
<tr>
<td>CMD = Command REP = Reply CODE = CODE USER = User Data</td>
<td></td>
</tr>
<tr>
<td>AUTO = AT Actions OVR = AT Override</td>
<td></td>
</tr>
<tr>
<td>Action</td>
<td>Message ID</td>
</tr>
<tr>
<td>Description</td>
<td>LOGREC</td>
</tr>
<tr>
<td>Specifications for LOGREC</td>
<td>1</td>
</tr>
<tr>
<td>SYSLOG</td>
<td></td>
</tr>
<tr>
<td>Specifications for SYSLOG</td>
<td></td>
</tr>
<tr>
<td>MVSDUMPTAKEN</td>
<td></td>
</tr>
<tr>
<td>Dump is taken</td>
<td></td>
</tr>
<tr>
<td>MVSDUMPRESET</td>
<td></td>
</tr>
<tr>
<td>Dump reset</td>
<td></td>
</tr>
</tbody>
</table>

Figure 209. Message Processing Panel

The MESSAGES/USER DATA policy item allows you to specify message IDs, and special ones like MVSDUMPTAKEN, MVSDUMPRESET, MVSDUMP, SMFDUMP, SYSLOG, LOGREC, and WTOBUF.

On this panel, you can define how SA z/OS should react to an issued message. Message processing provided here is the same as provided for Applications. For more information refer to “More about Policy Item MESSAGES/USER DATA” on page 165 and System Automation for z/OS Customizing and Programming.
More about Policy Item MINOR RESOURCE FLAGS

The Minor Resource Selection panel is displayed if you select policy item MINOR RESOURCE FLAGS on the Policy Selection panel for entry types Application or MVS Component.

The Minor Resource Selection panel allows you to manage your minor resource definitions. You can add, delete or rename minor resources. To add a minor resource, enter its name into the Minor Resource column. You can select a minor resource to edit the automation flags on the next panel Flag Automation Specification.

You can also define a hierarchy of minor resources. For example, for the minor resource TAPES in the example in Figure 210, you can define more detailed minor resources. The hierarchy of minor resources is denoted by dots. You can set automation flags for the complete hierarchy by setting them for the minor resource TAPES or you can set certain flags for selected lower level minor resources.

Example:

```
- TAPES
  - TAPES.SYS1
  - TAPES.SYS2
  - TAPES.SYS3
- MVSDUMP
- SCU0040
- SCU0050
- SCU0060
- SCU0070
- SCU0080
- SCU0090
- SMFDUMP
- SYSLOG
- TAPES
- WTOBUF
```

Figure 210. Minor Resource Selection Panel

The Minor Resource Selection panel allows you to set default values for MVS Component sets. Figure 211 on page 270 shows the policy items that are displayed when a new MVSCOMP Defaults entry is defined or when existing MVSCOMP Defaults entry is selected from the Entry Name Selection panel.

MVSCOMP Defaults Policy Object

MVSCOMP Defaults allow you to set default values for MVS Component sets.
The unique policy items for MVSCOMP Defaults are:

**AUTOMATION FLAGS**
This option allows you to specify automation settings that will be used as defaults for MVS Components. The settings specified here will be used if you do not provide specific automation settings for MVS Components. See "Automation Flags" on page 34 for more information about setting automation flags.

**THRESHOLDS**
This policy item is used to set default error thresholds for MVS Components. The settings specified here will be used if you do not provide specific thresholds for MVS components. See "How SA z/OS Uses Error Thresholds" on page 31 for more information about error thresholds.

More than one MVSCOMP Default may be attached to a system, but it is recommended that you attach only one to avoid data being overwritten during ACF LOAD.

**System Defaults Policy Object**

This policy object allows you to set various defaults that are used throughout the system. The values set here may be overridden by other objects, such as Application class, Application Defaults, or MVSCOMP Defaults, or by setting appropriate values in individual application or MVS Component objects. Figure 212 on page 271 shows the policy items that are displayed when a new System Defaults entry is defined or when an existing System Defaults entry is selected from the Entry Name Selection panel.
The unique policy items for System Defaults policy object are:

**AUTOMATION FLAGS**

This option allows you to specify automation settings that are used as defaults for the system. The settings specified here will be used if you do not provide specific automation settings for those policy objects which allow automation. See "Automation Flags" on page 34 for more information on setting automation flags.

**THRESHOLDS**

This policy item is used to set system-wide error thresholds. The settings specified here will be used if you do not provide object-specific threshold settings. See "How SA z/OS Uses Error Thresholds" on page 31 for more information on error thresholds.

**AUTOMATION OPTIONS**

This policy item is used to set the system-wide captured message limit. The value specified here will be used if you do not provide object-specific limits. See "More about Policy Item AUTOMATION OPTIONS" for more information.

**ENVIRONMENT**

This option allows you to set various defaults for applications you define to the system. The values entered in the Subsystem Defaults panel are used as default values for Application objects which are attached to this system and do not have this information specified.

More than one set of System Defaults may be attached to a system, but it is recommended that you attach only one to avoid data being overwritten during ACF LOAD.

**More about Policy Item AUTOMATION OPTIONS**

The System Automation Options panel, as shown in Figure 213 on page 272, is displayed if you select the AUTOMATION OPTIONS policy item on the Policy Selection panel for the System Defaults entry type.
System Defaults Policy Object

The following parameter can be specified for system-wide defaults. If the corresponding entry is not specified for individual resources, this value will prevail.

Captured Messages Limit
This specifies the maximum number of messages that will be captured for each resource while SA z/OS is running. This messages will show up in the DISPINFO or DISPMTR command output or SA z/OS Graphic Interface.

More about Policy Item ENVIRONMENT
The Subsystem Defaults panel, as shown in Figure 214, is displayed if you select the ENVIRONMENT policy item on the Policy Selection panel for the System Defaults entry type.

Application Defaults Policy Object
The Application Defaults policy object specifies default values for automation flags and error threshold settings. These values will be used as the default for any Application object on the system to which the Application Defaults set is attached. If the application belongs to an application class with automation flags or error thresholds specified, the class policy takes precedence over the defaults. Figure 215, on page 273, shows the policy items that are displayed when a new Application...
Defaults entry is defined or when an existing Application Defaults entry is selected from the Entry Name Selection panel.

<table>
<thead>
<tr>
<th>ACTIONS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0FGEPO1</td>
<td>Policy Selection</td>
</tr>
<tr>
<td>Command ===</td>
<td>SCROLL===</td>
</tr>
</tbody>
</table>

Entry Type : Application Defaults PolicyDB : DATABASE_NAME
Entry Name : APPL_DEFAULTS_NAME Enterprise Name : YOUR_ENTERPRISE

<table>
<thead>
<tr>
<th>Action</th>
<th>Policy Name</th>
<th>Policy Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESCRIPTION</td>
<td>Enter description</td>
<td></td>
</tr>
<tr>
<td>AUTOMATION FLAGS</td>
<td>Define Application automation flag defaults</td>
<td></td>
</tr>
<tr>
<td>THRESHOLDS</td>
<td>Define Application thresholds defaults</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>WHERE USED</th>
<th>COPY</th>
</tr>
</thead>
<tbody>
<tr>
<td>List Systems linked to this entry</td>
<td>Copy data from an existing entry</td>
</tr>
</tbody>
</table>

Figure 215. Policy Selection Panel for Application Defaults

The unique policy items for Application Defaults policy are:

**AUTOMATION FLAGS**
This option allows you to specify automation flag settings that will be used as defaults for all applications on the system. The settings specified here will be used if you do not provide application-specific automation settings. See “Automation Flags” on page 34 for more information on setting automation flags.

**THRESHOLDS**
This option allows you to specify threshold settings that are used as defaults for all applications on the system. Settings specified here will override system default settings and will be used if you do not provide application-specific threshold settings. See “How SA z/OS Uses Error Thresholds” on page 31 for more information on setting thresholds.

More than one set of Application Defaults may be attached to a system, but it is recommended that you attach only one to avoid data being overwritten during ACF LOAD.

**Auto Operators Policy Object**

*Automation operators* are automated operator tasks, that can respond to messages from the operating system, applications, and network without requiring a human operator. Sets of automation operators can be linked to a system. Each automation operator task takes actions similar to those of a human operator, such as monitoring the system and handling and coordinating recovery and shutdown operations.

Automation operators are assigned specific messages on which to act. During automated operations, the messages are routed to the appropriate operator using the NetView ASSIGN command. Figure 216 on page 274 shows the policy items that are displayed when a new Auto Operators policy object is created or Auto Operators is selected from the Entry Type Selection panel.
The unique policy item for Auto Operators is:

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 if you do not wish to use the sample definitions, 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 it uses. Certain automated function names must be defined. These are listed in Table 8. The automated function names are included in the sample policy databases 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 with which they are associated are defined in the sample NetView DSIOPF data set.

More about the OPERATORS Policy Item

Table 8 lists the automated function names that need to be defined.

<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 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>
</tbody>
</table>
Table 8. 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>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>General automation. 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.</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&lt;i&gt;</td>
<td>AUTCON</td>
<td>Used for the MVS and NetView console interface. The MVS console with which it is associated can be changed in the customization dialog.</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>
<tr>
<td>XCFOPER</td>
<td>AUTXCF</td>
<td>Used for enhanced Parallel Sysplex automation functions.</td>
</tr>
<tr>
<td>XCFOPER2</td>
<td>AUTXCF2</td>
<td>Used for enhanced Parallel Sysplex automation functions.</td>
</tr>
<tr>
<td>XCFOPER3</td>
<td>AUTXCF3</td>
<td>Used for enhanced Parallel Sysplex automation functions.</td>
</tr>
<tr>
<td>HWOPER1-33</td>
<td>AUTHW1-33</td>
<td>Used for Hardware Validation messages.</td>
</tr>
<tr>
<td>PLEXOPER,1,2</td>
<td>AUTPLEX, 1, 2</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.</td>
</tr>
</tbody>
</table>

It is highly recommended that you assign these required automated functions the suggested automated operators as listed in the above table. If you attempt to share automated operators across more than 1 automated function you will experience a significant performance degradation during the ACF load process.

Sample DSIOPF definition statements are included in the AOFOPF sample that is shipped with SA z/OS. You may define additional automation operators, to suit your automation needs.
Auto Operators Policy Object

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 DSIIPARM 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 System Automation for z/OS Customizing and Programming for more information on synonyms. Figure 217 shows an example of an Automation Operator Definition panel.

<table>
<thead>
<tr>
<th>COMMANDS</th>
<th>ACTIONS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFPIAOO</td>
<td>Automation Operator Definitions</td>
<td>Row 1 to 10 of 36</td>
</tr>
</tbody>
</table>

Command ==> SCROLL ==> PAGE

Entry Type : Auto Operators
Entry Name : BASE_AUTOOPS

PolicyDB Name : DATABASE_NAME
Enterprise Name : YOUR_ENTERPRISE

Actions: S = Select M = Move B = Before A = After I = Insert

Automated Action Function Messages for this Operator (* notation ok)
BASEOPER BASEOPER
GSSOPER GSSOPER
JESOPER JESOPER
LOGOPER *
MONOPER MONOPER
MSGOPER AOF*
MVSCONS
NETOPER
PPDOPER
RECORDER RECORDER ERB*,IEA*,IEE*,I0S*

Figure 217. Automation Operator Definition Panel

Messages are entered in the Messages for this Operator field. You can specify:

- A single message, such as IEE1151
- A group of messages, by specifying a message prefix and an asterisk (*), such as AOF* or $HASP*
- All messages, not otherwise assigned, by specifying an asterisk (*) only

How SA z/OS processes and assigns messages to automation operators depends on the order in which you specify automation operators on the Automation Operator Definition panel and the types of message you assign to each operator. The order in which operators are listed on the panel has meaning during operations. When a message occurs, SA z/OS searches this list of automation operators from the top to find which automation operator is associated with the message. To make sure SA z/OS assigns the right operator the right message, enter the more specific message ID assignments first in the automation operator list. Then, enter the more general message ID assignments later in the list. For example, if you want to make sure an automation operator receives all messages beginning with IEF4, you define that automation operator and message assignment in the list before an automation operator defined to receive all messages beginning with IE. The Automation Operator Definition panel has commands that help you change the order of automation operators in the list.

To continue specification of message IDs across multiple lines, insert a line where required and leave the automated function blank.

The following automated functions should not have messages assigned to them on the Automation Operator Definition panel:
For these automation operators, you should only specify the automated function name in the Automated Function field, and leave the Messages for this Operator field blank.

When you have entered the automated function and assigned messages a new panel will be displayed in which you can enter automation operators and NetView operators to receive messages. Figure 218 provides an example.

The fields are:

**Automation Operators**

Each automation operator relates to a specific NetView operator ID. Each message is assigned to a list of operators. This list includes the primary NetView ID, the secondary NetView ID (if specified), the primary NetView ID for SYSOPER, and the primary NetView ID for BASEOPER. The message is processed on the first operator in the list that is logged on when the message is received. If none of the operators are logged on, it will be processed on the task that started the system NetView Subsystem Interface Router task (SIR task), if it was started by a NetView task. In the NetView default initialization sequence, AUTO1 starts the SIR task. If the SIR task was not started by an autotask (that is, it has INIT=Y coded in DSIDMN) or the task that started it is no longer active, the message will not be automated. Similar lists of autotasks are defined in AOFMSGSY and are used to route commands to tasks from the NetView automation table.

If a group is specified (the ID starts with a “+”), then all the IDs that are assigned to the group are inserted into the list at that point.

You may also use SYSCLONE variables and the &DOMAIN. variable for the automation operators and NetView operators specification.
Auto Operators Policy Object

NetView Operators
Defines NetView IDs to receive copies of the messages assigned to the automation operator. The ID must be logged on to receive the message and either the primary or backup operator must also be logged on. Operator "groups" can be defined by specifying a "+" as the first character.

SA z/OS displays a different panel if MVSCONSi is entered on the Automation Operator Definition panel. Figure 219 provides an example.

![Figure 219. Automation Operator: MVSCONS Panel](image)

This panel allows you to specify an MVS console from which you can enter NetView commands. The MVS console, specified either by ID or by name, is associated with an automation operator.

The operator ID you enter here is used as the base for a series of sequentially numbered autotasks that are set up as servers for any real MVS consoles which SA z/OS detects attached to the machine. With the default ID of AUTCON and three consoles attached to the machine, autotasks AUTCON, AUTCON01 and AUTCON02 is used. These should be defined in DSIOPF. The allocation of a server autotask to a real console allows it to issue NetView and SA z/OS commands by prefixing them with the NetView Subsystem Interface identification character. This character is set by the DSIG parameter in the JCL procedure for the NetView SSI. It defaults to %.

Additional MVS consoles may be specified by entering MVSCONSi in the Automated Function column on the Automation Operator Definition panel (AOFPIAO0). The allowed values for the i suffix on MVSCONS are blank (MVSCONS), A-Z (MVSCONSA, MVSCONSB..., MVSCONSZ) and 0-9 (MVSCONS0, MVSCONSI..., MVSCONS9).

Structure of AOFWRKxxx Operators
AOFWRKxxx operators are used to distribute the work on different NetView tasks, so that they can run in parallel. AOFWRKxxx operators are used as dedicated automated function responsible for all the work on a particular subsystem. However, since 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 A0FWRKxx where xx is any number or character. The xx part of the automated function name does not have to be in sequence.

At ACF load time, each defined subsystem will be assigned an automated function from the pool of defined A0FWRKxx 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 A0FWRKxx automation operators. GSSOPER will also be used as backup when there is no A0FWRKxx automation operator defined.

---

**Notify Operators Policy Object**

The Notify Operators policy object allows you to define notification messages for certain operators. Select 38 Notify Operators on the Entry Selection panel to define these operators:

```
Command ===>
Policy Selection_Row 1 to 5 of 5_SCROLL===>
```

**ENTRY TYPE:** Notify Operators

**ENTRY NAME:** NOTIFY_OPERATORS

**ACTION Policy Name:** NOTIFY OPERATORS

**DESCRIPTION:** Define forward messages to operators

**WHERE USED:** List systems linked to this entry

**COPY:** Copy data from an existing entry

---

**Figure 220. Policy Selection Panel for Notify Operators**

**NOTIFY OPERATORS**

Notification operators are human NetView operators who receive SA z/OS notification messages, which are messages about events SA z/OS detects or actions it takes. These notification messages are routed to one or more
logged-on human operators, using the message forwarding path you set up in the Network policy object. The messages can optionally be held until the operator clears them.

<table>
<thead>
<tr>
<th>Commands</th>
<th>Actions</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFPIN00 Operator Notification</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Command ===>

Entry Type: Notify Operators
Entry Name: SAMPLE_NFYOP_01
PolicyDB Name: DATABASE_NAME
Enterprise Name: YOUR_ENTERPRISE

Enter operator information and select message type(s) to be held on screen.

Action: S = Select for message class assignment
Notify: Yes or No
Messages: S = Select messages to be held

<table>
<thead>
<tr>
<th>Operator ID</th>
<th>Operator Description</th>
<th>Notify</th>
<th>Messages</th>
<th>Information</th>
<th>Eventual Action</th>
<th>Immediate Decision</th>
<th>System Wait</th>
<th>Immediate Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>XXAOC05</td>
<td>SHIFT LEADER</td>
<td>YES</td>
<td>S S S S</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XXAOC06</td>
<td>SENIOR OPERATOR</td>
<td>NO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 221. Operator Notification Panel

On this panel you can select notification operators or define new notification operators that you want to be notified about selected message classes from applications on this system.

When you specify Yes for Assign User message classes, to have the operator notified about user-defined message classes, SA z/OS displays existing message classes that have been defined with policy item Auto Msg Classes of entry type ENT.

How to define messages for a message class and where to store them, is described in System Automation for z/OS Customizing and Programming.
When defining notification operators, you can customize:

- Which operators receive notification messages
- The types of message held on notification operators’ console screens:
  - Immediate Action
  - Immediate Decision
  - Eventual Action
  - Information
  - System Wait
- The classes of messages sent to the operators. (See "More about Policy Item AUTO MSG CLASSES" on page 87 for information about defining your own message classes.) Each SA z/OS message can have one or more message classes associated with it. Similarly, each notification operator can have one or more message classes assigned. When messages are processed during automated or manual operations, the class of each message is compared to the classes defined for each notification operator. If a match occurs, the notification operator receives the message. All one and two digit numeric message classes are reserved for IBM use. You may use message classes made up of one alphabetic and one numeric character (or one numeric and one alphabetic).

Table 9 shows the message classes defined by SA z/OS. Message classes from 80 to 90 are relevant for monitoring the automation manager. These message classes are discussed in more detail in System Automation for z/OS Messages and Codes.

Table 9. SA z/OS Message Classes

<table>
<thead>
<tr>
<th>Message Class Description</th>
<th>Message Class Number</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General Message Classes</strong></td>
<td></td>
</tr>
<tr>
<td>Errors</td>
<td>0</td>
</tr>
<tr>
<td>Debug</td>
<td>1</td>
</tr>
<tr>
<td><strong>Automation Agent Related Message Classes</strong></td>
<td></td>
</tr>
</tbody>
</table>
Table 9. SA z/OS Message Classes (continued)

<table>
<thead>
<tr>
<th>Message Class Description</th>
<th>Message Class Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>All automation agent messages</td>
<td>40</td>
</tr>
<tr>
<td>Information</td>
<td>43</td>
</tr>
<tr>
<td>Action</td>
<td>44</td>
</tr>
<tr>
<td>Status</td>
<td>45</td>
</tr>
<tr>
<td>User intervention</td>
<td>46</td>
</tr>
<tr>
<td>Critical threshold setting exceeded</td>
<td>50</td>
</tr>
<tr>
<td>Frequent threshold setting exceeded</td>
<td>51</td>
</tr>
<tr>
<td>Infrequent threshold exceeded</td>
<td>52</td>
</tr>
<tr>
<td>Assist mode alert</td>
<td>55</td>
</tr>
<tr>
<td>IMS Automation Messages</td>
<td>60</td>
</tr>
</tbody>
</table>

Automation Manager Related Message Classes

<table>
<thead>
<tr>
<th>Message Class Description</th>
<th>Message Class Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>All automation manager messages</td>
<td>80</td>
</tr>
<tr>
<td>Intervention required</td>
<td>81</td>
</tr>
<tr>
<td>Automation denied</td>
<td>82</td>
</tr>
<tr>
<td>Automation Impacted</td>
<td>83</td>
</tr>
<tr>
<td>Awaiting Automation</td>
<td>84</td>
</tr>
<tr>
<td>In Automation</td>
<td>85</td>
</tr>
<tr>
<td>Degraded - Target resources</td>
<td>86</td>
</tr>
<tr>
<td>Degraded - All resources</td>
<td>87</td>
</tr>
<tr>
<td>Satisfactory - Target resources</td>
<td>88</td>
</tr>
<tr>
<td>Satisfactory - All resources</td>
<td>89</td>
</tr>
<tr>
<td>SA z/OS Infrastructure/Environment</td>
<td>90</td>
</tr>
</tbody>
</table>

For example, any operator defined with a notification operator class of all SA z/OS messages (class number 40 and 80) receives all SA z/OS messages. Operators responsible for system operations should receive all SA z/OS messages or a subset of those messages. As another example, you may want to have your lead operator receive all debug messages.

SA z/OS uses the general and console automation message classes only. The SA z/OS customization dialog supports the message classes used by SA z/OS only.

SA z/OS uses notification messages to update the status display facility (SDF).

Network Policy Object

When multiple z/OS systems are interconnected and require consolidated operations at one focal point system, you can use focal point services to monitor and control the systems. These services allow an operator to view the status of multiple systems from a single system acting as a focal point and to send messages, commands, and responses from a focal point system to other systems. The Network policy object allows you to:

- Define paths for sending messages, commands, or responses from one system to another.
- Define primary and backup focal point systems.
Network Policy Object

- Define gateway sessions connecting systems.
  A gateway session uses autotasks both inbound to and outbound from a system
to send data between systems.
  Communication with the focal point system is now done using the NetView
  RMTCMD command or XCF. RMTCMD allows you to issue commands
cross-domain, and see the responses on your operator task.
- Manage passwords for gateway autotasks (for downlevel systems only).
- Define user Terminal Access Facility (TAF) full-screen sessions and specify
  whether or not each of these should be initiated automatically when the
  operator logs on to the SA z/OS NetView Interface. If a TAF session is not
defined to start automatically, you can initialize it using the OPER dialog.

Chapter 10, “Automating Networks,” on page 347 provides more information on
the definition of automation networks. Figure 223 shows the policy items that are
displayed when a new Network policy object is defined or when an existing
Network object is selected from the Entry Name Selection panel.

Figure 223. Policy Selection Panel for Network

The unique policy items for Network are:

**ADJACENT NETVIEW**

**Note:**
Use this policy only to define the adjacent NetView domains for
systems running SA OS/390 Version 2.2 or earlier. For newer
SA z/OS releases the forwarding of status updates via an
intermediate node to the focal point is no longer supported.

This option is used to define NetView domains that route commands,
replies, and notification messages between NetView domains that do not
have a direct connection between themselves. The adjacent domain should
have a direct connection to the To domain, and the current domain should
have a gateway to the Adjacent domain.

The fields are:
Network Policy Object

To Domain
The domain that is to receive the commands, replies, and notification messages from the current domain.

Adjacent Domain
The domain through which the commands, replies, and notification messages are forwarded to the To domain.

Alternate Adjacent
This domain is used if the adjacent domain is not active.

To-Domain Description
A one-line description of the To domain.

FORWARD
This option is used to define the domain which is to receive the notification messages to be forwarded from the current domain. Two domains can be specified in the following fields of the upcoming panel:

Primary Domain
In this field you enter the domain that is to receive the notification messages from the current domain. If this field is not filled in, the current domain will be the focal point (other domains forwarding messages to the current one without forwarding any itself). The name must consist of up to 5 alphanumeric characters or can be &DOMAIN.

Backup Domain
If the primary domain is not active, the domain defined here acts as the backup and receives its messages. If this field is not filled in, the backup defaults to the current domain. The name must consist of up to 5 alphanumeric characters or can be &DOMAIN..

FULL SESSIONS
With this option you can define applications to which TAF full screen sessions can be established. The following fields can be specified:

Session Name
This field acts as a description of the application that is displayed on the interface panel.

Application ID
The application name as specified in VTAMLST.

System
The name of the system that the application runs on. This field also acts as a description field that is displayed on the interface panel.

GATEWAY

<table>
<thead>
<tr>
<th>Note:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Because communication with the focal point is done by means of either the NetView RMTCMD command or XCF communication, the specification of password and logmode is no longer required.</td>
</tr>
</tbody>
</table>

Each domain that is to receive commands, replies, or notification messages must have a gateway defined for it. This includes the domain name and password for the outbound gateway autotask. The Gateway policy item consists of the following fields:
Network Policy Object

Domain
The names of the domains that should receive anything forwarded from the current domain.

Password
The password for the autotask to logon. The value RACFNNNT can be entered if RACF is to be used to control passwords.

Logmode
The logmode that SA z/OS uses when logging on to the specified domain for the outbound gateway session.

Description
A description of the domain you are communicating with.

SAF ENVIRONMENT

Note:
Use this policy only to define the System Authorization Facility (SAF) for systems running SA OS/390 version 2.2 or earlier. For newer SA z/OS releases the establishing of a gateway connection is only supported by means of the RMTCMD facility which does not require the specification of a password. Therefore SAF is not used to manage gateway access.

When the System Authorization Facility (SAF) is identified as being used to manage gateway access, SA z/OS maintains a VSAM database containing password and other information pertaining to the gateway autotasks. This option allows you to specify how this database is to be used and maintained. The following fields can be filled in:

Owner
If SAF is to be used to control the gateway autotasks, and a group of domains share the same SA z/OS password database, the owner of that database should be entered here. The owner is responsible for the maintenance of the passwords.

Group
This field is only used in situations where multiple shared RACF databases are in use.

Share
This field is used to list all the domains that share the SA z/OS password data set.

Mask
The Mask field is used to describe the format to be used when generating passwords for gateway autotasks. This mask is used only for SAF-controlled NNT sessions and is applied when a new password is required. The password mask specified here may be overridden on an individual session basis by the INGPW command specification.

The password generation mask consists of 4 to 8 pairs of characters, each pair defining the attributes of a single password character. The first character of each pair determines how the second character is to be used.

! - A password character of the value indicated by the following mask character is required.
% - A password character of the type indicated by the
following mask character is required.
? - A password character of the type indicated by the following mask character is optional.

The password mask is processed left to right and will generate a string of characters having a length minimally equal to the number indicated as being required. SA z/OS requires passwords consisting of at least 4 characters. The password character type codes indicate the set of characters to be used when generating a new password. The valid type codes are as follows:

$ - National only
A - Alphabetic only
B - Alphabetic/National
C - Consonant (Alphabetic - No Vowels)
E - Even Numerics
N - Numeric only
O - Odd Numerics
V - Vowels only
W - Alphanumeric - No Vowels
X - Alphanumeric
Y - Alphanumeric/National
Z - Alphanumeric/National - No Vowels

The following mask specification would cause SA z/OS to generate SAF passwords beginning with 'NNT' followed by a national character, a consonant, and a two or three digit even number.

Password Mask ==> !N!N!T%$%C?N%N%E

NNT Sessions Policy Object

Note:
Entry type NNT is no longer supported since SA z/OS Version 2.3. Use this function only to maintain systems with SA OS/390 releases up to 2.2 and for cleanup.

This policy object is used to define definitions for NetView to NetView sessions. Each NNT Sessions set can contain multiple defined NNT sessions. Figure 224 on page 287 shows the policy items that are displayed when a new NNT Sessions object is defined or when an existing NNT Sessions object is selected from the Entry Name Selection panel.
The unique policy item for NNT is:

**SESSIONS**

When you select this option, a panel similar to Figure 225 is displayed. You can list the user IDs you want to be used for NNT sessions. Each user ID must be valid on the domain to which the NNT session is being established. You can also enter a description for each ID.

![Policy Selection Panel for NNT Sessions](image)

**Figure 224. Policy Selection Panel for NNT Sessions**

To complete the definition for each ID, enter an S (Select) in the Action column. A panel similar to Figure 226 is displayed.

![NNT Sessions Definition Panel](image)

**Figure 225. NNT Sessions Definition Panel**

The fields are:
NNT Sessions Policy Object

Domain
This field names the NetView domain to which a session is routed, that is, the NetView domain in which the NNT is activate.

Password
The password of the ID in the To-Userid field can be hardcoded here or the value PROMPT can be entered. If PROMPT is entered, the NetView to NetView session user is prompted for the password on an SA OS/390 logon panel.

Initialize
A YES in this field means that SA OS/390 attempts to establish the NetView to NetView session as soon as logging onto the current domain is complete. If NO is specified, the user has to initiate the NetView to NetView session from the operator interface.

Description
This description of the domain in the To-Domain field is displayed on the operator interface panel.

Resident CLISTs Policy Object

| Note: | Since NetView 1.4, NetView automatically loads frequently used CLISTs into storage. You therefore no longer need to define your own resident CLISTs. |

A resident CLIST is a CLIST that is loaded into and kept in NetView storage for the first and all subsequent calls.

Resident CLISTs can improve performance by removing the overhead required to access CLISTs stored on a direct access storage device (DASD) each time the CLIST is called. Resident CLISTs are automatically loaded into main storage, using the NetView LOADCL command, during initialization of the automation environment. Refer to Tivoli NetView for z/OS Automated Operations Network User’s Guide for more information on using the LOADCL command for CLISTs.

If a new automation control file is loaded after SA z/OS is up and initialized, and that automation control file contains fewer automation procedures in the resident portion than the original automation control file loaded at initialization, then even if the procedures are not specified in the new file, they will not be dropped from storage. For example, if the original automation control file contained procedures A, B and C to be loaded resident, and the new automation control file contains procedures A, B, and D to be loaded resident, procedure D will be loaded but procedure C will not be dropped. [Figure 227 on page 289] shows the policy items that are displayed when a new Resident CLIST entry is defined or when an existing Resident CLIST entry is selected from the Entry Name Selection panel.
The unique policy item for Resident CLIST sets is:

**CLISTS**

This option allows you to enter the names of the CLISTs you want to be made resident during initialization of SA z/OS.

### Status Details Policy Object

The status display facility (SDF) provides a display of automated systems and resources using assigned status colors. An operator monitors the status of automated systems and resources by viewing the SDF main panel. Typically, for example, an application shown in green on an SDF status panel indicates the application is up, while red indicates the application is stopped or in a problem state. Operators can use the SDF to monitor the system and decide which actions to take when problems occur. This section describes the entries used to define the SDF in the customization dialog. You only need to change these entries if you use values other than the SA z/OS-provided defaults. The entries specified here define statuses and the priorities assigned to those statuses. These entries are used by SA z/OS common routines to gather data for requests to add status descriptors to status components. Additional parameters manage existing status descriptors and eliminate redundancy.

**Note:**

Default SDF entries are provided in all sample policy databases and will default appropriately even if they are not included in an automation control file build.

[Figure 228 on page 290](#) shows the policy items that are displayed when a new Status Details policy object is defined or when an existing Status Details object is selected from the Entry Name Selection panel.
The unique policy item for Status Details is:

**SDF DETAILS**

You can use this policy item to define colors and highlights associated with each status.

Refer to “More about Policy Item SDF DETAILS” for more information about this policy item.

For more information on SDF customization, see System Automation for z/OS User’s Guide

**More about Policy Item SDF DETAILS**

A panel like the following is displayed if you select the SDF DETAILS policy item from the Policy Selection panel for Status Details.

**Figure 229. Status Display Facility Details Panel**

The fields are:

**Status** Any required name. The name must start with an alphabetic or numeric character or one of ‘@’, ‘#’, or ‘$’. All other characters must be alphabetic, numeric, or one of ‘@’, ‘#’, ‘$’, ‘?’, ‘_’, or ‘€’. This value can be up to 8 characters long. It is recommended that you use the SA z/OS default statuses.
Priority

The priority number assigned to a particular status type. PRIORITY must be a valid number in the range specified in the AOFINIT member. Normally, priority determines the color displayed for a status. This is the recommended method. If the priority is not defined, the default is 99999999.

Highlight

The type of highlighting used for a particular status type. If this field is not specified, or HL=P is specified, highlighting is determined by the panel definition.

Color

The color used for a particular status type. This is the color override entry. Use this parameter to override the priority and color relationship defined in the AOFINIT member. Although the color of the status descriptor is changed, the priority is still determined by the order in which the status descriptor is chained.

There is no default color associated with an SA z/OS status. Any color specifications defined using this parameter override the color defined for the range in which the priority falls.

Clear

The CLEAR parameter helps manage redundant status descriptors. With SA z/OS, the reference value on a status descriptor is usually the status component name. This option is only available for statuses defined in the automation control file and accessed by SA z/OS common routines described in System Automation for z/OS Programmer’s Reference. The following values are allowed:

- CLEAR=Y
  Deletes any existing status descriptor for this status component with the same reference value as the status descriptor being added. This verifies that, for any given status component, there is only one status descriptor reflecting the true status.

- CLEAR=(Y,RV*)
  Performs the same function as CLEAR=Y, and also deletes status descriptors with reference values prefixed with the same reference value (followed by anything else). For example, if the existing status descriptor for RMF has a reference value of RMF1, and the status descriptor being added has a reference value of RMF, the RMF1 status descriptor is deleted before the new RMF status descriptor is added.

- CLEAR=(Y,RV)
  Performs the same function as CLEAR=(Y,RV*) but the reference value must be the same (not a wildcard).

- CLEAR=(Y,RV*,DA)
  Performs the same function as CLEAR=(Y,RV*), and also deletes all status descriptors with similar data in the data field. (The data field is defined with the DATA keyword on the SDF request that adds the status descriptor to a chain of status descriptors.)

- CLEAR=(Y,RV2*,CO)
  Performs the same function as CLEAR=(Y,RV*) with the following differences:
  1. The color of the existing status descriptor must be the same as that specified in the color entry before it is deleted.
2. Status descriptors having reference values prefixed with the new reference value followed by the number 2 are deleted, provided the color matches the status descriptor being added.

- CLEAR=N
  Indicates that redundant status descriptors are not deleted.

**Srv**
Determines the format of the reference value generated for this entry. If Y is specified for this entry, the generated reference value has the following format:

```
Status_component/Status
```

**Req (no add)**
Checks the SDF request. REQ=NOADD means the request is not added.

### Notes:

1. Choosing reference values carefully simplifies how SDF handles adding and deleting status descriptors.
2. Always use the status component either as the reference value or as a prefix to a reference value.

---

**Includes Policy Object**

This policy object is available only for cleanup or for SA z/OS systems below version 2 and will no longer be supported. The BUILD function will still include this data set member in the automation control file, but SA z/OS ignores all user-defined members and issues an error message.

**User E-T Pairs Policy Object**

This policy object allows you to create your own entry-type pairs and associated keywords in the same format as those used by SA z/OS.

You can use **User E-T Pairs** to store data for your own automation procedures in the automation control file. **User E-T Pairs** are provided for situations in which there is no suitable, predefined SA z/OS policy object. Using the automation control file to store policy-specific information helps keep your automation procedures generic.

[Figure 230 on page 293](#) shows the policy items that are displayed when a new **User E-T Pairs** policy object is defined or when an existing **User E-T Pairs** object is selected from the **Entry Name Selection** panel.
User E-T Pairs Policy Object

The unique policy item for User E-T Pairs is:

E-T DATA

This policy item enables you to specify entry-type pairs for the set. You can also enter a description for each pair. Refer to "More about Policy Item E-T DATA" for more information on this policy item.

More about Policy Item E-T DATA

Figure 231 shows the UET Entry-Type Selection panel.

When you enter an E for Edit in the Action column, another panel, similar to Figure 232 on page 294 is displayed in which you can enter the keywords and user data for each Entry-Type pair.

A keyword is an identifier for the data that follows it. The set of keyword/data pairs may be null, and each keyword does not have to be unique; that is, each keyword may have multiple sets of data assigned to it.

The following rules apply for the user data:

- Data containing the special characters (`,`)," or imbedded spaces must be enclosed in delimiters. Valid delimiters are single quotation marks, double quotation marks, or parentheses.
- Any imbedded single quotation marks will be automatically doubled by the processing logic of the UET Keyword-Data Specification panel. For example, ABC’DEF’GHI will be changed to ABC’DEF’GHI
User E-T Pairs Policy Object

- Imbedded single quotation marks must appear in pairs.

Figure 232. UET Keyword-Data Specification Panel

From within a CLIST, you can retrieve the data entered in the UET Keyword-Data Specification panels using the ACFFQRY command. An example is provided in “How to Use User E-T Pairs” on page 72.

More than one set of User E-T Pairs may be attached to a system, but it is recommended that you attach only one to avoid data being overwritten during ACF LOAD.
Chapter 6. Policy Database Maintenance

Option 4 (Policies) of the Customization Dialog Primary Menu panel allows you to maintain the list of your policy databases and to access various panels for selected policy databases.

Policy Database Selection List

If you have already created one or more policy databases, they are listed in the Policy Data Base Selection panel, as shown in Figure 233.

Figure 233. Policy Data Base Selection Panel

Databases created with the NEW command automatically appear in your policy database list. In addition you can add other databases. To add a database, you need to know the name of its data set and have the appropriate authority to update it. You can work with one policy database from your list at a time.

To select a policy database:
1. Move the cursor to the line containing the database you want to use.
2. Type s in the Action column on that line.

The selected policy database is opened, and the Entry Type Selection panel for that database is displayed. At this point you are ready to start defining your enterprise.

The commands and actions available from the Policy Data Base Selection panel are:

(O) OPEN
This option lets you access the policy data for that database. The Entry Type Selection panel is displayed. With this option, you can edit the contents of policy database. See Figure 2 on page 7.

(N) NEW
This option lets you create a new policy database as described in "Creating a New Policy Database" on page 297. The Create a New Policy Database panel is displayed. See "Creating a New Policy Database" on page 297 for more details.

(E) EDIT
This option lets you change the Enterprise name and data set name of your policy database for the build, report, and log output. To change the data within the database, you must OPEN it. Using the EDIT option, you can
Policy Database Maintenance

also start and stop the logging of user activity. For more information, see “Specifying Settings for Policy Databases” on page 301.

(D) DELETE

This option lets you delete the policy database data set, and all references to the policy database. SA z/OS will display a panel asking you to confirm your request.

SA z/OS does not provide any extended security to prevent you from deleting policy database data sets. This must be provided by a System Authorization Facility (SAF) product such as RACF®.

Note:

SA z/OS does not allow you to delete more than one policy database at a time. If you enter multiple delete requests in the Action column, the first request is processed and the remainder are ignored.

(A) ADD

This option lets you add a policy database to the displayed list. You enter a policy database name, an enterprise name and the name of a policy database data set that already exists.

(M) REMOVE

This option lets you remove the policy database details from the displayed list. The data set itself is not affected.

Note:

If you remove a policy database from all display lists, the policy database still exists as an z/OS data set.

(B) BUILD

This option displays the Build Functions Menu. You can build various configuration files. This function is explained in Chapter 8, “Building and Distributing Control Files,” on page 313.

(G) MANAGE

This option will bring you to the Data Management Select panel shown in Figure 249 on page 327 which allows you to import policies or migrate an automation control file to a policy database of the current release of SA z/OS.

You can find more information about this option in Chapter 9, “Data Management,” on page 327.

(R) REPORT

This option displays the Report Selection Menu for the selected policy database. You can create a report of all automation data in the policy database. The report offers options to report either the full automation policy, or to select a specific entry type and name to be reported.

You can either create an unformatted file to be viewed in a TSO session, or an HTML file to be viewed by an Internet Browser.

The following reports can be created:

• A report for all objects in the policy database, that will not be used for building the system operations configuration files (unlinked entries and relationships).
Creating a New Policy Database

A policy database is a partitioned data set (PDS) that contains policy data in its members. You create a new policy database from the Policy Data Base Selection panel by selecting 4 Policies from the Customization Dialog Primary Menu (this is the first panel you see when you start the SA z/OS customization dialog). Initially the Policy Data Base Selection panel shows no policy databases. Otherwise it shows all available policy databases. Figure 234 shows an example. You can see the name of the policy database together with its data set name on the Policy Data Base Selection panel if you select the “Name and data set name” option (option 2) from the VIEW pull-down. You can also choose to view just the policy database name by selecting option 1 from this pull-down.

<table>
<thead>
<tr>
<th>MENU</th>
<th>COMMANDS</th>
<th>ACTIONS</th>
<th>VIEW</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFGPD</td>
<td>Policy Data Base Selection</td>
<td>Row 1 to 3 of 3</td>
<td>SCROLL===&gt;</td>
<td>PAGE</td>
</tr>
<tr>
<td>Command ===</td>
<td>Action</td>
<td>PolicyDB Name</td>
<td>Enterprise Name/Data Set Name</td>
<td>DATABASE_NAME</td>
</tr>
<tr>
<td></td>
<td>RESOURCE_PDB</td>
<td>'MY_ENTERPRISE'</td>
<td>USER.RESOURCE.PDB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SH_PolicyDB</td>
<td>SH_ENTERPRISE</td>
<td>USER.SH.POLICYDB</td>
<td></td>
</tr>
</tbody>
</table>

Figure 234. Policy Data Base Selection Panel

To create a new database, type N (New) on the command line. A panel similar to Figure 235 on page 298 is displayed.
Creating a New Policy Database

Figure 235. Create a New Policy Database Panel

The fields for which you must enter data are listed below. For those fields that are not explicitly explained, refer to the appropriate z/OS literature.

**PolicyDB Name**

The name of this policy database, up to twenty characters in length.

**Enterprise Name**

The name of the enterprise for which this policy database is being defined, up to twenty characters in length.

**Data Set Name**

A valid z/OS data set name for the policy database. Standard rules for quoted and unquoted data set names apply. A quoted data set name is regarded as absolute. An unquoted data set name has your user ID prefixed automatically.

**Managed Storage**

This indicates whether the data set should be allocated on managed storage. If this field is set to YES, the fields marked with an * are used to allocate the data set. If NO is specified then these fields may be specified but are not used.

**Management class**

This specifies the management class that should be used to obtain the data management related information for the allocation of the data set.

**Storage class**

This specifies the storage class that should be used to obtain the storage related information for the allocation of the data set. The following data, related to storage class, can be specified if you are not using managed storage or you have not specified a storage class or if you wish to override the default storage class values.
Volume serial
The volume on which the database is to be stored. If this field is left blank, it defaults to your local system default.

Data class
This specifies the data class that should be used to obtain the data related information for the allocation of the data set. The following data, related to data class, can be specified if you are not using managed storage or you have not specified a data class or if you wish to override the default data class values:

Space Units
Valid allocation units for the data set are:
• CYLINDERS (this is the default)
• TRACKS
• BLOCKS
• MEGABYTES
• KILOBYTES

Primary quantity
This specifies the primary space allocation in space units. The default is 1.

Secondary quantity
This specifies the secondary space allocation in space units. The default is 1.

Directory blocks
Number of blocks to allocate for the directory. This field is only used when allocating a PDS. The default is 50.

Record format
This is FB and cannot be changed.

Record length
This is 80 and cannot be changed.

Block size
This defines the block size for the data set. It must be a multiple of 80.

Data Set Name type
This indicates whether the data set is to be allocated as a partitioned data set (PDS) or partitioned data set extended (LIBRARY).

Device Type
The type of storage device. This field defaults to SYSDA.

Note:
Many installations regularly delete data sets created on SYSDA volumes. If you want to create a permanent policy database you should not create it on a SYSDA volume. If the data set for a policy database is deleted, you will receive an error message when you attempt to open it. You can either recreate the data set if you have a backup, or enter M (REMOVE) against the policy database to remove it from your policy database list. Entering D (DELETE) will result in an error, since the data set has already been deleted.
Creating a New Policy Database

Model PolicyDB name
Enter the name of the policy database you want to use as a model for the new policy database. You can use one of the SA z/OS sample policy databases, or you can use a database you have already created. Enter "?" to see a list of the available databases. The Select Model Policy Database panel is displayed, from which you can select the database you want to use.

Note:
The DASD allocation figures shown in Figure 235 on page 298 are a suggested minimum only. Your storage requirements will depend on the size of your enterprise.

When you have finished entering the required data and have chosen a model database, enter the END command to create the new database. SA z/OS displays a Command Progress Display panel. When database creation is finished, the Entry Type Selection panel is displayed for the new database. See Figure 2 on page 7 for an example. If you want to leave the Create a New Policy Database panel without creating a new policy database, enter the CANCEL command.

Using Sample SA z/OS Policy Databases
SA z/OS provides sample enterprise organizations implemented as SA z/OS policy databases. Sample databases are installed when the SA z/OS customization dialog is installed. To use the samples, you must create a new policy database and select a sample database as a model.

The asterisk character (*) at the start of the database name indicates that this is a sample database.

The sample databases are:

*EMPTY
This sample is simply an empty database.

*MULTISYS - Multiple z/OS Images, Non Sysplex
This sample consists of several z/OS systems and standard applications containing basic definitions of the standard SA z/OS test configuration.
There is also a complete processor operations environment that includes a PSM processor and VM guest system, and a sample class definition of processor operations including IXC message automation.
In addition, it contains a sample class definition.

The following applications are included:
- APPC
- ASCH
- GMFHS
- JES2
- JES3
- LLA
- NetView
- RMF
- RODM
Specifying Settings for Policy Databases

Specifying the Edit option from the Policy Data Base Selection panel for a policy database leads to the Settings for Policy Data Base panel, as shown in Figure 236 on page 302.
Specifying Settings for Policy Databases

The first set of fields on this panel allows you to specify enterprise and data set names, as follows:

Enterprise Name
This allows you to re-specify the enterprise name. If this policy database has also been added to the Policy Data Base Selection list of other users they need to REMOVE or ADD the policy database to get the new name.

PolicyDB data set name
This allows you to re-specify the enterprise data set name.

Build output data set
This allows you to specify where you want the configuration files that are created by the System Operations build to be stored.

Report output data set
This allows you to specify where you want the output, created by the REPORT function, to be stored. It is not recommended that you store the output as a member of the policy database itself due to record length problems.

If you leave the field empty, SA z/OS allocates a data set with a default name (the policy database data set name extended with .REP as the low level qualifier) and a default size.

If you do specify a data set name, SA z/OS allocates a data set with that name and a default size.

If you preallocate a data set and specify the name here, then SA z/OS uses that data set. It is recommended that you use format VB and a record length of 1000 to avoid truncations.

The second set of fields allows you to set logging options for this policy database. See “Logging Policy Database Modifications” on page 303 for more details.

The last field on this panel, AT Scope, allows you to define the scope of a NetView Automation Table. Valid definitions are:

NONE
No NetView automation table is built.

ENTERPRISE
One NetView automation table is built for the entire policy database.
Specifying Settings for Policy Databases

SYSPLEX
One NetView automation table is built per Sysplex Group.

No NetView automation table is built for systems that are not linked to a sysplex group.

Note:
When the AT Scope is set to SYSPLEX, it is strongly recommended that you link all systems contained in the policy database to a group (GRP) of type ‘SYSPLEX’.

If this is not done, no AT is built for systems that are not linked to groups at all, or that are linked to groups of type STANDARD. A corresponding warning is issued at configuration build time.

If an AT was not built for a system, the hard coded AT, named INGMSG02, as it is delivered with SA z/OS 2.3, is loaded at automation run time.

SYSTEM
One NetView automation table is built per system (default). If the AT scope is changed to any value except NONE a build of type ALL will be performed.

Logging Policy Database Modifications
With SA z/OS you can write a log entry whenever a modification has been made for a specific Policy Database by using the policy database activity log option on the Settings for Policy Data Base panel (see Figure 236 on page 302). Fill in the following fields:

Write log entries
Turns logging ON or OFF for this Policy Database. The default value is OFF.

Log data set name
Allows the specification of two data sets for the activity log. Whenever a data set runs out of space, the dialog switches automatically to the other data set and continues writing the log records there.

If you leave this field blank, SA z/OS allocates data sets with a default name (the policy database data set name extended with .LOG1 or .LOG2 as low level qualifier) and a default size.

If you specify a data set name, SA z/OS allocates a data set with this name and a default size.

If you preallocate a data set and specify the name here, then SA z/OS uses this data set.

It is recommended that you use format VB and a record length of 250 to avoid truncations.

If the Settings for Policy Data Base panel is invoked and the specified policy database data set cannot be accessed, the log option fields are initially empty.

Whenever the policy database data set name is changed, the current log options are read from the policy database and displayed on the panel.
The following information is logged:

- Editing a policy item (for example, changing an automation flag or a command)
- Changing a link, which includes
  - Entry type links, for example, a link between an application and an application group
  - Links of instances and classes
  - Resource links
- Adding events to a trigger
- Invoking the MIGRATE and MERGE commands (these commands change the policy database content)
- System operations BUILD, because it writes build statistics back into the policy database
- Creating and deleting objects
- Copying objects
- Errors (as long as a policy database is open)

The following information is not logged:

- Option changes (using selection 0 Settings)
- Data set name changes (EDIT)
- REPORT function invocations
- VIEW functions (for example, view build report)
- Processor operations BUILD
- Initial conversion of policy database (when invoked the first time with SA z/OS)

A log entry includes the user ID, timestamp, and the policy item that was changed, or the command name and parameters if available. An example of an activity log is shown in Figure 237 on page 305.
<table>
<thead>
<tr>
<th>Date/Time</th>
<th>Event Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2004/04/21 16:50:35</td>
<td>RTL: Log Data Set Created</td>
</tr>
<tr>
<td>2004/04/21 16:51:01</td>
<td>RTL: PDB activity logging turned on</td>
</tr>
<tr>
<td>2004/04/21 17:05:39</td>
<td>RTL: APL AUTOMATION_MANAGER EDIT</td>
</tr>
<tr>
<td>2004/04/21 17:05:50</td>
<td>modified policy LINK TOInstances</td>
</tr>
<tr>
<td>2004/04/21 17:05:51</td>
<td>delinked from INSTANCE CICS_FOR_ENTRY</td>
</tr>
<tr>
<td>2004/04/22 08:40:45</td>
<td>RTL: APL AUTOMATION_MANAGER_1 EDIT</td>
</tr>
<tr>
<td>2004/04/22 08:41:17</td>
<td>modified policy DESCRIPTION</td>
</tr>
<tr>
<td>2004/04/22 08:41:36</td>
<td>modified policy AUTOMATION INFO</td>
</tr>
<tr>
<td>2004/04/22 08:41:45</td>
<td>modified policy LINK TO CLASS</td>
</tr>
<tr>
<td>2004/04/22 08:41:46</td>
<td>delinked from CLASS AUTOMATION_MANAGER</td>
</tr>
<tr>
<td>2004/04/22 08:41:52</td>
<td>modified policy LINK TO CLASS</td>
</tr>
<tr>
<td>2004/04/22 08:41:53</td>
<td>linked to CLASS AUTOMATION_MANAGER</td>
</tr>
<tr>
<td>2004/04/22 08:42:16</td>
<td>TMR ALL_TIMERS EDIT</td>
</tr>
<tr>
<td>2004/04/22 08:42:26</td>
<td>modified policy WHERE USED</td>
</tr>
<tr>
<td>2004/04/22 08:42:27</td>
<td>delinked from SYS AOCMIG1</td>
</tr>
<tr>
<td>2004/04/22 08:42:27</td>
<td>linked to SYS 99999</td>
</tr>
<tr>
<td>2004/04/22 08:42:51</td>
<td>modified policy TIMERS</td>
</tr>
<tr>
<td>2004/04/22 08:43:18</td>
<td>delinked from SYS AOCMIG1</td>
</tr>
<tr>
<td>2004/04/22 08:43:30</td>
<td>APL APG_AOCSYSM1 DELETE</td>
</tr>
<tr>
<td>2004/04/22 08:45:13</td>
<td>APL APG_AOC_ALLSYS NEW</td>
</tr>
<tr>
<td>2004/04/22 08:45:26</td>
<td>modified policy WHERE USED</td>
</tr>
<tr>
<td>2004/04/22 08:45:26</td>
<td>linked to GRP AOCPLEX</td>
</tr>
<tr>
<td>2004/04/22 08:45:48</td>
<td>modified policy APPLICATIONS</td>
</tr>
<tr>
<td>2004/04/22 08:45:48</td>
<td>linked to APL OPC_DATASERVER_ENTRY</td>
</tr>
<tr>
<td>2004/04/22 08:45:49</td>
<td>linked to APL OPC_DUMY1</td>
</tr>
<tr>
<td>2004/04/22 08:45:50</td>
<td>linked to APL OPC_DUMY2</td>
</tr>
<tr>
<td>2004/04/22 08:46:05</td>
<td>* * BUILD</td>
</tr>
<tr>
<td>2004/04/22 08:46:05</td>
<td>options: 1 REPORT.CFG ONLINE MODIFIED NORMAL</td>
</tr>
<tr>
<td>2004/04/22 08:47:18</td>
<td>terminated: return code 0</td>
</tr>
</tbody>
</table>

**Figure 237. Example of an Activity Log**
# Chapter 7. Creating Reports about Policy Databases

If you select option 3 Report of the Customization Dialog Primary Menu shown in "How to Start the Customization Dialog" on page 12, the panel shown in Figure 238 is displayed, where you can create reports about various SA z/OS objects.

![Figure 238. SA z/OS Report Selection Menu](image)

You can select the following:

## 1 Report

With this option, you can create a report which includes all automation data of the selected policy database. SA z/OS displays the Policy Data Base Report Parameters panel, as shown in Figure 239 on page 308, where you can choose to create either the full automation policy report, or to create a report about a specific entry type and name.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Report</td>
<td>Create Policy Data Base Report</td>
</tr>
<tr>
<td>2 Unlinked</td>
<td>Create List of entries not linked in Policy Data Base</td>
</tr>
<tr>
<td>3 ViewUnlinked</td>
<td>View List of unlinked entries</td>
</tr>
<tr>
<td>4 Resources</td>
<td>Create List of generated Resources</td>
</tr>
<tr>
<td>5 ViewResources</td>
<td>View List of generated Resources</td>
</tr>
<tr>
<td>6 Unreferenced</td>
<td>Create List of members not referenced by configuration</td>
</tr>
<tr>
<td>7 ViewUnreferenced</td>
<td>View List of unreferenced members</td>
</tr>
<tr>
<td>8 StatusMsgs</td>
<td>Create Status Messages Report</td>
</tr>
<tr>
<td>9 ViewStatusMsgs</td>
<td>View Status Messages Report</td>
</tr>
</tbody>
</table>

### Note:

Entry type ICL is not available in the Entry Type field and this entry type is not included in the policy database reports that are produced.

As the report generation can take a long time, you can also run this function in batch mode. The output may be extensive, therefore a separate data set is recommended.

You can also specify the format of the report. FLAT creates an unformatted file to be viewed in a TSO session. The name of this file is $RPTFLAT. HTML creates an HTML file you can view with your internet browser. The name of this file is $RPTHTML. The default value for the output format is FLAT.

### Note:

Microsoft® Internet Explorer can only handle files with extension .HTML.
Creating Reports about Policy Databases

The Command Progress Display panel informs you about the progress of the report generation.

```
OPTIONS HELP
-----------------------------------------------
AOFGRPTP Policy Data Base Report Parameters
Option ===>
1 Create report for a complete enterprise
2 Create report for entry type or entry name
   Entry Type. ................. (*, ?, or type)
   Entry Name. ................. (*, ?, or name)

Report options:
   Output Data Set ........
   Mode ............ ONLINE (ONLINE BATCH)
   Format ............ FLAT  (FLAT HTML)

Job statement information: (used for BATCH report)
//AOFRPT JOB
//*
//*
```

Figure 239. Policy Data Base Report Parameters Panel

The example in Figure 240 on page 309 shows a small part of the data listed in the report output format FLAT for the application JES2.
2 Unlinked

This option lets you create a report of all entries within the current policy database that would be ignored during BUILD processing. This includes:

- all objects in the policy database that are not linked to a system or sysplex
- all non-MVS systems
Creating Reports about Policy Databases

- all relationships that are not used during a build

It does not include the entry type Processor (PRO).

3 ViewUnlinked

With this option, you can look at the report of all unreferenced objects created by option 2 Unlinked.

4 Resources

With this option, you can create a list of all generated resources. In the Command Progress Display you will see in which member of the current policy database data set the generated list has been stored.

5 ViewResources

With this option, you can view a list generated with option 4 Resources for each existing system.

---

Figure 241. List of Generated Resources

6 Unreferenced

This option creates a report of all members in the system operations build output data set, that are not referenced by the actual configuration anchored in the ACF map member AOFACFMP. It helps to identify members that can be deleted when you perform a data set cleanup. The report is divided in two sections:
1. The first section lists all unreferenced members that have been identified as valid configuration files, together with some additional information such as entry name, or build time.

2. The second section lists all other unreferenced members, where either no build statistics could be found, or which have not been identified as valid configuration files.

Report members created by the Customization Dialog itself such as build report member $BLDRPT are not included in the report. To run this report, a successful build is required. Otherwise the control file fragments may be damaged and incomplete, and the content of the report is unpredictable.

7 View Unreferenced
With this option you can read the report of all unreferenced members created by option 6.

![Figure 242. Unreferenced Member Report](image)

8 StatusMsgs
Select this option to create a Status Messages Report. This includes Status Messages that are predefined by SA z/OS as well as user-defined Status Messages. The report shows the message ID, message description, and resource status indicated by this message and, if it is defined in the policy database, the entry name and the entry type that the message has been defined for. An example of the start of the Status Messages Report is shown in Figure 244 on page 313

---

Note:
The existing Policy Data Base Report remains unchanged.
Selecting option 8 produces the response as shown in Figure 243.

Figure 243. Command Progress Display for the Status Messages Report

Pressing Enter returns you to the Report Selection panel from where you can view the report by selecting option 9.

9 ViewStatusMsgs

Select this option to view the Status Message Report that was created by using option 8. If no report has been created, a message will inform you of this. An example of the start of the Status Messages Report is shown in Figure 244 on page 313.

In the report, user-defined messages are listed with their associated entry name and entry type, whereas those messages listed without these columns are predefined SA z/OS messages.
Chapter 8. Building and Distributing Control Files

How to Define Data Sets for Build Processing . . . 314
How to Build the Configuration Files . . . . . . 314
Building the Processor Control File . . . . . . . . . 315
   Usage Notes. . . . . . . . . . . . . . . . . . . . . . . 316
Building the System Operations Configuration Files 317
   System Operations Configuration Files Build
   Options . . . . . . . . . . . . . . . . . . . . . . . . . 318

Note:
Throughout this book the term automation control file is used to refer to the master include file and all the members that it includes.
Building and Distributing Control Files

After you have completed the SA z/OS customization process, the following data and control files can be built:

- Processor operations control file (needed for processor operations functions)
- System Operations configuration files (ACF, AMC, AT, and MPFLSTSA files), which are needed for automation
- NetView operator definitions

**Note:**
You must use the customization dialog to create the system operations configuration files. Do not edit any of the system operations configuration files manually.

**How to Define Data Sets for Build Processing**

To use the batch options of the build facility for automation control files, you must first specify the names of the data sets required for the batch processing. These names form the DD name concatenations to be included in the JCL of the batch processing job.

The data set names are saved in an ISPF table in the partitioned data set referenced by the AOFTABL DD name, for example, ING.CUSTOM.AOFTABL (allocated during installation of SA z/OS).

On the *Customization Dialog Primary Menu* panel of the customization dialog, select the *Settings* option. This brings up the *Customization Dialog Settings Menu* (see Figure 266 on page 370).

On this panel, choose option 11, **DSNAMES**. The *Data Set Customization Details* panel is displayed (see Figure 270 on page 375).

For more information, refer to “Defining Data Sets for Batch Processing” on page 375.

**How to Build the Configuration Files**

Exiting the customization dialog does not automatically build these files. You build them by selecting the BUILD (B) function either on the OPTIONS pull-down menu of the *Entry Type Selection* panel, or on the COMMANDS pull-down menu of the *Policy Data Base Selection* panel. The following panel is displayed:

```
<table>
<thead>
<tr>
<th>MENU</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFGBLDK</td>
<td>Build Functions Menu</td>
</tr>
</tbody>
</table>

Option ===>

1. ALL Build all files
2. Proc-Ops Build Processor Operations Control File
3. Sys-Ops Build System Operations Configuration Files
```

*Figure 245. Build Functions Menu Panel*

The *Build Functions Menu* offers the following functions to create control files:

1. **Build all files**
   
   Causes all files required by SA z/OS system operations and processor operations to perform all automation to be built. These files are:
How to Build the Configuration Files

- the processor control file
- the system operations Automation Control Files (ACF)
- the system operations Automation Manager Configuration File
- the NetView automation table (AT)
- the MPFLSTSA member

Selecting option 1 is equivalent to selecting options 2 and 3 sequentially.

2. **Build the Processor Operations Control File**

Causes the information required by processor operations to be generated in a processor operations control file. Also provides an option to specify the name of the message log for the processor operations control file generation process. The processor operations control file should be rebuilt after any changes to the definitions of enterprise, groups, subgroups, systems, processors, communications paths, or communications tasks. The processor operations control file should be built prior to building the system operations configuration files because the name of the processor operations file to be used for automation is placed in the ACF.

3. **Build the System Operations Configuration Files**

Causes enterprise automation data for SA z/OS system operations to be built. This data consists of:
- the automation control file (ACF), which is built into a set of file fragments and a master include file
- the automation manager configuration file (AMC file)
- the NetView automation table (AT)
- the MPFLSTSA member

### Building the Processor Control File

To create the processor control file, you can use option 2 on the *Build Functions Menu.*

When you request a build of the processor control file, it is built from information on the enterprise, groups, subgroups, systems, and processors, already entered in the policy database. This means that any time you change definitions of any of these, the processor control file needs to be rebuilt.

Use the panel shown in [Figure 246](#) to identify the fully qualified names of both the SA z/OS processor control file and the control file message log.

![Figure 246. Processor Operations Control File Build Panel](#)
Building the Processor Control File

For SA z/OS processor operations, the processor control file, and control file log must reside in a sequential data set.

This panel has the following sections:

1 Build Processor Control File

Builds the ProcOps Control File and writes all build messages in a separate log file. The following needs to be specified:

Control file
Identifies the data set containing the ProcOps Control File

Control file log
Identifies the data set containing information about a successful build or errors in the configuration definition.

2 Build NetView Operator Definitions

Builds a sample NetView Operator Definitions file. The number of target control tasks and message monitor tasks is calculated as the number of current processors that are defined with a connection protocol other than INTERNAL.

The file can be used to update the AOFOPFO sample that is included in the NetView DSIOPF member.

Data set name
Identifies the data set containing the NetView Operator Definitions sample file.

3 Browse Control file log

Allows you to view the ProcOps control file log containing all messages sent from the control file generation process. The log is accessed in browse mode.

Note:
For the resulting NetView operator definition file, each message monitor task and target control task is a NetView autotask. These autotasks must be defined to the NetView program to enable SA z/OS processor operations.

To retain any operator and autotask definitions that you already use, make sure you generate the SA z/OS definitions into a new data set or member. The generate function erases any statements in the chosen data set. Integrate the contents of the new NetView operator definition file into the DSIOPF member of the NetView DSIPARM data set. The statements can be anywhere in the NetView DSIOPF member.

The customization dialog generates automation operator definitions with an operator profile of ISQPROF. ISQPROF is provided in the SINGNPRF data set and should be added as a member of a data set in the DSIPRF path. If your data center has changed the ISQPROF profile name, you must change the profile name in each of the automation operator definition statements.

Usage Notes

The SA z/OS processor control file contains information specific to your SA z/OS configuration and is used to start SA z/OS processor operations. Before you generate the processor control file, you must allocate and catalog the data sets for it and its log. The INGEDLGA job does this for you early in the installation.
Building the Processor Control File

process. If you decide to use more than one control file before you generate each processor control file, you must allocate and catalog the data sets for them and their logs.

If the control file you specified already contains data, a confirmation panel is displayed to allow you to confirm whether you want the existing control file to be overwritten.

If the data set name of either the control file or control file log has been changed (using PF3), the values are stored into the enterprise policy database. To avoid this, exit the panel with CANCEL.

The processor operations control file should be built prior to building the automation control file because the name of the processor control file to be used for automation is placed in the ACF.

The enterprise member of the automation control file should be rebuilt after any change to the name of the processor control file. This name is used in the automated startup of the processor operations application.

After you have used the generation function, check the Control File Log panel by selecting the Browse control file log option on the Processor Operations Control File Build panel. This is option 3 in Figure 246 on page 315. To determine the correct response for any processor control file log messages, see System Automation for z/OS Messages and Codes.

Once you complete the actions recommended for all of the messages, generate the processor control file again.

The control file log receives messages from the control file generation process. These messages indicate why an attempt to generate a control file produced an incomplete control file or was unsuccessful. When a control file is generated successfully, a message to that effect is displayed, and no messages are written in the control file log.

To successfully generate an SA z/OS control file, you must have completed the required fields in all the configuration panels. If you try to generate a processor operations control file for incomplete definitions, a message is displayed that the control file is incomplete or was not generated. When this occurs, examine the control file log to determine what needs to be corrected.

Building the System Operations Configuration Files

To build a new enterprise automation configuration, that is, a set of new system operations configuration files (automation control file, automation manager configuration file, NetView automation table, and MPFLSTSA member), you have the following options to invoke the BUILD process:

- On the Policy Data Base Selection panel, type the command B into the Action column of the policy database for which you want to invoke the BUILD process. This will bring you to the Build Functions Menu (see Figure 245 on page 314).
- Use option 3 on the Build Functions Menu (Figure 245 on page 314). This will bring you to the Build Parameters panel shown in Figure 247 on page 319.
- From the Entry Type Selection panel, you can invoke the build functions either by choosing the BUILD (B) command from the command pull-down. This will bring you to the Build Functions Menu. Or, if you want to process a certain entry
Building the System Operations Configuration Files

type, you enter the BUILDF (F) command on the command line and specify the identifier for the entry type that you want to build. For example, enter: BUILD SYS to build all defined system objects. You are presented with the Build Parameters panel (see Figure 247 on page 319) with the entry fields filled in according to your previous selection. The build function builds all systems irrespective of whether they have been modified or not.

• From the Entry Name Selection panel, you can enter the BUILDF (or F) command in the action column against the object you want to build. For example, to build the JES2 Subsystem enter F against JES2 in the Entry Name Selection panel for Application policy objects.

Optionally, you can enter the command on the command line, for example:

F JES2

You get the Build Parameters panel (see Figure 247 on page 319) with entry fields filled in according to your previous selection.

A Command Progress Display panel shows the progress of the build. If errors are encountered during the build process, the Command Progress Display will display a message. The automation configuration files are placed in the data sets that you defined in the Output Data Set field.

Every new build generates the automation configuration map AOFACFMP which contains

• for each system one ACFxnnn system include member which is used by the automation agents

• for each sysplex one HSAxnnn sysplex include member, which describes the automation manager configuration

The ACF, AMC, AT, and MPF build will always be invoked together. This is due to the fact that the configuration data always needs to be synchronized so that the automation agents and the automation manager operate on the same data.

Each automation control file fragment contains information about a specific resource defined through the dialogs. Each z/OS system in the enterprise has its own master include file. The master include file references the automation members that are appropriate for that system. The name of the master include file for each system is created by SA z/OS. It starts with prefix ACF followed by four alphanumeric characters.

System Operations Configuration Files Build Options

The Build Parameters panel allows you to specify the information required to build the system operations configuration files that are required for automation. You can build these system operations configuration files for standalone systems, individual sysplexes or your complete enterprise, using the options shown in Figure 247 on page 319.
Building the System Operations Configuration Files

Figure 247. Build Parameters Panel

These options are:

1. Builds all System Operations automation configuration files for a complete enterprise.

   **Note:**
   This option must be successfully completed at least once before you can use options 2 or 3.

2. Allows the selection of a single sysplex group or a standalone system.
   This build option performs a complete build of all objects linked to the selected sysplex or standalone system. For a big policy database, this option helps to reduce build time for small changes, while creating a complete set of configuration files that are ready for automation.

3. Allows you to specify an entry type and name to be rebuilt. Only the ACF fragments for that entry will be updated.

   If for example, a timer object is modified and linked to another system, a build of this object with option 3 would only build that fragment and update the timestamp in the system include member, but not build the link changes for the system.

   The same is true if an object is renamed. A build with option 3 of the new name would not add this new object to the system include member. Instead a build with option 2 is required.

   For entry type *System*, this has the consequence that a selective build is rejected if there is also a link change pending.

   However, if this option is used with entries of type *Events*, *Triggers*, *Service Periods*, *Application*, *ApplicationGroup*, *System*, or *Group*, a check is made as to whether data that affect the automation manager has been changed. In this case, building only the ACF fragment for the requested entry would mean that your previously generated AMC files would no longer be compatible with your current ACF files. To avoid this, a complete build (using build type MOD) will
Building the System Operations Configuration Files

automatically be invoked. Before the build actually starts a confirmation panel is displayed explaining this situation, and allowing you to defer the build.

You will see the same behavior with automation tables if this option is used with entries of type APL or MVC. A check is made as to whether data has been changed that affect the NetView automation table. If so, a confirmation panel is displayed allowing you to defer or to start a complete build (using build option MOD).

If you enter a '?' into this field, SA z/OS will display a list of either entry types and/or a list of available entry names for the selected entry type.

4 Allows you to view the build report available on the build output data set. It invokes the ISPF VIEW function. If there is no build report, an empty screen is displayed.

Notes:

1. If a policy database is new, or the build output data set is empty, a full build is required to produce a complete set of system operations configuration files. This means a build with option 1 and Type = ALL needs to be issued. For all subsequent builds, option 2 or 3 and Type = MODIFIED can be used.

2. While build option 2 only updates a single system or sysplex, deleted standalone systems and sysplexes still remain in the configuration files. For example, a deleted sysplex or standalone system is not removed from the ACF map AOFACFMP. So, you may consider issuing a build with option 1 whenever a large number of parts has been deleted, to clean up unnecessary entries from the configuration files.

3. No fragments are built for processors of type PSM.

You can enter values for the following fields:

Output Data Set

Use this field to specify where you want to store the build output. The data set must be an existing partitioned data set.

If the data set is not empty, but already contains a control file, the policy database name and the configuration name within the data set are compared with those of the requested build. If they do not match, a message is issued and the build is not started.

Mode

Use this field to specify how to process the BUILD request:

ONLINE

Indicates that the BUILD should run in the foreground.

BATCH

Indicates that the BUILD should run in the background.

The default value is ONLINE (if this field is set to blank).

Type

Use this field to specify which entries are to be processed.

MODIFIED

Indicates that only those entries that have been changed since the last BUILD request should be processed.

With this option, the build distinguishes between:

- data changes in the ACF
Building the System Operations Configuration Files

- link changes in the ACF
- data changes in the AMC file
- link changes in the AMC file

Depending on the type of change, the affected objects are identified, and only the necessary parts of the configuration file are rebuilt.

For example:
- If a new timer is linked to one system, only the system include member for that system is rebuilt. For all other system include members only the header timestamps are updated, and the build of the AMC file, AT, and MPFLSTSA member is skipped, because timer data does not go to the automation manager configuration file, AT, or MPFLSTSA member.
- If a new application is linked to a Sysplex Application Group, the build determines which systems are linked to that sysplex, and the include members for those systems are rebuilt automatically, together with the AMC, AT, and MPFLSTSA member for that sysplex.
- If the AT Scope is changed, only the AT and MPFLSTSA member are built.

**ALL** Indicates that all entries are to be processed unconditionally, even if there has been no change to those entries since the last BUILD request.

If a build type “ALL” is selected together with build option 2, only the configuration files belonging to the selected sysplex or standalone system are rebuilt.

The default value is MODIFIED (if this field is set to blank).

The very first build after installing a new release must be a build of type ALL in order to get a complete set of configuration files for the automation manager and automation agent structure.

**Configuration**

This field allows you to select between alternative configurations.

**NORMAL**

indicates that only those entries linked to the systems of the NORMAL configuration are processed by the current build.

**ALTERNATE**

indicates that only those entries linked to the systems of the ALTERNATE configuration are processed by the current build. These systems must have been marked with a “+” in their MVS system name.

NORMAL and ALTERNATE configurations must be build into different data sets. The default value is NORMAL (if this field is set to blanks).

If you specified that the BUILD request should run in BATCH mode, then you need to provide a valid JOB statement for the batch job.

The information that you specify here is saved in your ISPF profile data set and is used as the initial values the next time you issue the BUILD command.
BUILD Error Messages

The messages in the popup panel showing the command progress indicate error and warning situations. In most cases, it should not be necessary to check the build output member for unusual entries or error indications. Furthermore, the message log can be viewed from the panel where the build has been issued.

There are build errors that may stop the build, for example, resources in a loop, and other messages that are tolerated, for example, duplicate application subsystem names linked to the same system. Terminating build errors are shown with the label ERROR in the message prefix section, while tolerated errors are shown with the label WARNING in the message prefix section.

When an error is shown, no usable set of configuration files has been created. It is necessary to correct the cause of the problem and run the build again.

When a warning is shown, the configuration files can be used for automation. However it is recommended that you fix the problem and run the build again.

A possible cause for a warning during ACF fragment build may be that there are policy database objects with missing or conflicting definitions, that are not critical for automation, for example, an application default entry (type ADF) without any definitions. In this case, the build function continues, but issues a warning message. This message is displayed whenever a full enterprise build is issued, and does not require that the application be linked to a system.

There is a summary message at the end of the AT build report, and specific messages that show, for example, the object name somewhere within the build progress messages. The specific message reads:

```
......WARNING. System MIGR_TEST includes the following entries with
............cont. an unsatisfactory build status in main member ACFZ995:
.................. -- SYSTEM_DEFAULTS (NUL)
```

The summary message at the bottom of the build message log reads:

```
......WARNING. Missing or conflicting definitions found for entry
............cont. types: SDF. The entry names are listed in the build
............cont. report.
```

Possible causes for a warning during system main include member build are:

- Classes with duplicate subsystem names linked to the same system. In this situation, the first subsystem is used, and all following subsystems with the same name are skipped. So these subsystems are not known to the automation agent, and may remain as initialized objects in the automation manager.

- Applications with duplicate MVS jobnames linked to the same system. The processing is the same as for duplicate subsystems.

- If missing or conflicting definitions are detected, which are not critical for automation, a warning message is shown and the entry name is listed together with the status NUL. There are only a few situations where this can occur, for example, an application default entry (type ADF) without any definitions.

- If an object is linked and has been previously built, however it does not exist in the build output data set, then a warning message is shown and the entry name is listed together with the status DEL. The build function continues, but the object is not included in the system operations control files.

Note that such a warning message may cause a rebuild of the objects in error and the linked systems and sysplexes. So, a build of type MOD may rebuild more
Building the System Operations Configuration Files

systems and sysplexes than you would expect due to the actual changes. Possible causes for a warning during automation manager configuration file build are:

- The AMC logic member could not be found. The build function searches in the sample PDB data set for the current logic member. If this data set is not allocated, or contains no logic member, it could be added to the Configuration Files by the user.
- A trigger is linked to the current sysplex, but no conditions have been specified. The trigger will not be added to the automation manager configuration file.
- One or more application groups are linked to the current sysplex without an automation name being specified. The build function continues, but no resources will be created for this application group(s).
- One or more supporting application resources are not linked to the system. The relationships are not added to the automation manager configuration file.

Possible causes for a warning during AT and MPFLSTSA build are:

- Conflicting status message definitions for the same message ID but on separate policy database entries
- AT entry override definitions for the same message ID for one policy database entry but a status message definition for different policy database entries
- The AT scope being set to SYSPLEX but the changes that affect the AT being made for systems that are not linked to any sysplex

It is possible to tailor the level of detail of the messages for the trigger, application group, and AT warning. When invoking option 0 Settings, there is a selection 3 MESSAGE that allows the message level to be set. With level 1 (which is the default), in the above situation just one summary message is displayed per manager configuration, whilst level 2 will display all details of the definitions causing the problem.

Furthermore, that panel offers an option for message logging. By default, this value is ON, which means that all important dialog messages, including messages shown in a command progress popup window, are written into the ISPF LOG. The value OFF suppresses this, and it may be useful to switch this message logging off, if a build is run with message level 2.

Independent from the message logging switch, error messages are always written to the ISPF LOG. You cannot turn this off.

If the user profile AOFPROF for batch build is not allocated, the default message level is 3 to write all possible messages to the batch report and the ISPF LOG.

**Batch Build:** It is possible to start a batch build independent from the customization dialog. So you can for example, add further job steps or you can automatically start the batch build. However, this build also needs exclusive access to the policy database, because build progress and statistics are written to the policy database while the job is running. A sample JCL of such a job is provided in member INGEBBLD of the SINGSAMP library. Build return codes are:

0 successful
4 build complete, however, some warning messages are issued. Also issued if user exit INGEX02 ended with a RC > 0.
8 user exit INGEX01 ended with RC > 0. Build processing not started. No configuration files built.
Distributing the System Operations Configuration Files

The SA z/OS automation configuration consists of four components:

- the automation manager’s automation manager configuration file (AMC)
- the automation agent’s automation control files (ACF)
- the NetView automation table (AT)
- the MPFLSTSA member

These files are also referred to as system operations configuration files in the customization dialog.

The partitioned data set generated by the BUILD contains these four components. SA z/OS runs on a sysplex basis, that is, the same ACF, AMC, and AT files must be available to all of the automation agents and automation managers in the sysplex.

The automation agent automatically finds the automation configuration data set and dynamically allocates it. It is highly recommended that you do not allocate the data set within DSIPARM. (For further information, refer to the section that describes using SA z/OS partitioned data sets in System Automation for z/OS Planning and Installation.) The automation configuration data set must either be shared or have exactly the same name on all systems within a sysplex. It can be shared within an enterprise.

You then have the choice to:

- Copy the members into the initial configuration data set that the automation manager was started with. Then issue an INGAMS REFRESH command with an * as the data set name.
- Issue an INGAMS REFRESH command with the name of the new partitioned data set. This will cause the automation manager and the automation agent to read from the new partitioned data set.

An automation agent needs to run on every system that is automated by SA z/OS and at least one automation manager needs to run per sysplex and one per standalone system. Automation managers can manage systems sysplex-wide. Figure 248 on page 325 shows a sample automation configuration and presents the build process.

The MPFLSTSA member must be copied to a z/OS parmlib data set.
Figure 248 shows a sample enterprise consisting of one sysplex Sysplex1 and one standalone system SYS5. The sysplex consists of four systems SYS1, SYS2, SYS3, SYS4 and has a primary and a secondary automation manager (AM). The sysplex shares a copy of the system operations configuration files and the standalone system has its own copy of the same files.

The sysplex in our example has one primary and optionally one secondary automation manager and one automation agent on each system. The standalone system must have one automation manager (AM) and one automation agent (AA).

The enterprise wide policy database for your sysplex and the standalone system is created and maintained with the customization dialog on any suitable system. For the purpose of the diagram shown in Figure 248 it is called the configuration focal point.
The configuration build is also performed with the customization dialog and the configuration files are distributed from there to the locations where needed.

Using the customization dialog’s BUILD function, you produce the following output:

- the Processor Operations Control File
- the System Operations Configuration Files

You could share one common automation configuration file in your enterprise, or you can copy the automation configuration file to where it is needed as shown in our scenario. Both automation managers feed the enterprise-wide RODM data base dynamically with information that is available from both AMCs and from information that becomes available during automation processing (like status changes). This feeding is done automatically by the SA z/OS topology manager after you have started it. The generated data is used by the NMC workstation to display predefined graphical views.

The automation manager’s AMC file and the automation agent’s ACF files must match. Therefore the automation manager and an automation agent exchange a token to verify that the automation configuration matches. You can share one common data set containing all automation manager configuration files and all automation control files within a sysplex.

---

**Note:**

It is mandatory to create and maintain policy databases with the customization dialog.

The automation configuration files must be distributed to each system in the SA z/OS enterprise where SA z/OS system operations functions are to run. In general, this is all of the z/OS systems in the SA z/OS enterprise. In a sysplex, the data set name of the automation configuration file must be equal on all systems in the sysplex.

However, it is highly recommended that you use a shared data set in a sysplex, and it is preferable that you use a shared data set for the whole enterprise.

---

**Generating a Job for Batch Build**

You can run a job for batch build that performs the same processing as the BUILD function of the customization dialog. The SINGSAMP sample library contains a sample job for this task in member INGEBBLD. Before submitting this job, you need to edit the jobcard appropriately as described in the sample itself.

The build can only run if the output data set exists and is allocated. If the data sets are not SMS-managed, allocating the data set within the same job step does not work because the cataloging is done by z/OS at end of the job (or job step). Therefore the allocation and cataloging of the build output data set must be done in a separate job (or job step) prior to the build job (or job step).
Chapter 9. Data Management

This chapter provides information about functions provided by SA z/OS to import selected entries from another policy database, or to migrate agent control file data into a policy database.

Panel Interaction

If you select option 5 Data Management from the Customization Dialog Primary Menu, you reach the Data Management Select panel, as shown in Figure 249.

Figure 249. Data Management Select Panel

This panel allows you to select how to manage the policy database data. Your options are:

1 Import

Allows you to select entry data from another policy database and to add them to the current policy database. See "Importing Policy Database Data" for more details.

Note: The entry data are imported with all their policies, except the link information and the resource definitions.

9 Migrate

Allows you to migrate an Automation Control File (Agent Data) or Automation Control File fragments (ACF fragments), that is automation agent data only, into an existing policy database. See "Migrating the Automation Control File" on page 329 for more details.

Note: This function should mainly be used to migrate an ACF or ACF fragments from an SA OS/390 release 1.3 or earlier.

Importing Policy Database Data

Selecting the import option leads to the panel shown in Figure 250 on page 328.

© Copyright IBM Corp. 1996, 2004
This panel allows you to specify the information required to import entry data from another policy database. It has the following options:

1 Import Policy Data
   This option allows you to select the following:
   - A policy database from which entry data will be imported. This field is initially empty. Entering a question mark (?) in this field and pressing Enter displays the Policy Data Base Selection panel.
   - An entry type for which selected entries will be imported. This field is initially empty. Entering a question mark (?) in this field and pressing Enter displays the Entry Name Selection panel.

   **Note:**
   The import of entry type ENT (enterprise policies) is not supported.

2 View import report
   This allows you to view the import report that is available in the target policy database. It invokes the ISPF VIEW function. If there is no import report, an empty screen is displayed.

The information that you specify here is saved in your ISPF profile data set and used as the initial values the next time you issue the import function.
Requesting Option 1 on this panel will take you to a list of the existing entries for the specified type (see Figure 251 for an example). Here you can select which entries are to be copied. After making your selections:

<table>
<thead>
<tr>
<th>COMMANDS</th>
<th>ACTIONS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFGIMPS</td>
<td>Entry Name Selection</td>
<td>Row 1 to 9 of 9</td>
</tr>
<tr>
<td>Command ==&gt;</td>
<td></td>
<td>SCROLL ==&gt; PAGE</td>
</tr>
<tr>
<td>Action</td>
<td>Status</td>
<td>Entry Name</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>ER0_ALERT</td>
<td>SELECTED</td>
<td>ICLI Alert Router</td>
</tr>
<tr>
<td>ER0_APPSRV</td>
<td></td>
<td>SAP R/3 Application Server Instance</td>
</tr>
<tr>
<td>ER0_CO</td>
<td></td>
<td>SAP R/3 Syslog Collector</td>
</tr>
<tr>
<td>ER0_DB2DVM1</td>
<td></td>
<td>DB2 for SAP; DBM1</td>
</tr>
<tr>
<td>ER0_DB2DIST</td>
<td></td>
<td>DB2 for SAP ER0; DIST</td>
</tr>
<tr>
<td>ER0_DB2IRLM</td>
<td></td>
<td>DB2 for SAP ER0; IRLM</td>
</tr>
<tr>
<td>ER0_DB2MSTR</td>
<td></td>
<td>DB2 for SAP ER0; MSTR</td>
</tr>
<tr>
<td>ER0_DB2SPAS</td>
<td></td>
<td>DB2 for SAP ER0; SPAS</td>
</tr>
<tr>
<td>ER0_ERS</td>
<td></td>
<td>SAP R/3 Enqueue Replication Server</td>
</tr>
</tbody>
</table>

Figure 251. Entry Name Selection Panel for Import Policy Data

and pressing ENTER, a panel is displayed where you can confirm your selections, as shown in Figure 252.

<table>
<thead>
<tr>
<th>COMMANDS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFGIMP2</td>
<td>Confirm Selected Entry Names For Import</td>
</tr>
<tr>
<td>Command ==&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SCROLL ==&gt; PAGE</td>
</tr>
<tr>
<td>Press ENTER to start Import Process.</td>
<td></td>
</tr>
<tr>
<td>Press CANCEL or PF3 to return.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Selected Names</td>
</tr>
<tr>
<td>ER0_APPSRV</td>
<td></td>
</tr>
<tr>
<td>ER0_DB2IRLM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 252. Confirm Selected Entry Names For Import Panel

Pressing ENTER will start the import processing, while CANCEL or PF3 will take you back to the previous panel.

Migrating the Automation Control File

For the automation manager, it is necessary to create the automation manager configuration file, which can only be built using the customization dialog. Thus all data must be available in a policy database.

If the customization dialog has not yet been used, and all automation definitions are stored in an ACF that has been edited manually, the MIGRATE function of the customization dialog should be used. It recognizes all valid ACF keywords, and merges the corresponding values into an existing policy database. In detail, the following steps are required:

1. If the customization dialog has never been used before, no user specific policy database exists. To create one, specify option 4 on the Customization Dialog Primary Menu to go to the Policy Data Base Selection panel, and specify the command NW. On the subsequent panel, a model policy database (PolicyDB) name is required. Entering a question mark in the corresponding field gives a list of samples, where one can be selected. A description of the samples can be found in "Using Sample SA z/OS Policy Databases" on page 300. This allows
you to select a sample that best matches your installation requirements. If a new policy database has been created, it will be listed on the Policy Data Base Selection panel.

2. On the Policy Data Base Selection panel, use the Data Management command for the policy database into which you want to copy the automation control file data. Then the Data Management Select panel will be displayed. On this panel, the entry name for the system that will be used as the target and the MVS system name needs to be specified. It is not necessary that this system already exists. If it does not exist, the MIGRATE function will create the system automatically.

In the Source Data Set field, enter the data set name and the member name that contains the ACF definitions (or the main member containing the include statements for the fragments) for the system to be migrated.

If you have multiple systems defined, each with its own ACF, they have to be migrated one after the other. It is not possible to migrate multiple systems at the same time, because a specific target system must always be specified.

3. Before the migration is started, specific objects can be selected or excluded from migration. However, if a complete ACF is to be migrated into a new policy database, it is not necessary to go into these details. By default, all entry types are selected, and the migration can be started without any further specification. The MIGRATE function will copy all source definitions into the corresponding entry types automatically.

4. If it is necessary to select specific objects for migration, the corresponding option on the Data Management Select panel has to be selected. This gives a list of all entry types found in the source data set. There each entry type can be selected or excluded for migration. And if it is necessary to be even more specific, for each entry type a list of the entry names found for that type can be displayed. There it is possible to check for duplicates, and rename entries, if required.

If a complete ACF is migrated into a new policy database, this selection of specific entry types and names can be skipped.

5. If the preparations are done, the migration itself can be started. As it may take some time to complete, the progress is indicated on a special screen that is updated automatically and shows all migration steps as they are executed. Furthermore, a report of the migration will be written and stored as member in the target data set. The member name needs to be specified on the Data Management Select panel.

6. All data belonging to one entry in the ACF are copied into one entry within the target policy database. Depending on the specification in the ACF, it may be more or less difficult for the migration to find the name of such an entry.

7. Any lowercase characters encountered in the automation control files that are being migrated will be translated to uppercase unless they are enclosed in matching pairs of either " or ".

Note: Migration of MTR resources is not supported.

You have the following alternative methods of invoking the migration function:

- From the Customization Dialog Primary Menu [Figure 5 on page 12] select option 5 Data Management.
Data Management

- From the Policy Data Base Selection panel, you can select action Migrate (G) against the policy database to that you want to migrate your data to.

In both cases, you reach the Data Management Select panel shown in Figure 249 on page 327 that you can use to migrate an automation control file into a policy database of the current release of SA z/OS.

For every enterprise (every policy database) you migrate on a System level, that is you migrate each single automation control file main include fragment for each system.

**Note:**
The more selectively you specify your migration process the better the performance will be. For example, specifying a certain automation control file fragment instead of specifying the main include member may be useful if you only want to migrate certain parts of an enterprise. The various selection methods that are available are described below.

Selecting the migrate option leads to the panel shown in Figure 253.

![Figure 253. System Migration Definitions Panel](image)

This panel allows you to specify the information required to migrate a system’s data from an automation control file fragment into a target policy database. The system name you specify in the System Entry Name field, is either created in the target policy database or, if it already exists, it is overwritten. You have the following options:

1. **Select Migration Objects**
   This option allows you to specify the selective migration of objects from ACF fragments. It leads to the System Object Migration panel (Figure 254 on page 334) where you can select those entry types that you want to migrate to the current policy database whilst explicitly excluding the rest of the inputs.
Within the selected entry types you can later select certain existing entry names that you want to migrate. To achieve this, enter the command EDIT (E) against a selected entry type. The Entry Name Migration panel (Figure 255 on page 335) is then displayed.

After you have selected all objects (entry types and entry names) for migration, return to the Data Management Select panel where you can now start the migration using one of the two following Start Migration options.

- **2 Start Online Migration**
  This option immediately starts the migration of the specified ACF fragment to the current policy database in the foreground.

- **3 Start Batch Migration**
  This option starts the migration process in the background. In this case, you need to provide a valid JOB statement in Job statement information on the same panel.
  
  For definitions of the data set allocations needed for batch migration, see “Defining Data Sets for Batch Processing” on page 375.

- **4 View Migration Report**
  This option invokes the ISPF VIEW function to view the migration report that was created as specified in the Report member field for this target policy data set.

**Notes:**

1. If option 1, 2, or 3 is entered the first time, all ACF data is read and saved for further processing until you exit this panel.
2. If option 2 or 3 is used after option 1 has been used, and the selections have been saved, this data is used for migration.
3. If the selections have been canceled, processing starts again by reading the ACF data.
4. If option 1 is used again after the selections have been canceled, processing starts again by reading the ACF data.

In the fields below these options, specify the following source and target definitions:

**Dataset & member**
This specifies the name of a partitioned data set together with the member name containing the ACF of the system to be migrated, for example, 'ADMIN1.FOCAL.LIB(ACFZ998)' or 'KEY3.SH.POLICYDB(ACFZ998)'. The member name must be specified. The source data set cannot be the same as the target data set.

**PolicyDB name**
Already specified and cannot be changed.

**Dataset name**
Already specified and cannot be changed.

**Report member**
This specifies the member name where the migration report will be written. The member will be created in the TARGET data set.

**Overwrite System**
This specifies whether the system entry in the policy database will be overwritten during migration if the system entry already exists. For existing system entries you have to specify YES to continue.
Note: In this field, you can prevent SA z/OS from unintentionally overwriting existing information in the policy database for this system. The default is NO. A message, System entry name exists, is displayed if this field is set to NO and you try to use the same system entry name as already exists in the policy database. To proceed, you must either change the system name or set this field to YES.

System Entry Name
This specifies the entry name of the system to be migrated into. Note especially that you also define the system name into which you want to migrate the old system’s ACF fragment.

Note: If the system already exists in the target policy database it will be updated with the migration data, otherwise the new system will be created. To verify that the system already exists just press ENTER.

MVS Sysname
This specifies the MVS system identifier for the migrated system.

Note: If the system already exists in the target policy database the MVS SYSNAME must be the one already defined in the policy database. The special notation with a trailing ‘plus-sign’ is allowed. The first three characters of the SYSNAME must not be AOF because these are reserved for internal processing of the SA z/OS System Operations functions.

Short Description
This specifies the short description of the migrated system.

If you specified that the MIGRATE request should be run in BATCH mode then you need to provide a valid JOB statement in the Job statement information field.

The information that you specify here is saved in your ISPF profile data set and used as the initial values for the next MIGRATE request.

Selecting option 1 Select Migration Objects on the Data Management Select panel leads to the panel shown in Figure 254 on page 334.
On this panel, all entry types are initially selected. You can issue the following actions for the entry types:

- **ADD (A)** to add an entry type to the SELECTED list; if you type A * into the command line, all entry types will be selected.

- **REMOVE (M)** to remove an entry type from the SELECTED list; if you type M * into the command line, all selected entry types will be deselected.

- **EDIT (E)** to edit the selected entry type before migration; you will be presented with the *Entry Name Migration* panel with a list of entries of that type from which you can select the ones you want to migrate or blank out the names of those that you do not want to migrate (see Figure 255 on page 335).

If you enter the command CANCEL, all changes made on this panel and on the *Entry Name Migration* panel (which is displayed if action "EDIT" was selected) are deleted and the migration process starts as if it has been invoked for the first time.
Notes:

1. To migrate ACF fragments of AMC data from releases prior to SA OS/390 2.2, you must select ENT.
2. To migrate an application with product automation data, you must select APL and PRD.
3. To migrate ACF fragments of type SYS from releases prior to SA OS/390 2.2, you must select SYS and NFY.
4. The migration of MTR resources is not supported.

Note: To migrate ACF fragments of AMC data from releases prior to SA OS/390 2.2, you must select ENT. To migrate an application with product automation data, you must select APL and PRD. To migrate ACF fragments of type SYS from releases prior to SA OS/390 2.2, you must select SYS and NFY.

Figure 255. Entry Name Migration Panel

An asterisk (*) in the Dup column indicates that the specified name already exists in the policy database. The existing entry will be replaced during migration.

Remember:

You can only start migration from the Data Management Select panel using either option 2 Start Online Migration or 3 Start Batch Migration.

When you have finished migrating your existing automation control file to the selected policy database, you need to use the BUILD command to build a new automation control file.
### Notes:

1. Migrating an automation control file does not build an SA z/OS automation control file. The information from your old automation control file is migrated into an SA z/OS policy database. To build an SA z/OS automation control file, you must choose Option 3 from the Build Functions Menu.

2. If you have automation control files from multiple systems to migrate, migrate each automation control file into your policy database for each system.

### Entry Name Creation

Table 10 describes how SA z/OS creates names and descriptions for the entry types to be migrated.

#### Table 10. Creation of Entry Names

<table>
<thead>
<tr>
<th>Entry Type</th>
<th>Entry Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENT</td>
<td>From table AOFTENT (ENTNAME)</td>
<td>From table AOFTENT (ENTDECS)</td>
</tr>
<tr>
<td>SYS</td>
<td>From panel AOFGMSYN</td>
<td>From panel AOFGMSYN</td>
</tr>
<tr>
<td>GRP</td>
<td>From ACF line 'GROUP entryname'</td>
<td>From ACF line &quot;Desc : short_description' or '&quot; if the &quot;Desc' line is not available</td>
</tr>
<tr>
<td>SBG</td>
<td>From ACF line 'SUBGROUP entryname'</td>
<td>From ACF line &quot;Desc : short_description' or '&quot; if the &quot;Desc' line is not available</td>
</tr>
<tr>
<td>APG</td>
<td>From ACF line 'APPLGROUP entryname'</td>
<td>From ACF line &quot;Desc : short_description' or '&quot; if the &quot;Desc' line is not available</td>
</tr>
<tr>
<td>APL</td>
<td>From ACF line &quot;* Built : userId yy/mm/dd hh:mm:ss entryname' or from ACF line 'SUBSYSTEM_sysname subsysname' if the &quot;* Built' line is not available. The name created will be: 'subsysname_sysname'.</td>
<td>From ACF line &quot;Desc : short_description' or '&quot; if the &quot;Desc' line is not available</td>
</tr>
<tr>
<td>PRO</td>
<td>From ACF line 'PROCESSOR entryname'</td>
<td>From ACF line &quot;Desc : short_description' or '&quot; if the &quot;Desc' line is not available</td>
</tr>
<tr>
<td>TMR</td>
<td>From ACF line &quot;* Built : userId yy/mm/dd hh:mm:ss entryname' or 'TIMER_sysname' if the &quot;* Built' line is not available</td>
<td>From ACF line &quot;Desc : short_description' or '&quot; if the &quot;Desc' line is not available</td>
</tr>
<tr>
<td>TMO</td>
<td>From ACF line &quot;* Built : userId yy/mm/dd hh:mm:ss entryname' or 'TMO_SET_sysname' if the &quot;* Built' line is not available</td>
<td>From ACF line &quot;Desc : short_description' or '&quot; if the &quot;Desc' line is not available</td>
</tr>
<tr>
<td>TPA</td>
<td>From ACF line &quot;* Built : userId yy/mm/dd hh:mm:ss entryname' or 'TAPE_ATT_sysname' if the &quot;* Built' line is not available</td>
<td>From ACF line &quot;Desc : short_description' or '&quot; if the &quot;Desc' line is not available</td>
</tr>
</tbody>
</table>
Table 10. Creation of Entry Names (continued)

<table>
<thead>
<tr>
<th>Entry Type</th>
<th>Entry Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVC</td>
<td>From ACF line &quot;* Built : userid yy/mm/dd hh:mm:ss entryname' or 'MVS_COMP_sysname' if the 'Build' line is not available</td>
<td>From ACF line &quot;* Desc : short_description' or &quot; if the 'Desc' line is not available</td>
</tr>
<tr>
<td>MDF</td>
<td>From ACF line &quot;* Built : userid yy/mm/dd hh:mm:ss entryname' or 'MVSCOMP_DFT_sysname' if the 'Build' line is not available</td>
<td>From ACF line &quot;* Desc : short_description' or &quot; if the 'Desc' line is not available</td>
</tr>
<tr>
<td>SDF</td>
<td>From ACF line &quot;* Built : userid yy/mm/dd hh:mm:ss entryname' or 'SYSTEM_DFT_sysname' if the 'Build' line is not available</td>
<td>From ACF line &quot;* Desc : short_description' or &quot; if the 'Desc' line is not available</td>
</tr>
<tr>
<td>ADF</td>
<td>From ACF line &quot;* Built : userid yy/mm/dd hh:mm:ss entryname' or 'APPL_DFT_sysname' if the 'Build' line is not available</td>
<td>From ACF line &quot;* Desc : short_description' or &quot; if the 'Desc' line is not available</td>
</tr>
<tr>
<td>AOP</td>
<td>From ACF line &quot;* Built : userid yy/mm/dd hh:mm:ss entryname' or 'AUTOOPS_sysname' if the 'Build' line is not available</td>
<td>From ACF line &quot;* Desc : short_description' or &quot; if the 'Desc' line is not available</td>
</tr>
<tr>
<td>NFY</td>
<td>From ACF line &quot;* Built : userid yy/mm/dd hh:mm:ss entryname' or 'NOTIFYOP_sysname' if the 'Build' line is not available</td>
<td>From ACF line &quot;* Desc : short_description' or &quot; if the 'Desc' line is not available</td>
</tr>
<tr>
<td>NTW</td>
<td>From ACF line &quot;* Built : userid yy/mm/dd hh:mm:ss entryname' or 'NETWORK_sysname' if the 'Build' line is not available</td>
<td>From ACF line &quot;* Desc : short_description' or &quot; if the 'Desc' line is not available</td>
</tr>
<tr>
<td>NNT</td>
<td>From ACF line &quot;* Built : userid yy/mm/dd hh:mm:ss entryname' or 'NNT_SESSION_sysname' if the 'Build' line is not available</td>
<td>From ACF line &quot;* Desc : short_description' or &quot; if the 'Desc' line is not available</td>
</tr>
<tr>
<td>RES</td>
<td>From ACF line &quot;* Built : userid yy/mm/dd hh:mm:ss entryname' or 'RES_CLISTS_sysname' if the 'Build' line is not available</td>
<td>From ACF line &quot;* Desc : short_description' or &quot; if the 'Desc' line is not available</td>
</tr>
<tr>
<td>SCR</td>
<td>From ACF line &quot;* Built : userid yy/mm/dd hh:mm:ss entryname' or 'STAT_DETAIL_sysname' if the 'Build' line is not available</td>
<td>From ACF line &quot;* Desc : short_description' or &quot; if the 'Desc' line is not available</td>
</tr>
<tr>
<td>EVT</td>
<td>From ACF line '@EXTCOND entryname'</td>
<td>From ACF line &quot;* Desc : short_description' or &quot; if the 'Desc' line is not available</td>
</tr>
<tr>
<td>SVP</td>
<td>From ACF line '@SERVICE entryname'</td>
<td>From ACF line &quot;* Desc : short_description' or &quot; if the 'Desc' line is not available</td>
</tr>
<tr>
<td>TRG</td>
<td>From ACF line '@TRIGGER entryname'</td>
<td>From ACF line &quot;* Desc : short_description' or &quot; if the 'Desc' line is not available</td>
</tr>
<tr>
<td>PRD</td>
<td>not applicable</td>
<td>not applicable</td>
</tr>
</tbody>
</table>
The MIGRATE function checks for the comment line starting with * Built, and takes the entry name from there. Such a line is created automatically, if the ACF is built using the customization dialog.

If such a comment line is not available, a fixed entry name is created as described in Table 10 on page 336 and all definitions are stored into this entry. In a manually edited ACF, it may be helpful to insert comment lines with entry names as shown above to group the definitions accordingly, so that the migration is able to store the definitions into the policy database with their meaningful names.

**Restrictions:** If an application or an application group entry name is migrated into the policy database and the same entry name already exists, only the existing subsystem/automation name will be used during the migration process. The subsystem/automation name from the input will be ignored. The reason for this is that these names are used for creating resource names and when the subsystem name or automation name has changed, the resource name would be invalidated and the automation manager data would be incorrect afterwards.

**Notes:**

1. Definitions relevant for subsystems need to be migrated together with the subsystem’s ACF data in one step in order to have them migrated to the correct application. If this is not observed, the definitions will be migrated to User E-T Pairs.

2. If migrating Application fragments, an application group is automatically created. This is required to guarantee that the application is linked to the appropriate system. All applications found in the migration ACF-source are linked to this application group. The entry name of this application group is APG_sysname.

3. PARENT definitions in application fragments are converted to either HASPARENT or HASPASSIVEPARENT relationships. The type of relationship created is determined by the value specified in the APL Relationship Type field in the policy database Converts Options panel (see "Initial Policy Database Conversion" on page 355). The default value is HASPARENT.

4. If no description has been given in the entry name migration panel (AOFGMENM), the description found in the ACF fragment (keyword SDESC or DESC) will be used.
Migration of Include Files and Product Automation Definitions

Note: 
/Product Automation stands for automation of the DB2, CICS, IMS and OPC products, formerly referred to as Features.

With the introduction of the automation manager, it is required to have all configuration data in one place. Any data added via a user include file that is maintained outside of the customization dialog will not be available for SA z/OS Version 2 releases. Therefore, it is mandatory to migrate these members.

If the includes contain definitions that are available as entry types within the customization dialog, they are migrated into the corresponding entry types of the current SA z/OS release. If the definitions are not known to the current SA z/OS release, they will be migrated to entries of User E-T Pairs (UET).

As an example, the following definitions using non-ACF keywords are assumed to be part of the source ACF:

* Desc : User E-T Pairs HSM
* Built : OPER1 99/08/11 14:31:23 HSM
* HSMMAIL BCDS,
  MAILLIMIT=80%,
  MAILOP=(I-SC.DATABASESYSTEM),
  MAILTXT=(BACKUP CONTROL DATASET OF HSM),
  MAILTXT=(IS 80% FULL),
  LIMIT=97%

The MIGRATE function checks for the comment line starting with * Built, and takes the entry name from there. Such a line is created automatically, if the ACF is built using the customization dialog. Each User E-T Pairs (UET) entry consists of one or more entry-type pairs each of which consists of one or more keyword-data pairs. In this case, HSMMAIL BCDS is treated as an entry-type pair, which includes all the following keyword-data pairs. After migration, the customization dialog shows the definitions in the following way:

<table>
<thead>
<tr>
<th>Action</th>
<th>Keyword/Data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAILLIMIT</td>
</tr>
<tr>
<td></td>
<td>80%</td>
</tr>
<tr>
<td></td>
<td>MAILOP</td>
</tr>
<tr>
<td></td>
<td>(I-SC.DATABASESYSTEM)</td>
</tr>
<tr>
<td></td>
<td>MAILTXT</td>
</tr>
<tr>
<td></td>
<td>(BACKUP CONTROL DATASET OF HSM)</td>
</tr>
<tr>
<td></td>
<td>MAILTXT</td>
</tr>
<tr>
<td></td>
<td>(IS 80% FULL)</td>
</tr>
<tr>
<td></td>
<td>LIMIT</td>
</tr>
<tr>
<td></td>
<td>97%</td>
</tr>
</tbody>
</table>

Figure 256. UET Keyword-Data Specification for Entry Name HSMMAIL
Until now, definitions for product automation for CICS, IMS, and OPC were contained in separate members, that were included in the ACF either via entry type ICL (Includes), or via manually added include statements. As mentioned above, Includes via the Dialog will no longer be supported, and manually added include members will be rejected by the ACF load, so they also have to be migrated into the policy database.

With SA OS/390 2.1, the customization dialog has been extended to allow specification of all the ACF entries required for product automation. Also the migration will automatically move these definitions to their appropriate entry types. Refer to the relevant product-specific reference manuals for further details of these new entry types.

As a specific example of product automation definitions, the following CICS definitions are assumed to be part of the source ACF:

```plaintext
* SUBSYSTEM SPECIFICATION
SUBSYSTEM CICS4A,
   PARENT=(CICS4),
   SHUTDLY=00:03:00,
   STRTDLY=00:03:00,
   JOB=CICS4A,
   SDESC='TEST-CICS'
*
* DEFINES CICS
CICSNTL CICS4A,
   APPLID=IPUAJCIA,
   MAJNODE=IPUJBC,
   AUTOOPS=CICSTOP01,
   KEYPOINTREQ=YES,
   SMFID=215
*
* HEALTH CHECKING
HEALTHCK CICS4A,
   FUNCTION=(0,EVECHLTH,00:10:00,30,AUTO,'SAMPLE COBOL HEALTH')
*
* LINK MONITORING
CONNECT CICS4A.IM4A,
   DESC='IMS521 (IPUAHI51) on AOC2',
   DAY=(WEEKDAY,0830-1500),
   DAY=(WEEKEND,0900-1800),
   TYPE=IMS,
   ECHO=(x,60),
   ACQDELAY=15,
   RELDELAY=15,
   MAXREPAIR=5,
   CRITICAL=YES
*
* SERVICE PERIODS
SERVICE CICS4A,
   DAY=(DAILY,0815-1715),
   DAY=(WEEKEND,DOWN-DOWN)
*
* TRIGGER DEFINITION
TRIGGER CICS4A,
   STARTUP=(SERVICE)
*
* MINOR RESOURCE THRESHOLDS
THRESHOLDS CICS4A.TRAN,
   CRIT=(2,00:05),
   FREQ=(4,01:00),
   INFRA=(5,24:00)
*
```

Figure 257. ACF Definitions for CICS

All entries (except SERVICE and TRIGGER) will be migrated into an application definition. If the target policy database does not yet contain an application with
that name, the migration can be run without problems. If however the target policy database has already an application with that name, there are three possibilities:

1. The conflict can be resolved by changing the name of the source application on the Application Entryname Migration panel before the migration is started. This panel shows that an entry with the same name already exists, and allows you to change that name.

2. If the migration is run with the duplicate name, then the entry in the target PolicyDB is overwritten with the definitions from the ACF keywords.

3. As mentioned before, you can specify an entry name by inserting a * Built line in the source before migrating the ACF.

Keywords SERVICE and TRIGGER will be migrated into service period and trigger definitions of the current SA z/OS release.

The following screens show how the ACF entries from Figure 257 on page 340 will be migrated into the customization dialog.

**SUBSYSTEM SPECIFICATION**

```
SUBSYSTEM CICS4A,
    PARENT=(CICS4),
    SHUTDLY=00:03:00,
    STRTDLY=00:03:00,
    JOB=CICS4,
    SDESC='TEST-CICS'
```

After migration, you can find the subsystem specification from the automation control file under policy item APPLICATION INFO of entry type APL with entry name CICS4A_sysname. sysname is the MVS sysname chosen for the migration.

**CICS DEFINITION**

```
CICSCNTL CICS4A,
    APLID=IPUAJCIA,
    MAJNODE=IPUJBC,
    AUTOOPS=CICSOP01,
    KEYPOINTREQ=YES,
    SMFID=215
```

The CICSCNTL entry will be migrated to entry type APL (application type CICS) with entry name CICS4A_sysname. Selecting policy item CICS CONTROL from this application will present the following panel:
HEALTH CHECKING

HEALTHCHK CICS4A,
   FUNCTION=(0,EVECHLTH,00:10:00,30,AUTO,'SAMPLE COBOL HEALTH')

The HEALTH CHECKING entry will be migrated to entry type APL (application type CICS) with entry name CICS4A_sysname. Selecting policy item MESSAGES/USER DATA from this application will present the following panel:

Selecting message HEALTHCHK with action USER will present the following panel:
LINK MONITORING

CONNECTION CICS4A.IM4A,
DESC='IMS521 (IPUAHI51) on AOC2',
DAY=(WEEKDAY,0830-1500),
DAY=(WEEKEND,0900-1800),
TYPE=IMS,
ECHO=(X,60),
ACQDELAY=15,
RELDELAY=15,
MAXREPAIR=5,
CRITICAL=YES

The LINK MONITORING entry from the ACF will be migrated to entry type CCN (CICS link definition) under the name CICS4AIM4A. Note that the period (.) has been removed from the name found in the ACF. Selecting policy item CICS CONNECTION from this CICS link, you will be presented with a panel like the following one:

The monitoring period information from the ACF is migrated to entry type CVP under the name in our example, CICS4AIM4A.

Selecting its policy item MONITORING PERIOD will display the panel shown in Figure 258 on page 344.
The migration also links the generated monitoring period (see Figure 258) to the migrated CICS connection CICS4AIM4A. You can view this link by selecting policy item MONITORING PERIOD from this CICS link CICS4AIM4A. You will be presented with a panel like the following one:

**Figure 258. Monitoring Service Period**

SA z/OS will also link the application (APL) CICS4A_sysname and the CICS connection (CCN) CICS4AIM4A. You can view the result by selecting policy item CICS CONNECTION from application CICS4A_sysname. This will present a panel similar to the following:

**MINOR RESOURCE THRESHOLDS**

THRESHOLDS CICS4A.TRAN,

CRIT=(2,00:05),
FREQ=(4,01:00),
INFR=(5,24:00)
The THRESHOLDS entry will be migrated to entry type APL to the application with entry name CICS4A_sysname (application type CICS). Selecting policy item RESOURCE THRESHOLDS from this application and then selecting the CICS resource threshold TRAN that you want to see will present the following panel:

```
COMMANDS HELP

Thresholds Definition

Command ==> Thresholds Definition

Entry Type : Application
Entry Name : CICS4A_sysname
PolicyDB Name : DATABASE_NAME
Enterprise Name : YOUR_ENTERPRISE
Resource: CICS4A.TRAN
Description:

Specify the number of times an event must occur to define a particular level.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Critical Number</th>
<th>Critical Interval (hh:mm)</th>
<th>Frequent Number</th>
<th>Frequent Interval (hh:mm)</th>
<th>Infrequent Number</th>
<th>Infrequent Interval (hh:mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICSK4D.TRAN</td>
<td>02</td>
<td>00:05</td>
<td>04</td>
<td>01:00</td>
<td>05</td>
<td>24:00</td>
</tr>
</tbody>
</table>
```

**SERVICE PERIODS**

SERVICE CICS4A,
  DAY=(DAILY,0815-1715),
  DAY=(WEEKEND,DOWN-DOWN)

You can find this extract from the ACF under policy item SERVICE WINDOW of the service period with the entry name CICS4A (entry type SVP). The link between the application (APL) and the SVP can be found under policy item SERVICE PERIOD of entry type APL with entry name CICS4A_sysname.

**TRIGGER DEFINITION**

TRIGGER CICS4A,
  STARTUP=(SERVICE)

You can find this extract from the ACF under policy item CONDITION of the trigger with the entry name CICS4A (entry type TRG). The link between the application (APL) and the TRG can be found under policy item TRIGGER of entry type APL with entry name CICS4A_sysname.

**Generating a Job for Batch Migrate**

You can run a job for batch migration that performs the same processing as the Migration function of the customization dialog. The SINGSAMP sample library contains a sample job for this task in member INGEBMIG. Before submitting this job, you need to edit the jobcard appropriately as described in the sample itself.
Chapter 10. Automating Networks

Automation Network Definition Process 347
Defining an SDF Focal Point System 348
Defining Gateway Sessions 349
Defining an Outbound Gateway Autotask 349
Defining Automatically-Initiated TAF Full-Screen Sessions 350

Automation Network Definition Process

This section summarizes the steps for defining an automation network to SA z/OS. More detail for each step of the process is provided later in this chapter.

1. Define your message forwarding paths between different systems. To do this, you define:
   - A primary focal point, where all notifications are sent.
   - An optional backup focal point, used when the primary focal point is unavailable.
   - Target systems, which are monitored and controlled by the focal point system.
   - Gateway sessions between the systems. “Defining Gateway Sessions” on page 349 describes how to define gateway sessions.

2. Modify the NetView definitions to reflect your automation network configuration. The chapter on how to install SA z/OS on host systems in [System Automation for z/OS Planning and Installation] provides details.

For an example of the automation network definition process, also refer to the chapter about installing SA z/OS on host systems in [System Automation for z/OS Planning and Installation].

These definitions create a path allowing message forwarding from target systems to the focal point system.

A message forwarding path is best implemented by defining systems in the following top-down manner:
1. Primary focal point system
2. Backup focal point system
3. Target systems

Defining the primary focal point first ensures that it is ready to handle forwarded messages as soon as forwarding is turned on for the target systems.

If the message forwarding path is not yet implemented on all systems in an automation network, messages are displayed to notification operators on the target systems. Once the message forwarding path is implemented, notifications are forwarded to the focal point system.

If the target systems are implemented first, additional overhead occurs because the target systems unsuccessfully attempt to forward notifications, and the notifications are logged in the NetView log.
Defining an SDF Focal Point System

The focal point system and backup focal point systems are defined using the Network entry type in the customization dialog. Each system has a single entry in the automation policy defining the next system or domain in the message forwarding path. Figure 259 shows an example automation network. In this example, the primary focal point system is CHI01. The backup focal point is CHI02.

![Diagram of automation network showing CHI01 as the primary focal point and CHI02 as the backup focal point, with ATL01 and ATL02 as target systems.]

Figure 259. Focal Point Forwarding Definitions for Systems

The corresponding focal-point forwarding definitions in the automation policy for systems CHI02, ATL01 and ATL02 are as follows. You reach the required panel by selecting the FORWARD policy item of a Network entry type.

- CHI02 automation policy:

```
Command: AOPFINE2 Notification Forwarding
Entry Type: Network PolicyDB Name: DATABASE_NAME
Entry Name: FOCAL_NETWORK Enterprise Name: YOUR_ENTERPRISE

Enter the NetView domains for automation notification forwarding.
Primary Domain => CHI01 Current Domain ID
Backup Domain => &DOMAIN. Backup Domain ID
```

Figure 260. Notification Forwarding Panel for CHI02

- ATL01 and ATL02 automation policy:

```
Command: AOPFINE2 Notification Forwarding
Entry Type: Network PolicyDB Name: DATABASE_NAME
Entry Name: FOCAL_NETWORK Enterprise Name: YOUR_ENTERPRISE

Enter the NetView domains for automation notification forwarding.
Primary Domain => CHI01 Current Domain ID
Backup Domain => CHI02 Backup Domain ID
```

Figure 261. Notification Forwarding Panel for ATL01 and ATL02
In Figure 259 on page 348, the automation policies for system CHI01 do not have any forward focal point definitions. This is because SA z/OS considers the current system as the primary focal point and displays messages without forwarding them if either or both of the following is true:

- The forward focal point is not defined in the automation policy for the system.
- The system specified in a forward focal point definition is the current system.

**Defining Gateway Sessions**

To define gateway sessions:

1. For each system, define the outbound gateway autotask (GATOPER) on the Auto Operators policy object of the customization dialog. See "Defining an Outbound Gateway Autotask" for details.

2. On the SA z/OS Network policy object, use the GATEWAY policy item to define the destination systems to which the originating system connects.

3. Define operator IDs used for all inbound and outbound gateway autotasks used on the system in the NetView DSIPARM data set member DSIOPF. See the chapter on how to install SA z/OS on host systems in System Automation for z/OS Planning and Installation for details.

**Defining an Outbound Gateway Autotask**

In any system, only the outbound gateway task is defined using the Auto Operators entry type.

If GATOPER has not previously been defined, type the automation operator name, gatoper, in the Automated Function field of the Auto Operator Definition policy item, as shown in Figure 262.

![Figure 262. Automation Operator Definitions Panel](image)

You do not have to specify any messages for GATOPER. When you press Enter, the Automation Operator NetView Userids panel is displayed automatically, as shown in Figure 263 on page 350.

If GATOPER has previously been defined, then select it by entering an S in the Action column.
Enter the NetView operator ID that is associated with the GATOPER function on the Automation Operator NetView Userids panel.

**Note:**
This NetView operator ID must be unique within the enterprise.

For example, to define the outbound gateway autotask for system CHI01 in the automation network shown in System Automation for z/OS Planning and Installation, the values shown in Figure 263 are specified.

![Figure 263. Automation Operator NetView Userids Panel](image)

**Defining Automatically-Initiated TAF Full-Screen Sessions**

You can automatically establish Terminal Access Facility (TAF) full-screen sessions for applications that SA z/OS monitors, so that the operators need not define the sessions on a daily basis.

These TAF full-screen sessions are defined in the FULL SESSIONS policy item for a Network policy object.

In addition to defining TAF full-screen sessions using the customization dialog, you follow the NetView process for customizing TAF full-screen sessions, as outlined in Tivoli NetView for z/OS Customization Guide.

Once TAF full-screen sessions are set up, they can be managed using the TAF Fullscreen Menu in the SA z/OS Operator Interface. See System Automation for z/OS User’s Guide for more information on managing TAF full-screen sessions.

To define an application on the Fullscreen TAF Application Definition panel that you reach by selecting the FULL SESSIONS policy item of a Network policy object, specify the following:

- The session name, or the name of the application for which a TAF full-screen session is to be established, for example, TSO. This name is displayed in the Description field on the TAF Fullscreen Menu operator panel. This value can be the same as that used for the application ID.
Defining Automatically-Initiated TAF Full-Screen Sessions

- The application ID. You can obtain this value from the library containing the network definitions (VTAMLST) or from your network system programmer.
- The system name on which the application runs, for example, CHI01. This is an information-only field and is displayed in the System field on the TAF Fullscreen Menu operator panel.

For example, the following panel defines a TAF full-screen session for TSO in system CHI01:

```
COMMANDS  HELP
---------------------------------------------------------------
AOFPINE3   Fullscreen TAF Application Definition Row 1 to 10 of 20
Command ===>
SCROLL====> PAGE

Entry Type: Network  PolicyDB Name: DATABASE_NAME
Entry Name: FOCAL_NETWORK  Enterprise Name: YOUR_ENTERPRISE

Enter the applications with which SA z/OS operators can establish TAF sessions automatically using the operator interface.

<table>
<thead>
<tr>
<th>Session Name</th>
<th>Application ID</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSO</td>
<td>TAIN1</td>
<td>CHI01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 264. Fullscreen TAF Application Definition Panel
Chapter 11. How System Operations Coordinates with Automatic Restart Manager

SA z/OS system operations provides coordination with the Automatic Restart Manager. The Automatic Restart Manager (ARM) is a base OS/390 component. It is a recovery function that automatically restarts designated applications when:

• The application abends.
• The system on which the application is running is part of a sysplex, and that system fails. In this case, ARM will attempt to restart the application on another system within the sysplex.

SA z/OS coordinates with ARM to:

• Determine which facility is responsible for restarting a specific application.
• Avoid possible duplications or conflicts in application recovery attempts.
• Allow you to take full advantage of SA z/OS fallback capabilities for applications running on sysplexes. SA z/OS continues to automate an application after it has been moved to a fallback system, provided SA z/OS is installed on that system. If it is not installed on the fallback system, SA z/OS is still aware that the application is active on a system other than its primary one and does not attempt to restart it.

You have to define the Automatic Restart Manager policy using the administrative data utility for ARM policy data (IXCMIAPU) described in OS/390 MVS Setting Up a Sysplex.

SA z/OS resolves Automatic Restart Manager statuses to SA z/OS statuses, incorporates Automatic Restart Manager-related conditions, and provides one status related to Automatic Restart Manager:

• EXTSTART - The application is being started or restarted externally.

Defining an ARM Element Name

Automatic Restart Manager uses element names to identify the applications with which it works. Each Automatic Restart Manager enabled application must have a unique element name for itself that it uses in all communication with Automatic Restart Manager. Automatic Restart Manager tracks the element name and has its policy defined in terms of element names. If an application moves between systems it MUST continue to use the same element name as it did on the original system. For more information on defining Automatic Restart Manager names to SA z/OS, see "Application Policy Object" on page 145.

All Automatic Restart Manager elements are unregistered initially. Transitions between statuses are caused by:

• IXCARM macro invocations
• Application failures
• System failures
• Timeouts

A minor resource definition subsystem:0ARM can be used to tailor automation behaviour during ARM restart processing. As an example, a subsystem:0ARM minor resource could be specified with a RESTART EXIT enabled to drive a user supplied
Defining an ARM Element Name

exit during ARM restart. The user exit would control additional actions to be taken during ARM restart of the subsystem. If the RESTART flag for this minor resource is resolved to ‘N’, SA z/OS will not allow ARM to attempt a restart of the application.

Rather than use the subsystem.0ARM minor resource definition, a RESTART EXIT could also be specified against the major resource definition for the application. In this case the exit would be driven for all application restarts, not just ARM.

Defining a MOVE Group for Automatic Restart Manager

All resources with the same ARM element name should be linked to one Sysplex Application Group of nature MOVE (MOVE group).

An application’s ARM element name is defined either during creation on the Define New Entry panel for applications or after creation via policy item APPLICATION INFO, in both cases in the field MVS Automatic Restart Management Element Name.

In order to ensure an application in a MOVE group has completely deregistered from ARM before the automation manager attempts to restart it, a Prepareavailable/WhenObservedDown (passive) relationship must be defined for each ARMed application in the MOVE group with the MOVE group defined as the supporting resource.

To make sure that the Automation Manager will start the applications linked to a MOVE group, the applications should not be in a CTLDOWN status. Options start on IPL and start on RECYCLE should not be set to NO.

For more information on how to define MOVE groups refer to "Creating a New ApplicationGroup” on page 125.
Chapter 12. Conversion Function

This chapter provides information on what you should be aware of before using the new release of SA z/OS. In detail, the following topics are discussed:

- “Moved and Restructured Data”
- “Initial Policy Database Conversion”

The conversion function creates a report named $RPTCNV as a member in the data set of the policy database. All messages are logged.

Moved and Restructured Data

The following changes have been made:

- JES3 MONITOR for SA z/OS has been replaced by the new entry type MTR (MONITOR). Automatic conversion is not done during the initial conversion.
- For entry type APL, the Jobtype and Transient Rerun fields have been moved from policy item AUTOMATION INFO to APPLICATION INFO.
- For entry type SYS the Automation Symbol definitions have been moved from policy item SYSTEM INFO to AUTOMATION SYMBOLS.
- The selection value ALWAYS is no longer a special keyword for:
  - commands and replies in policy item MESSAGES/USER DATA
  - commands in policy item STARTUP
- It is set to blank during initial conversion.

Initial Policy Database Conversion

During the very first opening of a policy database with the customization dialog, a conversion to the new layout is done automatically. There are special conversion options available by selecting 0 Settings on the Customization Dialog Primary Menu, and then 4 CONVERT on the Customization Dialog Settings Menu. This displays the PDB Convert Option panel, as shown in Figure 265 on page 356. This panel is used to specify the options for the PDB Conversion. These options will be saved in your ISPF profile pool and remain in effect until set again.

On this panel, SA z/OS allows you to request the generation of an automation name for application groups during the very first conversion of a policy database. If an application group has an automation name, SA z/OS can generate resources from such entries.
Initial Policy Database Conversion

Figure 265. PDB Convert Option Panel

This panel contains the following fields:

Set message type

Specify whether the message type should be set to Ignore for AT build during the initial policy database conversion, which is the very first conversion done during the first opening of the policy database:

YES

This indicates that the AUTO action that is used to define status message automation is set to IGNORE for those message IDs that are defined in the policy database and where no AT entry is forced by SA z/OS. No AT entry is then created for these messages during AT build. This allows for the gradual exploitation of the SA z/OS-generated AT for individually selected messages.

NO

This indicates that all message definitions remain unchanged. The messages will be used to build AT entries when running the Configuration Build process.

Note:

This option takes effect for the initial policy database conversion only. Subsequent conversions that might be triggered due to the usage of the Migrate function do not honor this option.

APG Automation Name

Specify whether an automation name should be set up during the initial policy database conversion. Select one of the following values:

YES

Indicates that the automation name for each APG entry in the policy database will be set with the entry name of the APG entry, if it is less than or equal to 11 characters. If the entry name is more than 11 characters long, the automation name will be forced to blank.

NO

Indicates that the automation name for each APG entry in the PDB will be filled with blank.
Notes:

1. This conversion option should be set to YES. Otherwise SA z/OS cannot generate resources for these entries. The default value is NO.
2. This option takes effect for the initial policy database conversion only. Subsequent conversions that might be triggered due to the usage of the Migrate function do not honor this option.
3. This applies only for conversion from SA OS/390 1.3.

APL Relationship Type

Specify the relationship type for relationships created during initial policy database conversion, which is the very first conversion done during the first opening of the policy database, and also for new relationships that are created by use of the MIGRATE function. Select one of the following:

HASPARENT
Indicates that relationships that are created will be set up as HASPARENT relationships. The PARENT/CHILD relationships from earlier releases of SA z/OS will be converted to HASPARENT relationships during the conversion process. A start request will be issued against the supporting resource (PARENT) if it is down, and a start request is issued against the supported resource (CHILD).

HASPASSIVEPARENT
Indicates that relationships that are created will be set up as HASPASSIVEPARENT relationships. The PARENT/CHILD relationships from earlier releases of SA z/OS will be converted to HASPASSIVEPARENT relationships during the conversion process. A start request will not be issued against the supporting resource (PARENT) if it is down, and a start request is issued against the supported resource (CHILD).

The default value is HASPARENT.

Notes:

1. The APL Relationship Type option takes effect for both the initial PDB conversion and any subsequent conversions that might be triggered by the use of the Migrate function.
2. This option applies only to conversion from SA OS/390 1.3.

Converted Data

During initial conversion the following data will be converted:

- The PARENTS of all applications will be converted to either HASPARENT or HASPASSIVEPARENT relationships.
  Any instances that have inherited their startcmds or shutoptions parent relationships from a class will be assigned the new HASPARENT StartsMe/StopsMe relationships during the conversion process.
- The STARTUP definitions of all applications will be converted to EXTERNAL STARTUP definitions.
The SHUTOPTIONS definitions of all applications will be converted to EXTERNAL SHUTDOWN definitions. Additionally, if a class-instance pair is converted from SA OS/390 1.3 where the instance originally inherited its SHUTOPTIONS from the class, the following translation will be performed during the ACF build process:

If a HASPARENT, or HASPASSIVEPARENT relationship of either StopsMe or StartsMeAndStopsMe exists for an instance and a similar relationship of either StopsMe or StartsMeAndStopsMe exists for a linked class, then the ACF build output will contain a SHUTOPTIONS value for the instance. The SHUTOPTIONS value for the instance will be taken from the HASPARENT or HASPASSIVEPARENT relationship of the instance and not the class.

This is depicted in the following example:

**From SA OS/390 1.3:**

```
SUBSYSTEM_CLASS SYS_CLASS,
    PARENT=CLS_PARENT,
    SHUTOPTIONS=(PARENT)

SUBSYSTEM SYS_INST
    PARENT=INST_PARENT,
    OBJECTCLASS=SYS_CLASS
```

**To SA z/OS 2.3:**

In the dialogs, the following relationships are generated during the conversion process:

For APL class SYS_CLASS

1  HASPARENT  CLS_PARENT/APL/= (StopsMe)

For APL SYS_INST with Class SYS_CLASS

1  HASPARENT  INST_PARENT/APL/= (Stopsme)

These entries will be built as:

```
SUBSYSTEM_CLASS SYS_CLASS,
    PARENT=CLS_PARENT,
    SHUTOPTIONS=(CLS_PARENT)

SUBSYSTEM SYS_INST,
    PARENT=INST_PARENT,
    OBJECTCLASS=SYS_CLASS,
    SHUTOPTIONS=(INST_PARENT)
```

**Example:**

Imagine the following definitions for entry type Application from SA OS/390 1.3 on the Application Automation Definition panel:

Parents ... SUBSYS1, SUBSYS2, SUBSYS3
Startup ... PARENT
Shutoptions ... SUBSYS2, SUBSYS3

These will be converted to the following definitions:

- on the same Application Automation Definition panel:
  
  External Startup ... ALWAYS
  External Shutdown ... ALWAYS

- the following relationships will be created and will be visible on the Relationship Selection List panel:
• For all applications which are linked to systems, resources will be created
• If the convert option is set to ‘YES’ then for all application groups which have
an entryname of less than 11 characters, an automation name will be generated.
• For all application groups which are linked to systems or sysplex and an
‘automation name’ is given, resources are created
• For all application groups where a resource has been created, a
resource-to-resource link will be created for each application which is linked to
the application group.
• The commands specified under policy item RESTART for Applications will be
converted to commands with message ID ACORESTART under policy item
MESSAGES/USER DATA.
• Service period windows are converted to the new syntax in the following way:
  – As old service periods were the schedules where an application was expected
to be active, they are converted to new UP service periods.
  – DOWN-DOWN service periods are converted to the new DOWN service
periods from 00:00 - 24:00;
  – Service periods in **** notation are converted to 00:00 if they are coded as
start time of the first service period;
  – Service periods in **** notation are converted to 24:00 if they are coded as
stop time of the last service period;
• Service periods linked to triggers are converted to direct links to the application.
• User E-T pairs which refer to an existing subsystem name will be converted to
the policy item MESSAGES/USER DATA. Depending on the keyword the data
will be converted to the CMD, REPLY, CODE or USER part of the message.
• The policy items MVSDUMP AUTOMATION, SMFDUMP AUTOMATION,
LOGREC AUTOMATION and SYSLOG AUTOMATION of entry type MVC do
no longer exist. The data is converted to policy item MINOR RESOURCE
FLAGS which is introduced for entry type MVC.
• The message which could be specified under policy ‘THRESHOLD’ for
MVSDUMP, SMFDUMP, LOGREC and SYSLOG has been converted to policy
‘MESSAGES/USER DATA’.
• The policy item ‘MINOR RESOURCE FLAGS’ of entry type SYS no longer exists.
Minor resources of system (SYS) entries are converted to minor resources of
MVS Components (MVC). To ensure this occurs, a new MVC entry is created
with the following name structure \textit{MVS\_COMPn\_sysid} where ‘n’ might be
omitted or a number between 1 and 99 to ensure that a new entry is created and
an existing one is not overwritten. Additionally a link is created between the
newly created MVC entry and the SYS entry where the minor resources came
from.
Initial Policy Database Conversion

- Resync of ACF prefix value for each entryname in the PDB with table AOFTETP (formally done during each ACF build).
- Since the introduction of SA OS/390 2.1 it is no longer possible to link different applications with the same subsystem name to the same system. Those links are deleted by the conversion by deleting the link between the application and the application group.
- All feature definitions are converted to product automation entries of the customization dialog.
- All PC-related processor data is deleted.
- All PC-related communication path data is deleted.
- All communication task data is deleted.
- Remaining processor and communication path data is converted to the new layout.
- Notify operators defined in entry type SYS are converted to the new separate entry type NFY.
- Auto message classes defined in the separate entry type AMC are converted to the entry type ENT.
- All systems of type TPF are deleted.
- Application data of IMS applications is converted to the new layout.
- User E-T pairs with the following entry type pairs are converted to the entry types PRO, APL and GRP:
  - PROCESSOR proname
  - IMAGECNTL subsysname
  - ENVIRON TEMP@SYSPLEX
- The SAVE definitions of timers are converted to the new layout.

Cross Reference of SA OS/390 1.3 Feature Keywords to SA z/OS 2.3 Customization Dialog Entries

In SA OS/390 1.3 you had two ways to define CICS, IMS or OPC keywords to SA OS/390 automation: you either edited the ACF directly or entered the keywords into the User E-T Pairs via the customization dialog and had SA OS/390 build the ACF.

These definitions are automatically converted to SA z/OS 2.3 Customization Dialog entries.

The feature keyword conversions either take place by using the Conversion function, when a policy database in SA OS/390 1.3 format is opened or by using the Migration functions, when an ACF is migrated (and type PRD is selected). Table 11 on page 361 contains details of all the entries that are migrated or converted for product automation.

The following terms are used in Table 11 on page 361

Format change
This means the data entry format is slightly modified from the original ACF entries, refer to the Product Automation dialog documentation.

New
This means that a new dialog function has been implemented for this definition.

Obsolete
This means that the original entry type no longer exists.
Table 11. Cross Reference of SA OS/390 1.3 Feature Keywords to SA z/OS 2.3 Dialog Entries

<table>
<thead>
<tr>
<th>Keyword</th>
<th>SA z/OS 2.3 Migration/Conversion Target</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CICS Keywords</strong>, where Application Messages is stated, this refers to the MESSAGES/USER DATA policy item of the appropriate CICS subsystem definition.</td>
<td>Application Messages</td>
</tr>
<tr>
<td>ACORESTART</td>
<td>Application Messages</td>
</tr>
<tr>
<td>Automation Flags</td>
<td>Application Automation info</td>
</tr>
<tr>
<td>AUTOOPS</td>
<td>Auto Operators (not in CICSCNTL any more)</td>
</tr>
<tr>
<td>SDF</td>
<td>SDF</td>
</tr>
<tr>
<td>ENVIRON SETUP CICS</td>
<td>Obsolete</td>
</tr>
<tr>
<td>ENVIRON TIMEOUT</td>
<td>TIMEOUT</td>
</tr>
<tr>
<td>SHUTDOWN</td>
<td>Application</td>
</tr>
<tr>
<td>STARTUP</td>
<td>Application</td>
</tr>
<tr>
<td>SUBSYSTEM</td>
<td>Application (Application type = CICS)</td>
</tr>
<tr>
<td>THRESHOLDS</td>
<td>New Application policy item (CICS) RESOURCE THRESHOLDS</td>
</tr>
<tr>
<td>ABCODESYSTM</td>
<td>Application Messages</td>
</tr>
<tr>
<td>ABCODETRAN</td>
<td>Format change, Application Messages</td>
</tr>
<tr>
<td>AREA</td>
<td>Obsolete, new CICS State/Action, link to APL</td>
</tr>
<tr>
<td>CICSCNTL</td>
<td>New Application policy item CICS CONTROL</td>
</tr>
<tr>
<td>CICSGROUP</td>
<td>Obsolete, handled in base code</td>
</tr>
<tr>
<td>CONNECTION</td>
<td>New define connection, link to APL</td>
</tr>
<tr>
<td>EXTCOND</td>
<td>Migrated to entry type <em>Events</em></td>
</tr>
<tr>
<td>HEALTHCHK</td>
<td>Format change, Application Messages</td>
</tr>
<tr>
<td>LISTSHUT</td>
<td>Format change, Application Messages</td>
</tr>
<tr>
<td>PRODUCT</td>
<td>Obsolete, new CICS State/Action, link to APL</td>
</tr>
<tr>
<td>RCVRSOS</td>
<td>Format change, Application Messages</td>
</tr>
<tr>
<td>RCVRTRAN</td>
<td>Format change, Application Messages</td>
</tr>
<tr>
<td>SERVICE</td>
<td>Migrated to entry type <em>Service Periods</em></td>
</tr>
<tr>
<td>TRIGGER</td>
<td>Migrated to entry type <em>Triggers</em></td>
</tr>
<tr>
<td><strong>IMS Keywords</strong>, where Application Messages is stated, this refers to the MESSAGES/USER DATA policy item of the appropriate IMS subsystem definition.</td>
<td>Application Messages</td>
</tr>
<tr>
<td>ABCODEPROG</td>
<td>Format change, Application Messages</td>
</tr>
<tr>
<td>ABCODES, ACTCODES, ALTCODES</td>
<td>Application Messages</td>
</tr>
<tr>
<td>ABCODETRAN</td>
<td>Format change, Application Messages</td>
</tr>
<tr>
<td>ACORESTART</td>
<td>Application Messages</td>
</tr>
<tr>
<td>AREA</td>
<td>Obsolete, new IMS State/Action, link to APL</td>
</tr>
<tr>
<td>AUTOOPS</td>
<td>Auto Operators (not in IMSCNTL any more)</td>
</tr>
<tr>
<td>BRO</td>
<td>Application Messages</td>
</tr>
<tr>
<td>CHE</td>
<td>Application Messages</td>
</tr>
<tr>
<td>CQS0031A</td>
<td>Application Messages</td>
</tr>
<tr>
<td>CQS0032A</td>
<td>Application Messages</td>
</tr>
</tbody>
</table>
### Table 11. Cross Reference of SA OS/390 1.3 Feature Keywords to SA z/OS 2.3 Dialog Entries (continued)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>SA z/OS 2.3 Migration/Conversion Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>CQS0033A</td>
<td>Application Messages</td>
</tr>
<tr>
<td>CQSET</td>
<td>Application Messages</td>
</tr>
<tr>
<td>DFS2142</td>
<td>Application Messages</td>
</tr>
<tr>
<td>DFS2161I</td>
<td>Application Messages</td>
</tr>
<tr>
<td>DFS2169I</td>
<td>Application Messages</td>
</tr>
<tr>
<td>DFS3258A</td>
<td>Application Messages</td>
</tr>
<tr>
<td>DFS3869A/DFS3869I</td>
<td>Application Messages</td>
</tr>
<tr>
<td>DFS554A</td>
<td>Application Messages</td>
</tr>
<tr>
<td>DFS810A</td>
<td>Application Messages</td>
</tr>
<tr>
<td>DFS989I</td>
<td>Application Messages</td>
</tr>
<tr>
<td>DFS994I</td>
<td>Application Messages</td>
</tr>
<tr>
<td>ENVIRON</td>
<td>New Application policy item IMS ENVIRON</td>
</tr>
<tr>
<td>ENVIRON SETUP</td>
<td>Obsolete</td>
</tr>
<tr>
<td>EXTCOND</td>
<td>Migrated to entry type Events</td>
</tr>
<tr>
<td>FORCE</td>
<td>Application Messages</td>
</tr>
<tr>
<td>HOLDQ</td>
<td>Application Messages</td>
</tr>
<tr>
<td>IMSCNTL</td>
<td>New Application policy item IMS CONTROL</td>
</tr>
<tr>
<td>IMS CRITMSGS</td>
<td>Obsolete</td>
</tr>
<tr>
<td>IMSGROUP</td>
<td>no longer used</td>
</tr>
<tr>
<td>IMSRSENM</td>
<td>New, IMS RESOURCE NAME Policy</td>
</tr>
<tr>
<td>INITSTART</td>
<td>Application Messages</td>
</tr>
<tr>
<td>IPS</td>
<td>Application Messages</td>
</tr>
<tr>
<td>NTFYOP</td>
<td>Application Messages</td>
</tr>
<tr>
<td>OLDS</td>
<td>Application Messages</td>
</tr>
<tr>
<td>POSTCHKP</td>
<td>Application Messages</td>
</tr>
<tr>
<td>PRECHKP</td>
<td>Application Messages</td>
</tr>
<tr>
<td>PRODUCT</td>
<td>Obsolete, new IMS State/Action, link to APL</td>
</tr>
<tr>
<td>RECONS</td>
<td>Application Messages</td>
</tr>
<tr>
<td>RECOVERY</td>
<td>Application Messages</td>
</tr>
<tr>
<td>RELEASEQ</td>
<td>Application Messages</td>
</tr>
<tr>
<td>RESTART</td>
<td>Application Messages</td>
</tr>
<tr>
<td>RESTARTABORT</td>
<td>Application Messages</td>
</tr>
<tr>
<td>SDF</td>
<td>SDF policies</td>
</tr>
<tr>
<td>SERVICE</td>
<td>Migrated to entry type Service Periods</td>
</tr>
<tr>
<td>SHUTINIT</td>
<td>Application Messages</td>
</tr>
<tr>
<td>SHUTNORM</td>
<td>Application Messages</td>
</tr>
<tr>
<td>SHUTIMMED</td>
<td>Application Messages</td>
</tr>
<tr>
<td>SHUTFORCE</td>
<td>Application Messages</td>
</tr>
<tr>
<td>SHUTTYPES</td>
<td>Application Messages</td>
</tr>
</tbody>
</table>
Table 11. Cross Reference of SA OS/390 1.3 Feature Keywords to SA z/OS 2.3 Dialog Entries (continued)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>SA z/OS 2.3 Migration/Conversion Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNAPQ</td>
<td>Application Messages</td>
</tr>
<tr>
<td>START</td>
<td>Application Messages</td>
</tr>
<tr>
<td>STARTUP</td>
<td>Application Messages</td>
</tr>
<tr>
<td>STOPBMPREGION</td>
<td>Application Messages</td>
</tr>
<tr>
<td>STOPFFREGION</td>
<td>Application Messages</td>
</tr>
<tr>
<td>STOPREGION</td>
<td>Application Messages</td>
</tr>
<tr>
<td>STSFILE</td>
<td>New, ISF entry type, policy object IMS status file, policy item IMS STATUS FILE</td>
</tr>
<tr>
<td>SUBSYSTEM</td>
<td>Application Policy (TYPE=IMS)</td>
</tr>
<tr>
<td>TCO</td>
<td>Application Messages</td>
</tr>
<tr>
<td>TCOMEMBERS</td>
<td>Application Messages</td>
</tr>
<tr>
<td>THRESHOLDS</td>
<td>New IMS resource thresholds</td>
</tr>
<tr>
<td>TPABEND</td>
<td>Application Messages</td>
</tr>
<tr>
<td>TRIGGER</td>
<td>Migrated to entry type Triggers</td>
</tr>
<tr>
<td>UNLKAVM</td>
<td>Application Messages</td>
</tr>
<tr>
<td>UNLOCK</td>
<td>Application Messages</td>
</tr>
<tr>
<td>UP</td>
<td>Application Messages</td>
</tr>
<tr>
<td>USERSTART</td>
<td>Application Messages</td>
</tr>
<tr>
<td>VTAMTERMS</td>
<td>Application Messages</td>
</tr>
<tr>
<td>%INCLUDE</td>
<td>Obsolete (no longer permitted)</td>
</tr>
<tr>
<td>$PI</td>
<td>Application Messages</td>
</tr>
<tr>
<td>$SI</td>
<td>Application Messages</td>
</tr>
<tr>
<td>$TI</td>
<td>Application Messages</td>
</tr>
</tbody>
</table>

**OPC Keywords**, where Application Messages is stated, this refers to the MESSAGES/USER DATA policy item of the appropriate OPC subsystem definition.

<table>
<thead>
<tr>
<th>OPC (CODE)</th>
<th>Application Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPCACMD</td>
<td>Application Messages</td>
</tr>
<tr>
<td>OPC DOMAINID</td>
<td>New, WORKSTATION DOMAINS policy</td>
</tr>
<tr>
<td>ENVIRON OPCAO</td>
<td>New, OPC SYSTEM DETAILS policy</td>
</tr>
<tr>
<td>OPCAPARM</td>
<td>Application Messages</td>
</tr>
<tr>
<td>OPC PCS</td>
<td>New, CONTROLLER DETAILS policy</td>
</tr>
<tr>
<td>OPC OPCSRSSTAT</td>
<td>New, SPECIAL RESOURCES policy linked to CONTROLLER DETAILS</td>
</tr>
<tr>
<td>OPCCNTL</td>
<td>New, Application OPC CONTROL policy</td>
</tr>
</tbody>
</table>

**Selective Policy Database Conversion**

After your entries have been migrated to the policy database, the conversion is done to convert the old data which might have been migrated to the policy database to the new format. The migration functions automatically start the conversion function.
Initial Policy Database Conversion

Note:

The conversion function, after having migrated, will not start the product automation conversion as this has already been performed by the migration.

Generating a Job for Batch Conversion

You can run a job for batch conversion that performs the same processing as the conversion function of the customization dialog. The SINGSAMP sample library contains a sample job for this task in member INGEBCNV. Before submitting this job, you need to edit the jobcard appropriately as described in the sample itself.
Appendix A. Concurrent Customization by Multiple Users

Concurrent user access is supported for Application, ApplicationGroup, and Monitor Resource.

Concurrent user access allows simultaneous update for applications and application groups, that means, several users can open the same policy database, select entry types ApplicationGroup, Application, and Monitor Resource and perform updates for different objects at the same time.

If such entry types have been updated, and several builds are issued for different output data sets, the changed object is rebuilt whenever there are changes since the last build for that output data set, thus guaranteeing consistent and complete automation control files.

The following subsections describe the consequences for users working simultaneously on these entry types within one policy database.

Concurrent User Access Capabilities

If a policy database is selected for processing (command OPEN), access to that policy database for other users is possible, that is, multiple opening of a policy database is supported.

Working on entry types other than Application, ApplicationGroup, or Monitor Resource with several users, or performing commands other than OPEN or SELECT against a policy database while an application, application group, or monitor resource is being edited, may not work.

This limitation depends on the data tables that are needed for the selected action or command, and it is indicated by specific messages. The most common situations are described in the following sections.

Policy Definitions Usage

If you want to change your Enterprise Policy Options by selecting option 10 on the Customization Dialog Settings Menu panel (see Figure 266 on page 370), while another user works with a policy database, the selection is not possible and an appropriate message is displayed.

On the other hand, if you are accessing the SA z/OS policy items for modification, this will prevent other users from accessing any policy item and an appropriate message is displayed.

Policy Database Selection

If you want to issue a BUILD, DELETE, MIGRATE, or REPORT command from the Policy Data Base Selection panel while another user has accessed the same policy database, the command is rejected and an appropriate message is displayed.
Concurrent User Access Capabilities

Conversely, if you request access to a policy database via S (Select) or O (Open), while a BUILD, DELETE, MIGRATE, or REPORT is in progress for this policy database, the access is rejected and an appropriate message is displayed.

For other commands there are also limitations, if a user has already opened the specific policy database. These limitations depend upon the selected entry types and actions, and are indicated by an appropriate message.

Entry Type Selection
If a policy database is opened successfully (with S (Select) or O (Open)), the Entry Type Selection panel is displayed. Here a specific Entry Type can be selected, or a BUILD can be requested.

If a BUILD command is issued from this screen, but another user has also opened this policy database to work with applications or any other entry type, then the command is rejected and an appropriate message is displayed.

If an entry type other than ApplicationGroup, Application, or Monitor Resource is selected, and another user has also selected an entry type of the same policy database, the selection may be rejected depending on the data tables needed for the selected entry type, and an appropriate message is displayed. In this case, editing is not possible, because concurrent user access is provided only for Applications, ApplicationGroups, and Monitor Resources.

Entry Name Selection
On the Entry Name Selection panel a specific application can be selected, and a BUILD can be requested.

If you try a BUILD action, and the policy database has been simultaneously opened by another user, the command is rejected and an appropriate message is displayed.

If another user is currently editing the same application in the same policy database, then access is allowed only to view the data, that is, changing the displayed data is not allowed. In this case, however, it is possible to select that application and view the data. A warning message is displayed on the Policy Selection panel, indicating that no updates will be accepted.

If, nevertheless, the displayed values are changed on the data entry panel for that policy option, these changes are not accepted when leaving that panel. If you try to leave the panel with the END command, a message will remind you that you must leave this panel with CANCEL.

Changing Links
The WHERE USED policy item for Applications allows you to change the links of one application to one or more application groups.

The WHERE USED policy item for ApplicationGroups allows you to change the links of one application group to one or more systems or sysplexes.

The WHERE USED policy item for Monitor Resources allows you to change the links of one monitor resource to one or more monitor resources.
You can only add or remove such links, if you exclusively access the entry types 
*Application*, *ApplicationGroup*, or *Monitor Resource*. If another user is currently 
editing them, then a message is displayed indicating that updates are not possible.

The same is true for the trigger policy item where you can maintain the link of the 
current application, application group, or monitor resource to a trigger.

Automated NetView Automation Table Build

If you have the automated AT and MPFLSTSA build enabled (this is the default 
behavior), each policy change that affects the ATs or MPFLSTSA member is 
recorded for each system. This requires the exclusive access of entry type SYSTEM. 
If another user is currently using these files, then a message is displayed that 
updates are not possible.
Appendix B. More about Using the Customization Dialog

Getting Help .......................... 369  User-written Functions ................. 377
Changing Your Customization Dialog Environment 369  Alternate Configuration Support ........ 378
Extending Policy Definitions .............. 369  How to Define Alternate Configurations for
Defining Data Sets for Batch Processing .... 371  Systems .................................. 378
Define ISPF Temporary Data Sets for Batch .... 375  How to Build Alternate Configurations .... 379
Build .................................... 376  Restrictions .............................. 379

Use this appendix to find out how to get help when using the customization dialog and how to change the customization dialog environment.

Getting Help

Online help is available for the customization dialog. You can display online help by entering the HELP command (or press the PF key associated with the HELP command) while on a particular dialog panel.

When selecting HELP from the action bar, the pull-down lists the following selections:

Automation Help
This help provides online information about important functions of the customization dialog. It combines the help panels of the functional panels within the customization dialog. This combination allows you to view the help panels sequentially, or you can choose selected topics from lists that are displayed on many of the help pages.

Changes for this Release
This panel provides a list of the changes and enhancements that have been added to the customization dialog for the different releases of version 2.

Changing Your Customization Dialog Environment

You can invoke the Settings function from the Customization Dialog Primary Menu or from the Menu pull-down of the Policy Data Base Selection panel.

The first three options on the Customization Dialog Settings Menu panel (see Figure 266 on page 370) allow you to change the customization dialog environment for your user ID. This is a usability function only and has no effect on the policy database contents.
### ALIAS

This option allows you to change commands and command abbreviations that can be entered on the command line or action column in the customization dialog.

### DISPLAY

This option allows you to change the line spacing for lists that appear in the customization dialog.

### MESSAGE

This option allows you to specify the level of detail in the messages you receive when you run various customization dialog options or batch jobs, and whether or not the messages are written to the ISPF log.

These settings are stored in the ISPF application profile. To have them available to the batch processing (build or migrate), make sure that your ISPF profile data set is allocated to the table library, DDname: AOFTLIB, as described in "Defining Data Sets for Batch Processing" on page 375.

### CONVERT

This option lets you specify whether SA z/OS should create an automation name for an ApplicationGroup resource during the very first conversion of a policy database. Refer to "Initial Policy Database Conversion" on page 355 for more information.

### POLICY

See "Extending Policy Definitions” on page 371

### DSNAMES

See "Defining Data Sets for Batch Processing” on page 375

### WORKDSN

See "Define ISPF Temporary Data Sets for Batch Build” on page 376
Extending Policy Definitions

With SA z/OS you can extend your policy definitions by creating new policy items or modifying existing ones. Any user-defined policy items that you create or modify are now kept separate from the SA z/OS policies. This means that with future releases of SA z/OS you will not have to redefine those policy items that you have created or modified.

To create new policy items or modify existing ones, you should perform the following steps:

1. Select option 10 POLICY from the Customization Dialog Settings Menu (see Figure 266 on page 370). The Entry Type Selection panel is then displayed, as shown in Figure 267 (this figure does not show all 36 available entry types).

   ![Table showing entry types]

   **Figure 267. Entry Type Selection Panel**

2. Select the entry type for which you want to create a new policy item or modify one. The Edit Policy Selections panel is displayed, listing all current policy items for this entry type, as shown in Figure 268 on page 372. Any user-defined policy items are indicated by USR in the Status column.
Changing Your Customization Dialog Environment

3. On the Edit Policy Selections panel you can:
   - Create a new entry by entering NEW on the command line. The new entry is added to the top of the list.
   - Create a new entry by entering INSERT in the Action column. The new entry is added after the entry where INSERT was specified.
   - Edit an entry by entering EDIT in the Action column.

   **Note:** Any modifications to the existing SA z/OS policy definitions are solely your responsibility.

   - Delete a user-defined entry by entering DELETE in the Action column.
   - Suppress an SA z/OS entry by entering REMOVE in the Action column.
   - Activate a deselected entry by entering SELECT in the Action column.
   - Leave this panel saving all changes by entering the END command.
   - Discard all changes by entering the CANCEL command.

4. If you choose to insert a new policy item for an entry type, or edit an existing policy item, the Define Policy Selection panel is displayed. The example in Figure 269 on page 373 shows how to define policy item APPLICATION INFO of entry type Application.
If you want to create policy items you must specify an ISPF action for the selection of the new policy item from the Policy Selection panel. You must then create your own ISPF input panel for the new policy item.

Each policy selection consists of information defining:

- How the policy appears on the Policy Selection panel.
- How the policy is processed when it is selected.
- Under what conditions the policy should be displayed.

This panel has the following fields:

**Entry Type**
Indicates the entry for which the policy is being defined. This field is set based on the entry type you had previously selected.

**Policy Name**
This is the name of the policy you are defining. It appears in the first column of the Policy Selection panel. This field is required.

**Description**
This is a short description of the policy. It appears in the second column of the Policy Selection panel. This field is required.

**Help Action**
Specifies the action to be taken when policy help is requested (by entering HELP in the Action column on the Policy Selection panel). The action must be a valid ISPF SELECT service. For example, to display an ISPF panel enter:

```
PGM(ISPTUTOR) PARM(panel-name)
```

The help text should describe the function provided by this policy rather than describing how to enter the data for the policy.

This field is optional; if no action is specified then HELP will not be available from the Policy Selection panel for this policy.

**Edit Action**
Specifies the action to be taken when the policy is selected from the Policy Selection panel. The action must be a valid ISPF SELECT Service. This field is required.

---

**Figure 269. Define Policy Selection Panel**

If you want to create policy items you must specify an ISPF action for the selection of the new policy item from the Policy Selection panel. You must then create your own ISPF input panel for the new policy item.

Each policy selection consists of information defining:

- How the policy appears on the Policy Selection panel.
- How the policy is processed when it is selected.
- Under what conditions the policy should be displayed.

This panel has the following fields:

**Entry Type**
Indicates the entry for which the policy is being defined. This field is set based on the entry type you had previously selected.

**Policy Name**
This is the name of the policy you are defining. It appears in the first column of the Policy Selection panel. This field is required.

**Description**
This is a short description of the policy. It appears in the second column of the Policy Selection panel. This field is required.

**Help Action**
Specifies the action to be taken when policy help is requested (by entering HELP in the Action column on the Policy Selection panel). The action must be a valid ISPF SELECT service. For example, to display an ISPF panel enter:

```
PGM(ISPTUTOR) PARM(panel-name)
```

The help text should describe the function provided by this policy rather than describing how to enter the data for the policy.

This field is optional; if no action is specified then HELP will not be available from the Policy Selection panel for this policy.

**Edit Action**
Specifies the action to be taken when the policy is selected from the Policy Selection panel. The action must be a valid ISPF SELECT Service. This field is required.
These fields are not shown on the Policy Selection panel. They can be used for documentation purposes. These fields are optional.

You can also specify filtering conditions, that is, attributes of the entry being processed from the Policy Selection panel that can be used to determine whether or not this policy should be displayed. The following filtering information can be specified:

**Filter attributes**

This specifies attributes that an entry must have in order for this policy to be displayed. This field contains multiple words, each word identifying one attribute. All the attributes specified must be satisfied in order for the policy to be displayed. If no attributes are specified then the policy is valid for all entries. The attributes that can be specified are defined by SA z/OS and differ based on the entry type, as shown in Table 12.

**Table 12. Policy Selection Filter Attributes**

<table>
<thead>
<tr>
<th>Entry Type</th>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td>Group type</td>
<td>(blank) STANDARD SYSPLEX</td>
</tr>
<tr>
<td>System</td>
<td>Operating System type</td>
<td>(blank) CF LINUX MVS VM VSE</td>
</tr>
<tr>
<td>ApplicationGroup</td>
<td>ApplicationGroup type</td>
<td>(blank) SYSPLEX SYSTEM</td>
</tr>
<tr>
<td>Application</td>
<td>Object type</td>
<td>(blank) CLASS INSTANCE</td>
</tr>
<tr>
<td></td>
<td>Application type</td>
<td>(blank) IMAGE JES2 JES3 CICS IMS DB2 OPC USS</td>
</tr>
<tr>
<td>Processor</td>
<td>Processor type/mode</td>
<td>(blank) LPAR ESA PSM</td>
</tr>
<tr>
<td>Others</td>
<td>none</td>
<td></td>
</tr>
</tbody>
</table>

The following rules apply to the filtering attributes:

1. If no attributes are specified (indicated by *blank* in Table 12), the policy is always displayed for all entries.
2. If multiple attributes are specified then all attributes must be satisfied in order for the policy to be displayed. For example:
Changing Your Customization Dialog Environment

**JES2 CLASS** This displays the policy for application entries that are classes and are of type JES2.

**JES2** This displays the policy for application entries that are of type JES2 and either class or instances.

3. For each ATTRIBUTE, the words listed in the VALUE column are mutually exclusive. If you specify both values the policy will never be displayed as both conditions cannot be true at the same time. For example:

**JES2 JES3** This is never displayed because an application can be either JES2 or JES3 but not both.

4. No validity checking is done on the attribute values. If an invalid value is specified the policy will never be displayed since the condition cannot be satisfied. For example:

**JES4** This is never displayed because an application cannot be of type JES4.

---

**Defining Data Sets for Batch Processing**

Use Option 11 of the Customization Dialog Settings Menu panel to define the batch processing data sets. This needs to be done only once, when you install SA z/OS. For details, see System Automation for z/OS Planning and Implementation.

To use the batch options of the build or migrate function you must first specify the names of the data sets required for the batch processing. These names form the DD name concatenations to be included in the JCL of the batch processing job.

---

<table>
<thead>
<tr>
<th>Command</th>
<th>Data Set Customization Details</th>
<th>Volume Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFGCUSD</td>
<td>DDname Seq Bas Data set name</td>
<td>Volume Unit</td>
</tr>
<tr>
<td>AOFEXEC 98 YES 'ING.SINGIREX'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AOFEXEC 99 YES 'SYS1.ISPF.ISPEXEC'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AOFIPDB 99 YES 'ING.SINGIPDB'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AOFLLIB 98 YES 'ING.SINGIMOD'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AOFLLIB 99 YES 'SYS1.ISPF.ISPLLIB'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AOFMLIB 98 YES 'ING.SINGIMSG'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AOFMLIB 99 YES 'SYS1.ISPF.ISPMENU'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AOFPLIB 98 YES 'ING.SINGIPNL'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AOFPLIB 99 YES 'SYS1.ISPF.ISPPENU'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AOFSLIB 98 YES 'ING.SINGISKL'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AOFSLIB 99 YES 'SYS1.ISPF.ISPSLIB'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AOFSTEP 99 YES 'ING.SINGIMOD'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AOFLIB 98 NO 'HART.ISPF.ISPPROF'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AOFLIB 97 NO 'HART.USER.ISPITBL'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AOFLIB 98 YES 'ING.SINGITBL'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AOFLIB 99 YES 'SYS1.ISPF.ISPTENU'</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 270. SA z/OS Data Set Customization Details Panel**

On the Data Set Customization Details panel, a table is displayed with groups of data definition (DD) names and data set names. You must modify the data set names in accordance with your local naming conventions.

Each column can be modified as follows:

**DDname**

No new DD names may be added, but another entry for an existing DD name may be added at the end of the table. It will be inserted at the
correct position within the table when saved. An entry may be deleted by blanking out the DD name, but if an entry is removed incorrectly then the batch functions become inoperative.

**Seq** This entry should contain a number from 1 to 99. This value will position the entry into the correct concatenation sequence when the JCL is built. The higher the number the further down the concatenation sequence it will appear. The values 98 and 99 are reserved.

**Base** The Base column indicates a reserved DD name entry and should not be removed. All entries that are not reserved (see the Seq column description) must have the value of "NO".

**Data Set Name**
This is the fully qualified data set name, including quotation marks, that are represented by this DD name.

**Volume**
May be left blank for catalogued data sets, otherwise enter the label ID of the volume on which this data set resides (as would be coded for an ALLOC JCL entry). SMS may be entered for any data set under space management.

**Unit** Must be left blank for catalogued and/or space managed data sets, otherwise you must enter the unit relevant to the volume on which this data set resides (as would be coded for an ALLOC JCL entry).

AOFIPDB is needed to get access to the automation manager logic file.

AOFTLIB needs two data sets to be allocated additionally to the ISPF and SA z/OS table libraries:
- The user’s ISPF table output library is needed to get access to the tables with user modifications, for example, the data set specifications;
- The user’s ISPF profile is needed to get access to the dialog settings, for example, messages and logging options.

In our example from Figure 270 on page 375 these are:

'HART.ISPF.ISPROF'
'HART.USER.SINGITBL'

**Note:**
The user’s ISPF table output library should also be allocated to DDname AOFTABL.

**Define ISPF Temporary Data Sets for Batch Build**

Use Option 12 on the *Customization Dialog Settings Menu* panel to define the data sets for the batch build. SA z/OS displays the following panel:
This panel allows to add or modify job control data to preallocate temporary ISPF data sets, which are used for the batch build. The build function creates the ACF and AMC output files in two steps. First, all members are created and stored into a temporary file using ISPF file tailoring. In a second step this file is written to disk creating all the individual members.

To hold the intermediate files, ISPF work and control data sets are used. If the default allocations of ISPF are not sufficient, they can be changed on this panel.

The DDnames ISPWRKx are used by ISPF for file tailoring services. The DDnames ISPCTLx are used by ISPF to hold the file tailoring output and to pass it to the subsequent functions that write the members to the output data set.

User-written Functions

Option U User on the Customization Dialog Primary Menu displays the User-written Functions Selection Menu. This menu allows for adding installation specific functions to the customization dialog.

As an example, one selection has already been provided:
1 - Transmit Transmit build output to a remote node

If you select this option 1, SA z/OS displays the Send Policy Data Base panel (see Figure 271 on page 378), where you can distribute policy databases to remote nodes.

To add further selections, edit the panel (member AOFGUSR) and:
- add selection number and description to the panel )BODY section
- add a line to the TRANSLATE instruction in the panel )PROC section, containing the selection number and the command string which invokes the new function.

The Send Policy Data Base panel allows you to transmit a build output data set to a specified node and user ID.
With the parameter specified on this screen, a TSO TRANSMIT command is created. The transmission is started by issuing the SEND command (PF 6).

**Data set name**
This entry field is filled with the build output data set name of the current policy database, if that data set name is available. The name is always fully qualified and follows the TSO data set naming conventions.

**User ID or nickname**
allows to specify the target user ID or nickname. Input in this field is saved across sessions.

**Node ID**
allows to specify the target node. This field may remain empty, if a nickname has been specified. Input in this field is saved across sessions.

**Command parameters**
allows to specify parameters for the TRANSMIT command. Any values specified here are appended to the TRANSMIT command as they have been typed in. Input in this field is saved across sessions.

---

**Alternate Configuration Support**

There may be situations, for example, in disaster recovery or on weekends, when you want to have an alternative automation configuration for one or more systems out of your enterprise (policy database).

The customization dialog allows you to specify an alternate configuration by:
- marking a system as part of an alternate configuration
- allowing a specific build for normal or alternate configurations
- enforcing the build of normal or alternate configurations into different output data sets

All other definitions and links remain unchanged, for example, an application group can be linked to a normal systems, an alternate systems, or to both.

**How to Define Alternate Configurations for Systems**

For each system, either on the Define New Entry panel or in the policy item SYSTEM INFO, the existing MVS system name in field MVS SYSNAME allows you
Alternate Configuration Support

to add a "+" sign to the system name. This will be used as an indication that this system belongs to the alternate configuration. The "+" sign can only be added to the end of the system name.

Furthermore, this "+" sign can be specified in the supporting resource definition of a relationship, and on the initial migration panel.

When a new system is defined and its processor operations name (in field ProcOps name) is defaulted with the MVS system name, any "+" is removed.

In the policy item AUTOMATION SETUP for Systems the fields MVS SYSNAME and SDF Root Name are defaulted with the MVS system name and any "+" is removed (see also Figure 88 on page 117).

When linking systems to a group or subgroup, the configuration type is not checked. If for example, a sysplex is intended for an alternate configuration, you have to ensure that only systems for this configuration are linked to that sysplex.

How to Build Alternate Configurations

The Build Parameters panel (see Figure 247 on page 319) has an option to select either a normal or alternate configuration build. The default is NORMAL. To avoid overwriting an existing configuration with wrong data, the build checks the configuration type in the build output data set. If there is already a configuration of a different type, the build is not started, and a warning message is displayed. If the policy database contains a normal and an alternate configuration, they have to be built into different output data sets.

Furthermore, the policy database name is checked to prevent an automation control file being destroyed by building a configuration into the wrong output data set.

When building an alternate configuration, all "+" signs are removed from the MVS system name. Thus the generated automation configuration files can be used for automation simply by switching the data sets.

If a policy database contains two configurations, the build picks out the systems that are linked to the selected configuration:

- The ACF map AOFACFEMP contains only the systems marked for the selected configuration via the "+" sign added to the MVS system name.
- System include members (ACFZxxx) are generated only for the systems of the selected configuration.
- The AMC file, which is sysplex oriented, consists of members for each sysplex and standalone system. In each sysplex, it contains only the systems that are marked for the selected configuration, and standalone system members are created only for systems of the selected configuration.
- The ATs are generated only for the systems of the selected configuration.
- The MPFLSTSA member is built for the messages that belong to the selected configuration.

Restrictions

For performance reasons, the build with option 1, which is a build of the complete policy database, does not check each object as to which system it is linked to. So the build output data set will contain fragments of each object in the policy database, regardless of which configuration they are linked to. Of course,
fragments of the wrong configuration cannot be accessed by automation, because they are not included via system main include members. However, the build output data sets need to be large enough to hold all fragments.

Furthermore, with build option 3, it is possible to update each fragment in a data set, independent of its link to a configuration.

The Unlinked report (see Chapter 7, “Creating Reports about Policy Databases,” on page 307) does not include relationships that have been defined across configurations. That means, if a relationship is defined between a resource in configuration NRM and another resource in configuration ALT, it is not included by the build function. However, the Unlinked report does not show it as not used.

It is not possible to define processor operations proxy resources for systems belonging to the alternate configuration, because the Processor Operations name must be different for all systems. For examples on how to use proxy definitions see System Automation for z/OS Customizing and Programming.
Appendix C. Message Definitions That Do Not Create AT Entries

The following are reserved keywords that must not be used as messages IDs otherwise no AT entry will be built for them:

<table>
<thead>
<tr>
<th>ABENDED</th>
<th>ABENDING</th>
<th>ACORESTART</th>
<th>ACTIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTODOWN</td>
<td>BREAKING</td>
<td>BRO</td>
<td>BROKEN</td>
</tr>
<tr>
<td>CHE</td>
<td>CITIME</td>
<td>CQSET</td>
<td>CTDOWN</td>
</tr>
<tr>
<td>DOMAINID</td>
<td>FORCE</td>
<td>HALTED</td>
<td>HOLDQ</td>
</tr>
<tr>
<td>INACTIVE</td>
<td>JESABEND</td>
<td>LOGREC</td>
<td>MDSCOUNTA-MDSCOUNTZ</td>
</tr>
<tr>
<td>MVSDUMP</td>
<td>MVSDUMPRESET</td>
<td>NOJSM</td>
<td>OPCACMD</td>
</tr>
<tr>
<td>POSTCHKP</td>
<td>PPLACTIVE</td>
<td>PRECHKP</td>
<td>RCVRTRAN</td>
</tr>
<tr>
<td>RCVRVIOL</td>
<td>RELEASEQ</td>
<td>RESTARTABORT</td>
<td>RUNNING</td>
</tr>
<tr>
<td>SHUTFORCEDDF</td>
<td>SHUTTYPES</td>
<td>SNAPQ</td>
<td>STADC</td>
</tr>
<tr>
<td>START</td>
<td>STARTED2</td>
<td>STOPBMPREGION</td>
<td>STOPFPREGION</td>
</tr>
<tr>
<td>STOPPED</td>
<td>STOPPING</td>
<td>STOPREGION</td>
<td>STUCK</td>
</tr>
<tr>
<td>TCO</td>
<td>UNLKAVM</td>
<td>UNLOCK</td>
<td>UP</td>
</tr>
<tr>
<td>VTAMTERMS</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Glossary

This glossary includes terms and definitions from:

- The American National Standard Dictionary for Information Systems, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies can be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018. Definitions are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.

The following cross-references are used in this glossary:

Contrast with. This refers to a term that has an opposed or substantively different meaning.

 Deprecated term for. This indicates that the term should not be used. It refers to a preferred term, which is defined in its proper place in the glossary.

See. This refers the reader to multiple-word terms in which this term appears.

See also. This refers the reader to terms that have a related, but not synonymous, meaning.

Synonym for. This indicates that the term has the same meaning as a preferred term, which is defined in the glossary.

 Synonymous with. This is a backward reference from a defined term to all other terms that have the same meaning.

A

ACF. Automation control file.

ACF/NCP. Advanced Communications Function for the Network Control Program. See Advanced Communications Function and Network Control Program.


active monitoring. In SA z/OS, the acquiring of resource status information by soliciting such information at regular, user-defined intervals. See also passive monitoring.

adapter. Hardware card that enables a device, such as a workstation, to communicate with another device, such as a monitor, a printer, or some other I/O device.

Address Space Workflow. In RMF, a measure of how a job uses system resources and the speed at which the job moves through the system. A low workflow indicates that a job has few of the resources it needs and is contending with other jobs for system resources. A high workflow indicates that a job has all the resources it needs to execute.

adjacent hosts. Systems connected in a peer relationship using adjacent NetView sessions for purposes of monitoring and control.

adjacent NetView. In SA z/OS, the system defined as the communication path between two SA z/OS systems that do not have a direct link. An adjacent NetView is used for message forwarding and as a communication link between two SA z/OS systems. For example, the adjacent NetView is used when sending responses from a focal point to a remote system.

Advanced Communications Function (ACF). A group of IBM licensed programs (principally VTAM, TCAM, NCP, and SSP) that use the concepts of Systems Network Architecture (SNA), including distribution of function and resource sharing.

advanced program-to-program communication (APPC). A set of inter-program communication services that support cooperative transaction processing in a Systems Network Architecture (SNA) network. APPC is the implementation, on a given system, of SNA’s logical unit type 6.2.

alert. (1) In SNA, a record sent to a system problem management focal point or to a collection point to communicate the existence of an alert condition. (2) In NetView, a high-priority event that warrants immediate
attention. A database record is generated for certain event types that are defined by user-constructed filters.

**alert condition.** A problem or impending problem for which some or all of the process of problem determination, diagnosis, and resolution is expected to require action at a control point.

**alert focal-point system.** See entry for NPDA focal-point system under focal—point system.

**alert threshold.** An application or volume service value that determines the level at which SA z/OS changes the associated icon in the graphic interface to the alert color. SA z/OS may also issue an alert. See warning threshold.

AMC. (1) Automation Manager Configuration (2) The Auto Msg Classes entry type

APF. Authorized program facility.

API. Application programming interface.

APPC. Advanced program-to-program communications.

**application.** An z/OS subsystem or job monitored by SA z/OS.

**Application entry.** A construct, created with the customization dialogs, used to represent and contain policy for an application.

**application group.** A named set of applications. An application group is part of an SA z/OS enterprise definition and is used for monitoring purposes.

**ApplicationGroup entry.** A construct, created with the customization dialogs, used to represent and contain policy for an application group.

**application program.** (1) A program written for or by a user that applies to the user’s work, such as a program that does inventory or payroll. (2) A program used to connect and communicate with stations in a network, enabling users to perform application-oriented activities.

ARM. Automatic restart management.

ASCB. Address space control block.

**ASCB status.** An application status derived by SA z/OS running a routine (the ASCB checker) that searches the z/OS address space control blocks (ASCBs) for address spaces with a particular job name. The job name used by the ASCB checker is the job name defined in the customization dialog for the application.

**ASCII (American National Standard Code for Information Interchange).** The standard code, using a coded character set consisting of 7-bit coded characters (8-bit including parity check), for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters. (A)

**ASF.** Automation status file.

**assist mode facility.** An SA z/OS facility that uses SDF and enables interaction with automation before SA z/OS takes an automation action. SDF prompts the operator with a suggested action, then provides options for using that action, modifying and using the action, or canceling the action. Also called assist mode, it is enabled using the customization dialogs, or dynamically.

**authorized program facility (APF).** A facility that permits identification of programs that are authorized to use restricted functions.

**automated function.** SA z/OS automated functions are automation operators, NetView autotasks that are assigned to perform specific automation functions. However, SA z/OS defines its own synonyms, or automated function names, for the NetView autotasks, and these function names are referred to in the sample policy databases provided by SA z/OS. For example, the automation operator AUTBASE corresponds to the SA z/OS automated function BASEOPER.

**automated console operations (ACO).** The concept (versus a product) of using computers to perform a large subset of tasks ordinarily performed by operators, or assisting operators in performing these tasks.

**automatic restart management.** A z/OS recovery function that improves the availability of specified subsystems and applications by automatically restarting them under certain circumstances. Automatic restart management is a function of the Cross-System Coupling Facility (XCF) component of z/OS.

**automatic restart management element name.** In MVS 5.2 or later, z/OS automatic restart management requires the specification of a unique sixteen character name for each address space that registers with it. All automatic restart management policy is defined in terms of the element name, including SA z/OS’s interface with it.

**automation.** The automatic initiation of actions in response to detected conditions or events. SA z/OS provides automation for z/OS applications, z/OS components, and remote systems that run z/OS. SA z/OS also provides tools that can be used to develop additional automation.

**automation agent.** In SA z/OS, the automation function is split up between the automation manager and the automation agents. The observing, reacting and doing parts are located within the NetView address...
space, and are known as the **automation agents**. The automation agents are responsible for:

- recovery processing
- message processing
- active monitoring: they propagate status changes to the automation manager

**automation configuration file.** The data set that consists of:

- the automation control file (ACF)
- the automation manager configuration file (AMC)
- the NetView automation table (AT)
- the MPFLSTSA member

**automation control file (ACF).** In SA z/OS, a file that contains system-level automation policy information. There is one master automation control file for each NetView system on which SA z/OS is installed. Additional policy information and all resource status information is contained in the policy database (PDB). The SA z/OS customization dialogs must be used to build the automation control files. They must not be edited manually.

**automation flags.** In SA z/OS, the automation policy settings that determine the operator functions that are automated for a resource and the times during which automation is active. When SA z/OS is running, automation is controlled by automation flag policy settings and override settings (if any) entered by the operator. Automation flags are set using the customization dialogs.

**automation manager.** In SA z/OS, the automation function is split up between the automation manager and the automation agents. The coordination, decision making and controlling functions are processed by each sysplex’s **automation manager.**

The automation manager contains a model of all of the automated resources within the sysplex. The automation agents feed the automation manager with status information and perform the actions that the automation manager tells them to.

The automation manager provides **sysplex-wide** automation.

**Automation Manager Configuration.** The Automation Manager Configuration file (AMC) contains an image of the automated systems in a sysplex or of a standalone system.

**Automation NetView.** In SA z/OS the NetView that performs routine operator tasks with command procedures or uses other ways of automating system and network management, issuing automatic responses to messages and management services units.

**automation operator.** NetView automation operators are NetView autotasks that are assigned to perform specific automation functions. See also **automated function. NetView automation operators may receive messages and process automation procedures. There are no logged-on users associated with automation operators. Each automation operator is an operating system task and runs concurrently with other NetView tasks. An automation operator could be set up to handle JES2 messages that schedule automation procedures, and an automation statement could route such messages to the automation operator. Similar to **operator station task.** SA z/OS message monitor tasks and target control tasks are automation operators.

**automation policy.** The policy information governing automation for individual systems. This includes automation for applications, z/OS subsystems, z/OS data sets, and z/OS components.

**automation policy settings.** The automation policy information contained in the automation control file. This information is entered using the customization dialogs. You can display or modify these settings using the customization dialogs.

**automation procedure.** A sequence of commands, packaged as a NetView command list or a command processor written in a high-level language. An automation procedure performs automation functions and runs under NetView.

**automation status file.** In SA z/OS, a file containing status information for each automated subsystem, component or data set. This information is used by SA z/OS automation when taking action or when determining what action to take. In Release 2 and above of AOC/MVS, status information is also maintained in the operational information base.

**automation table (AT).** See NetView automation table.

**autotask.** A NetView automation task that receives messages and processes automation procedures. There are no logged-on users associated with autotasks. Each autotask is an operating system task and runs concurrently with other NetView tasks. An autotask could be set up to handle JES2 messages that schedule automation procedures, and an automation statement could route such messages to the autotasks. Similar to **operator station task.** SA z/OS message monitor tasks and target control tasks are autotasks. Also called **automation operator.**

**available.** In VTAM programs, pertaining to a logical unit that is active, connected, enabled, and not at its session limit.

**B**

**basic mode.** A central processor mode that does not use logical partitioning. Contrast with **logically partitioned (LPAR) mode.**
BCP Internal Interface. Processor function of CMOS-390, zSeries processor families. It allows the communication between basic control programs such as z/OS and the processor support element in order to exchange information or to perform processor control functions. Programs using this function can perform hardware operations such as ACTIVATE or SYSTEM RESET.

beaconing. The repeated transmission of a frame or messages (beacon) by a console or workstation upon detection of a line break or outage.

BookManager. An IBM product that lets users view softcopy documents on their workstations.

C

central processor (CP). The part of the computer that contains the sequencing and processing facilities for instruction execution, initial program load (IPL), and other machine operations.

central processor complex (CPC). A physical collection of hardware that consists of central storage, one or more central processors, timers, and channels.

central site. In a distributed data processing network, the central site is usually defined as the focal point for alerts, application design, and remote system management tasks such as problem management.

CFR/CFS and ISC/ISR. I/O operations can display and return data about integrated system channels (ISC) connected to a coupling facility and coupling facility receiver (CFR) channels and coupling facility sender (CFS) channels.

channel. A path along which signals can be sent; for example, data channel, output channel. See also link.

channel path identifier. A system-unique value assigned to each channel path.

CHPID. In SA z/OS, channel path ID; the address of a channel.

CHPID port. A label that describes the system name, logical partitions, and channel paths.

channel-attached. (1) Attached directly by I/O channels to a host processor (for example, a channel-attached device). (2) Attached to a controlling unit by cables, rather than by telecommunication lines. Contrast with link-attached. Synonymous with local.

CI. Console integration.

CICS/VS. Customer Information Control System for Virtual Storage.

CLIST. Command list.

close. 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.

close ID. A generic means of handling system-specific values such as the MVS SYSCLOSE or the VTAM subarea number. Close IDs can be substituted into application definitions and commands to customize a basic application definition for the system that it is to be instantiated on.

CNC. A channel path that transfers data between a host system image and an ESCON control unit. It can be point-to-point or switchable.

command. A request for the performance of an operation or the execution of a particular program.

command facility. The component of NetView that is a base for command processors that can monitor, control, automate, and improve the operation of a network. The successor to NCCF.

command list (CLIST). (1) A list of commands and statements, written in the NetView command list language or the REXX language, designed to perform a specific function for the user. In its simplest form, a command list is a list of commands. More complex command lists incorporate variable substitution and conditional logic, making the command list more like a conventional program. Command lists are typically interpreted rather than being compiled. (2) In SA z/OS, REXX command lists that can be used for automation procedures.

command procedure. In NetView, either a command list or a command processor.

command processor. A module designed to perform a specific function. Command processors, which can be written in assembler or a high-level language (HLL), are issued as commands.

Command Tree/2. An OS/2-based program that helps you build commands on an OS/2 window, then routes the commands to the destination you specify (such as a 3270 session, a file, a command line, or an application program). It provides the capability for operators to build commands and route them to a specified destination.

common commands. The SA z/OS subset of the CPC operations management commands.

common routine. One of several SA z/OS programs that perform frequently used automation functions. Common routines can be used to create new automation procedures.

Common User Access (CUA) architecture. Guidelines for the dialog between a human and a workstation or terminal.
communication controller. A type of communication control unit whose operations are controlled by one or more programs stored and executed in the unit or by a program executed in a processor to which the controller is connected. It manages the details of line control and the routing of data through a network.

communication line. Deprecated term for telecommunication line.

connectivity view. In SA z/OS, a display that uses graphic images for I/O devices and lines to show how they are connected.

console automation. The process of having NetView facilities provide the console input usually handled by the operator.

console connection. In SA z/OS, the 3270 or ASCII (serial) connection between a PS/2 computer and a target system. Through this connection, the workstation appears (to the target system) to be a console.

console integration (CI). A hardware facility that if supported by an operating system, allows operating system messages to be transferred through an internal hardware interface for display on a system console. Conversely, it allows operating system commands entered at a system console to be transferred through an internal hardware interface to the operating system for processing.

consoles. Workstations and 3270-type devices that manage your enterprise.

Control units. Hardware units that control I/O operations for one or more devices. You can view information about control units through I/O operations, and can start or stop data going to them by blocking and unblocking ports.

controller. A unit that controls I/O operations for one or more devices.

couple data set. A data set that is created through the XCF couple data set format utility and, depending on its designated type, is shared by some or all of the z/OS systems in a sysplex. See also synplex couple data set and XCF couple data set.

coupling facility. The hardware element that provides high-speed caching, list processing, and locking functions in a sysplex.

CP. Central processor.

CPC. Central processor complex.

CPC operations management commands. A set of commands and responses for controlling the operation of System/390 CPCs.

CPC subset. All or part of a CPC. It contains the minimum resource to support a single control program.

CPCB. Command processor control block; an I/O operations internal control block that contains information about the command being processed.

CPU. Central processing unit. Deprecated term for processor.

cross-system coupling facility (XCF). XCF is a component of z/OS that provides functions to support cooperation between authorized programs running within a sysplex.

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

Customer Information Control System (CICS). A general-purpose transactional program that controls online communication between terminal users and a database for a large number of end users on a real-time basis.

customization dialogs. The customization dialogs are an ISPF application. They are used to customize the enterprise policy, like, for example, the enterprise resources and the relationships between resources, or the automation policy for systems in the enterprise. How to use these dialogs is described in Automation for z/OS Customizing and Programming.

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

D

DASD. Direct access storage device.

data services task (DST). The NetView subtask that gathers, records, and manages data in a VSAM file or a network device that contains network management information.

data set. The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

data set members. Members of partitioned data sets that are individually named elements of a larger file that can be retrieved by name.

DBCS. Double-byte character set.

DCCF. Disabled console communication facility.

DCF. Document composition facility.
DELAY Report. An RMF report that shows the activity of each job in the system and the hardware and software resources that are delaying each job.

Devices. You can see information about all devices (such as printers, tape or disk drives, displays, or communications controllers) attached to a particular switch, and control paths and jobs to devices.

DEVR Report. An RMF report that presents information about the activity of I/O devices that are delaying jobs.

dialog. Interactive 3270 panels.

direct access storage device (DASD). A device in which the access time is effectively independent of the location of the data; for example, a disk.

disabled console communication facility (DCCF). A z/OS component that provides limited-function console communication during system recovery situations.

display. (1) To present information for viewing, usually on the screen of a workstation or on a hardcopy device. (2) Deprecated term for panel.

disk operating system (DOS). (1) An operating system for computer systems that use disks and diskettes for auxiliary storage of programs and data. (2) Software for a personal computer that controls the processing of programs. For the IBM Personal Computer, the full name is Personal Computer Disk Operating System (PCDOS).

distribution manager. The component of the NetView program that enables the host system to use, send, and delete files and programs in a network of computers.

domain. (1) An access method and its application programs, communication controllers, connecting lines, modems, and attached workstations. (2) In SNA, a system services control point (SSCP) and the physical units (PUs), logical units (LUs), links, link stations, and associated resources that the SSCP can control by means of activation requests and deactivation requests.

double-byte character set (DBCS). A character set, such as Kanji, in which each character is represented by a 2-byte code.

DP enterprise. Data processing enterprise.

DSIPARM. This file is a collection of members of NetView’s customization.

DST. Data Services Task.

E

EBCDIC. Extended binary-coded decimal interchange code. A coded character set consisting of 8-bit coded characters.

ECB. Event control block. A control block used to represent the status of an event.

EMCS. Extended multiple console support.

enterprise. An organization, such as a business or a school, that uses data processing.

enterprise monitoring. Enterprise monitoring is used by SA z/OS to update the NetView Management Console (NMC) resource status information that is stored in the Resource Object Data Manager (RODM). Resource status information is acquired by enterprise monitoring of the Resource Measurement Facility (RMF) Monitor III service information at user-defined intervals. SA z/OS stores this information in its operational information base, where it is used to update the information presented to the operator in graphic displays.

entries. Resources, such as processors, entered on panels.

entry type. Resources, such as processors or applications, used for automation and monitoring.

environment. Data processing enterprise.

error threshold. An automation policy setting that specifies when SA z/OS should stop trying to restart or recover an application, subsystem or component, or offload a data set.

ESA. Enterprise Systems Architecture.

eServer. Processor family group designator used by the SA z/OS customization dialogs to define a target hardware as member of the zSeries or 390-CMOS processor families.

event. (1) In NetView, a record indicating irregularities of operation in physical elements of a network. (2) An occurrence of significance to a task; for example, the completion of an asynchronous operation, such as an input/output operation. (3) Events are part of a trigger condition, in a way that if all events of a trigger condition have occurred, a STARTUP or SHUTDOWN of an application is performed.

exception condition. An occurrence on a system that is a deviation from normal operation. SA z/OS monitoring highlights exception conditions and allows an SA z/OS enterprise to be managed by exception.

extended recovery facility (XRF). A facility that minimizes the effect of failures in z/OS, VTAM, the host processor, or high availability applications during sessions between high availability applications and designated terminals. This facility provides an alternate subsystem to take over sessions from the failing subsystem.
**F**

**fallback system.** See secondary system.

**field.** A collection of bytes within a record that are logically related and are processed as a unit.

**file manager commands.** A set of SA z/OS commands that read data from or write data to the automation control file or the operational information base. These commands are useful in the development of automation that uses SA z/OS facilities.

**focal point.** In NetView, the focal-point domain is the central host domain. It is the central control point for any management services element containing control of the network management data.

**focus host.** A processor with the role in the context of a unified system image

**focal point system.** (1) A system that can administer, manage, or control one or more target systems. There are a number of different focal point system associated with IBM automation products. (2) NMC focal point system. The NMC focal point system is a NetView system with an attached workstation server and LAN that gathers information about the state of the network. This focal point system uses RDOM to store the data it collects in the data model. The information stored in RDOM can be accessed from any LAN-connected workstation with NetView Management Console installed. (3) NPDA focal point system. This is a NetView system that collects all the NPDA alerts that are generated within your enterprise. It is supported by NetView. If you have SA z/OS installed the NPDA focal point system must be the same as your NMC focal point system. The NPDA focal point system is also known as the alert focal point system. (4) SA z/OS Processor Operations focal point system. This is a NetView system that has SA z/OS host code installed. The SA z/OS Processor Operations focal point system receives messages from the systems and operator consoles of the machines that it controls. It provides full systems and operations console function for its target systems. It can be used to IPL these systems. Note that some restrictions apply to the Hardware Management Console for an S/390 microprocessor cluster. (5) SA z/OS SDF focal point system. The SA z/OS SDF focal point system is an SA z/OS NetView system that collects status information from other SA z/OS NetViews within your enterprise. (6) Status focal point system. In NetView, the system to which STATMON, VTAM and NLDMS send status information on network resources. If you have a NMC focal point, it must be on the same system as the Status focal point. (7) Hardware Management Console. Although not listed as a focal point, the Hardware Management Console acts as a focal point for the console functions of an S/390 microprocessor cluster. Unlike all the other focal points in this definition, the Hardware Management Console runs on a LAN-connected workstation.

**frame.** For a System/390 microprocessor cluster, a frame contains one or two central processor complexes (CPCs), support elements, and AC power distribution.

**full-screen mode.** In NetView, a form of panel presentation that makes it possible to display the contents of an entire workstation screen at once. Full-screen mode can be used for fill-in-the-blanks prompting. Contrast with line mode.

**G**

**gateway session.** An NetView-NetView Task session with another system in which the SA z/OS outbound gateway operator logs onto the other NetView session without human operator intervention. Each end of a gateway session has both an inbound and outbound gateway operator.

**generic alert.** Encoded alert information that uses code points (defined by IBM and possibly customized by users or application programs) stored at an alert receiver, such as NetView.

**generic routines.** In SA z/OS, a set of self-contained automation routines that can be called from the NetView automation table, or from user-written automation procedures.

**group.** A collection of target systems defined through configuration dialogs. An installation might set up a group to refer to a physical site or an organizational or application entity.

**group entry.** A construct, created with the customization dialogs, used to represent and contain policy for a group.

**group entry type.** A collection of target systems defined through the customization dialog. An installation might set up a group to refer to a physical site or an organizational entity. Groups can, for example, be of type STANDARD or SYSPLEX.

**H**

**Hardware Management Console.** A console used by the operator to monitor and control a System/390 microprocessor cluster.

**Hardware Management Console Application (HWMCA).** A direct-manipulation object-oriented graphical user interface that provides single point of control and single system image for hardware elements. HWMCA provides customer grouping support, aggregated and real-time system status using colors, consolidated hardware messages support, consolidated operating system messages support, consolidated
service support, and hardware commands targeted at a single system, multiple systems, or a customer group of systems.

**heartbeat.** In SA z/OS, a function that monitors the validity of the status forwarding path between remote systems and the NMC focal point, and monitors the availability of remote z/OS systems, to ensure that status information displayed on the SA z/OS workstation is current.

**help panel.** An online panel that tells you how to use a command or another aspect of a product.

**hierarchy.** In the NetView program, the resource types, display types, and data types that make up the organization, or levels, in a network.

**high-level language (HLL).** A programming language that does not reflect the structure of any particular computer or operating system. For the NetView program, the high-level languages are PL/I and C.

**HLL.** High-level language.

**host system.** In a coupled system or distributed system environment, the system on which the facilities for centralized automation run. SA z/OS publications refer to target systems or focal-point systems instead of hosts.

**host (primary processor).** The processor at which you enter a command (also known as the issuing processor).

**HWMCA.** Hardware Management Console Application. Application for the graphic hardware management console that monitors and controls a central processor complex. It is attached to a target processor (a system 390 microprocessor cluster) as a dedicated system console. This microprocessor uses OCF to process commands.

**images.** A grouping of processors and I/O devices that you define. You can define a single-image mode that allows a multiprocessor system to function as one central processor image.

**IMS/VS.** Information Management System/Virtual Storage.

**inbound.** In SA z/OS, messages sent to the focal-point system from the PC or target system.

**inbound gateway operator.** The automation operator that receives incoming messages, commands, and responses from the outbound gateway operator at the sending system. The inbound gateway operator handles communications with other systems using a gateway session.

**Information Management System/Virtual Storage (IMS/VS).** A database/data communication (DB/DC) system that can manage complex databases and networks. Synonymous with IMS.

**INGEIO PROC.** The I/O operations default procedure name; part of the SYS1.PROCLIB.

**initial program load (IPL).** (1) The initialization procedure that causes an operating system to commence operation. (2) The process by which a configuration image is loaded into storage at the beginning of a workday or after a system malfunction. (3) The process of loading system programs and preparing a system to run jobs.

**initialize automation.** SA z/OS-provided automation that issues the correct z/OS start command for each subsystem when SA z/OS is initialized. The automation ensures that subsystems are started in the order specified in the automation control file and that prerequisite applications are functional.

**input/output support processor (IOSP).** The hardware unit that provides I/O support functions for the primary support processor and maintenance support functions for the processor controller.

**Interactive System Productivity Facility (ISP).** An IBM licensed program that serves as a full-screen editor and dialog manager. Used for writing application programs, it provides a means of generating standard screen panels and interactive dialogs between the application programmer and the terminal user.

**interested operator list.** The list of operators who are to receive messages from a specific target system.

**internal token.** A logical token (LTK); name by which the I/O resource or object is known; stored in IODF.

**IOCDS.** I/O configuration data set. The data set that describes the I/O configuration.

**I/O Ops.** I/O operations.

**IOSP.** Input/Output Support Processor.

**I/O operations.** The part of SA z/OS that provides you with a single point of logical control for managing connectivity in your active I/O configurations. I/O operations takes an active role in detecting unusual conditions and lets you view and change paths between a processor and an I/O device, using dynamic switching (the ESCON director). Also known as I/O Ops.

**I/O resource number.** Combination of channel path identifier (CHPID), device number, etc. See internal token.

**IPL.** Initial program load.

**ISA.** Industry Standard Architecture.
ISPF. Interactive System Productivity Facility.

ISPF console. From this 3270-type console you are logged onto ISPF to use the runtime panels for I/O operations and SA z/OS customization panels.

issuing host. See primary host; the base program at which you enter a command for processing.

J

JCL. Job control language.

JES. Job entry subsystem.

job. (1) A set of data that completely defines a unit of work for a computer. A job usually includes all necessary computer programs, linkages, files, and instructions to the operating system. (2) An address space.

job control language (JCL). A problem-oriented language designed to express statements in a job that are used to identify the job or describe its requirements to an operating system.

job entry subsystem (JES). A facility for spooling, job queuing, and managing I/O. In SA z/OS publications, JES refers to JES2 or JES3, unless distinguished as being either one or the other.

K

Kanji. An ideographic character set used in Japanese. See also double-byte character set.

L

LAN. Local area network.

line mode. A form of screen presentation in which the information is presented a line at a time in the message area of the terminal screen. Contrast with full-screen mode.

link. (1) In SNA, the combination of the link connection and the link stations joining network nodes; for example, a System/370 channel and its associated protocols, a serial-by-bit connection under the control of synchronous data link control (SDLC). (2) In SA z/OS, link connection is the physical medium of transmission.

link-attached. Describes devices that are physically connected by a telecommunication line. Contrast with channel-attached.

Linux for zSeries and S/390. UNIX-like open source operating system conceived by Linus Torvalds and developed across the internet.

local. Pertaining to a device accessed directly without use of a telecommunication line. Synonymous with channel-attached.

local area network (LAN). (1) A network in which a set of devices is connected for communication. They can be connected to a larger network. See also token ring. (2) A network in which communications are limited to a moderately-sized geographic area such as a single office building, warehouse, or campus, and that do not generally extend across public rights-of-way.

logical partition (LP). A subset of the processor hardware that is defined to support an operating system. See also logically partitioned (LPAR) mode.

logical switch number (LSN). Assigned with the switch parameter of the CHPID macro of the IOPC.

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 (PLU) and system services control point (SSCP).

logical unit (LU) 6.2. A type of logical unit that supports general communications between programs in a distributed processing environment. LU 6.2 is characterized by (a) a peer relationship between session partners, (b) efficient use of a session for multiple transactions, (c) comprehensive end-to-end error processing, and (d) a generic application program interface (API) consisting of structured verbs that are mapped into a product implementation. Synonym for advanced program-to-program communications (APPC).

logically partitioned (LPAR) mode. A central processor mode that enables an operator to allocate system processor hardware resources among several logical partitions. Contrast with basic mode.

LOGR. The sysplex logger.

LP. Logical partition.

LPAR. Logically partitioned (mode).

LU. Logical unit.

LU-LU session. In SNA, a session between two logical units (LUs) in an SNA network. It provides communication between two end users, or between an end user and an LU services component.

LU 6.2. Logical unit 6.2.
LU 6.2 session. A session initiated by VTAM on behalf of an LU 6.2 application program, or a session initiated by a remote LU in which the application program specifies that VTAM is to control the session by using the APPCCMD macro.

M

MAT. Deprecated term for NetView Automation Table.

MCA. Micro Channel* architecture.

MCS. Multiple console support.

Member. A specific function (one or more modules/routines) of a multisystem application that is defined to XCF and assigned to a group by the multisystem application. A member resides on one system in the sysplex and can use XCF services to communicate (send and receive data) with other members of the same group.

Message automation table (MAT). Deprecated term for NetView Automation Table.

Message class. A number that SA z/OS associates with a message to control routing of the message. During automated operations, the classes associated with each message issued by SA z/OS are compared to the classes assigned to each notification operator. Any operator with a class matching one of the message’s classes receives the message.

Message forwarding. The SA z/OS process of sending messages generated at an SA z/OS target system to the SA z/OS focal-point system.

Message group. Several messages that are displayed together as a unit.

Message monitor task. A task that starts and is associated with a number of communications tasks. Message monitor tasks receive inbound messages from a communications task, determine the originating target system, and route the messages to the appropriate target control tasks.

Message processing facility (MPF). A z/OS table that screens all messages sent to the z/OS console. The MPF compares these messages with a customer-defined list of messages on which to automate, suppress from the z/OS console display, or both, and marks messages to automate or suppress. Messages are then broadcast on the subsystem interface (SSI).

Message suppression. The ability to restrict the amount of message traffic displayed on the z/OS console.

Micro Channel architecture. The rules that define how subsystems and adapters use the Micro Channel bus in a computer. The architecture defines the services that each subsystem can or must provide.

Microprocessor. A processor implemented on one or a small number of chips.

Migration. Installation of a new version or release of a program to replace an earlier version or release.

MP. Multiprocessor.

MPF. Message processing facility.

MPFLSTSA. The MPFLST member that is built by SA z/OS.

Multiple Virtual Storage (MVS). An IBM licensed program. MVS, which is the predecessor of OS/390, is an operating system that controls the running of programs on a System/390 or System/370 processor. MVS includes an appropriate level of the Data Facility Product (DFP) and Multiple Virtual Storage/Enterprise Systems Architecture System Product Version 5 (MVS/ESA SP5).

Multiprocessor (MP). A CPC that can be physically partitioned to form two operating processor complexes.

Multisystem application. An application program that has various functions distributed across z/OS images in a multisystem environment.

Multisystem environment. An environment in which two or more z/OS images reside in one or more processors, and programs on one image can communicate with programs on the other images.

MVS. Multiple Virtual Storage, predecessor of z/OS.

MVS image. A single occurrence of the MVS/ESA operating system that has the ability to process work.

MVS/JES2. Multiple Virtual Storage/Job Entry System 2. A z/OS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system. In an installation with more than one processor, each JES2 processor independently controls its job input, scheduling, and output processing.

MVS/ESA. Multiple Virtual Storage/Enterprise Systems Architecture.

N

NAU. (1) Network accessible unit. (2) Network addressable unit.

NCCF. Network Communications Control Facility.

NCP. (1) Network Control Program (IBM licensed program). Its full name is Advanced Communications Function for the Network Control Program. Synonymous with ACF/NCP. (2) Network control program (general term).
NetView. An IBM licensed program used to monitor a network, manage it, and diagnose network problems. NetView consists of a command facility that includes a presentation service, command processors, automation based on command lists, and a transaction processing structure on which the session monitor, hardware monitor, and terminal access facility (TAF) network management applications are built.

network accessible unit (NAU). A logical unit (LU), physical unit (PU), control point (CP), or system services control point (SSCP). It is the origin or the destination of information transmitted by the path control network. Synonymous with network addressable unit.

network addressable unit (NAU). Synonym for network accessible unit.

NetView automation procedures. A sequence of commands, packaged as a NetView command list or a command processor written in a high-level language. An automation procedure performs automation functions and runs under the NetView program.

NetView automation table (AT). A table against which the NetView program compares incoming messages. A match with an entry triggers the specified response. SA z/OS entries in the NetView automation table trigger an SA z/OS response to target system conditions. Formerly known as the message automation table (MAT).

NetView Command list language. An interpretive language unique to NetView that is used to write command lists.

NetView (NCCF) console. A 3270-type console for NetView commands and runtime panels for system operations and processor operations.


NetView hardware monitor. The component of NetView that helps identify network problems, such as hardware, software, and microcode, from a central control point using interactive display techniques. Formerly called network problem determination application.

NetView log. The log in which NetView records events pertaining to NetView and SA z/OS activities.

NetView message table. See NetView automation table.

NetView Management Console (NMC). A function of the NetView program that provides a graphic, topological presentation of a network that is controlled by the NetView program. It provides the operator different views of a network, multiple levels of graphical detail, and dynamic resource status of the network. This function consists of a series of graphic windows that allows you to manage the network interactively. Formerly known as the NetView Graphic Monitor Facility (NGMF).

NetView-NetView task (NNT). The task under which a cross-domain NetView operator session runs. Each NetView program must have a NetView-NetView task to establish one NNT session. See also operator station task.

NetView-NetView Task session. A session between two NetView programs that runs under a NetView-NetView Task. In SA z/OS, NetView-NetView Task sessions are used for communication between focal point and remote systems.

NetView paths via logical unit (LU 6.2). A type of network-accessible port (VTAM connection) that enables end users to gain access to SNA network resources and communicate with each other. LU 6.2 permits communication between processor operations and the workstation.

network. (1) An interconnected group of nodes. (2) In data processing, a user application network. See SNA network.

Network Communications Control Facility (NCCF). The operations control facility for the network. NCCF consists of a presentation service, command processors, automation based on command lists, and a transaction processing structure on which the network management applications NLDM and NPDA are built. NCCF is a precursor to the NetView command facility.

Network Control Program (NCP). An IBM licensed program that provides communication controller support for single-domain, multiple-domain, and interconnected network capability. Its full name is Advanced Communications Function for the Network Control Program.

Networking NetView. In SA z/OS the NetView that performs network management functions, such as managing the configuration of a network. In SA z/OS it is common to also route alerts to the Networking NetView.

Network Problem Determination Application (NPDA). An NCCF application that helps you identify network problems, such as hardware, software, and microcode, from a central control point using interactive display methods. The alert manager for the network. The precursor of the NetView hardware monitor.

NGMF. Deprecated term for NetView Management Console.

NGMF focal-point system. Deprecated term for NMC focal point system.

NIP. Nucleus initialization program.
**NMC focal point system.** See focal point system

**NMC workstation.** The NMC workstation is the primary way to dynamically monitor SA z/OS systems. From the windows, you see messages, monitor status, view trends, and react to changes before they cause problems for end users. You can use multiple windows to monitor multiple views of the system.

**NNT.** NetView-NetView task.

**notification message.** An SA z/OS message sent to a human notification operator to provide information about significant automation actions. Notification messages are defined using the customization dialogs.

**notification operator.** A NetView console operator who is authorized to receive SA z/OS notification messages. Authorization is made through the customization dialogs.

**NPDA.** Network Problem Determination Application.

**NPDA focal-point system.** See focal-point system.

**NTRI.** NCP/token-ring interconnection.

**nucleus initialization program (NIP).** The program that initializes the resident control program; it allows the operator to request last-minute changes to certain options specified during system generation.

**O**

**objective value.** An average Workflow or Using value that SA z/OS can calculate for applications from past service data. SA z/OS uses the objective value to calculate warning and alert thresholds when none are explicitly defined.

**OCA.** In SA z/OS, operator console A, the active operator console for a target system. Contrast with OCB.

**OCB.** In SA z/OS, operator console B, the backup operator console for a target system. Contrast with OCA.

**OCF.** Operations command facility.

**OCF-based processor.** A central processor complex that uses an operations command facility for interacting with human operators or external programs to perform operations management functions on the CPC.

**OPC/A.** Operations Planning and Control/Advanced.


**operating system (OS).** Software that controls the execution of programs and that may provide services such as resource allocation, scheduling, input/output control, and data management. Although operating systems are predominantly software, partial hardware implementations are possible. (T)

**operations.** The real-time control of a hardware device or software function.

**operations command facility (OCF).** A facility of the central processor complex that accepts and processes operations management commands.

**Operations Planning and Control/Advanced (OPC/A).** A set of IBM licensed programs that automate, plan, and control batch workload. OPC/A analyzes system and workload status and submits jobs accordingly.

**Operations Planning and Control/ESA (OPC/ESA).** A set of IBM licensed programs that automate, plan, and control batch workload. OPC/ESA analyzes system and workload status and submits jobs accordingly. The successor to OPC/A.

**operator.** (1) A person who keeps a system running. (2) A person or program responsible for managing activities controlled by a given piece of software such as z/OS, the NetView program, or IMS. (3) A person who operates a device. (4) In a language statement, the lexical entity that indicates the action to be performed on operands.

**operator console.** (1) A functional unit containing devices that are used for communications between a computer operator and a computer. (T) (2) A display console used for communication between the operator and the system, used primarily to specify information concerning application programs and I/O operations and to monitor system operation. (3) In SA z/OS, a console that displays output from and sends input to the operating system (z/OS, LINUX, VM, VSE). Also called operating system console. In the SA z/OS operator commands and configuration dialogs, OC is used to designate a target system operator console.

**operator station task (OST).** The NetView task that establishes and maintains the online session with the network operator. There is one operator station task for each network operator who logs on to the NetView program.

**operator view.** A set of group, system, and resource definitions that are associated together for monitoring purposes. An operator view appears as a graphic display in the graphic interface showing the status of the defined groups, systems, and resources.

**OperatorView entry.** A construct, created with the customization dialogs, used to represent and contain policy for an operator view.

**OS.** Operating system.
**z/OS component.** A part of z/OS that performs a specific z/OS function. In SA z/OS, component refers to entities that are managed by SA z/OS automation.

**z/OS subsystem.** Software products that augment the z/OS operating system. JES and TSO/E are examples of z/OS subsystems. SA z/OS includes automation for some z/OS subsystems.

**z/OS system.** A z/OS image together with its associated hardware, which collectively are often referred to simply as a system, or z/OS system.

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

**OST.** Operator station task.

**outbound.** In SA z/OS, messages or commands from the focal-point system to the target system.

**outbound gateway operator.** The automation operator that establishes connections to other systems. The outbound gateway operator handles communications with other systems through a gateway session. The automation operator sends messages, commands, and responses to the inbound gateway operator at the receiving system.

**P**

**page.** (1) The portion of a panel that is shown on a display surface at one time. (2) To transfer instructions, data, or both between real storage and external page or auxiliary storage.

**panel.** (1) A formatted display of information that appears on a terminal screen. Panels are full-screen 3270-type displays with a monospaced font, limited color and graphics. (2) By using SA z/OS panels you can see status, type commands on a command line using a keyboard, configure your system, and pass thru to other consoles. See also help panel. (3) In computer graphics, a display image that defines the locations and characteristics of display fields on a display surface. Contrast with screen.

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

**parameter.** (1) A variable that is given a constant value for a specified application and that may denote the application. (2) An item in a menu for which the user specifies a value or for which the system provides a value when the menu is interpreted. (3) Data passed to a program or procedure by a user or another program, namely as an operand in a language statement, as an item in a menu, or as a shared data structure.

**partition.** (1) A fixed-size division of storage. (2) In VSE, a division of the virtual address area that is available for program processing. (3) On an IBM Personal Computer fixed disk, one of four possible storage areas of variable size; one can be accessed by DOS, and each of the others may be assigned to another operating system.

**partitionable CPC.** A CPC that can be divided into 2 independent CPCs. See also physical partition, single-image mode, MP, side.

**partitioned data set (PDS).** A data set in direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

**passive monitoring.** In SA z/OS, the receiving of unsolicited messages from z/OS systems and their resources. These messages can prompt updates to resource status displays. See also active monitoring.

**PCE.** Processor controller. Also known as the “support processor” or “service processor” in some processor families.

**PDB.** Policy Database

**PDS.** Partitioned data set.

**physical partition.** Part of a CPC that operates as a CPC in its own right, with its own copy of the operating system.

**physical unit (PU).** In SNA, the component that manages and monitors the resources (such as attached links and adjacent link stations) of a node, as requested by a system services control point (SSCP) through an SSCP-PU session. An SSCP activates a session with the physical unit to indirectly manage, through the PU, resources of the node such as attached links.

**physically partitioned (PP) configuration.** A mode of operation that allows a multiprocessor (MP) system to function as two or more independent CPCs having separate power, water, and maintenance boundaries. Contrast with single-image (SI) configuration.

**POI.** Program operator interface.

**policy.** The automation and monitoring specifications for an SA z/OS enterprise. See System Automation for z/OS Defining Automation Policy.

**policy database.** The database where the automation policy is recorded. Also known as the PDB.

**POR.** Power-on reset.

**port.** (1) System hardware to which the I/O devices are attached. (2) On an ESCON switch, a port is an addressable connection. The switch routes data through the ports to the channel or control unit. Each port has a name that can be entered into a switch matrix, and you
can use commands to change the switch configuration. (3) An access point (for example, a logical unit) for data entry or exit. (4) A functional unit of a node through which data can enter or leave a data network. (5) In data communication, that part of a data processor that is dedicated to a single data channel for the purpose of receiving data from or transmitting data to one or more external, remote devices. (6) Power-on reset (POR) (7) A function that re-initializes all the hardware in a CPC and loads the internal code that enables the CPC to load and run an operating system.

**PP.** Physically partitioned (configuration).

**PPT.** Primary POI task.

**primary host.** The base program at which you enter a command for processing.

**primary POI task (PPT).** The NetView subtask that processes all unsolicited messages received from the VTAM program operator interface (POI) and delivers them to the controlling operator or to the command processor. The PPT also processes the initial command specified to execute when NetView is started and timer request commands scheduled to execute under the PPT.

**primary system.** A system is a primary system for an application if the application is normally meant to be running there. SA z/OS starts the application on all the primary systems defined for it.

**problem determination.** The process of determining the source of a problem; for example, a program component, machine failure, telecommunication facilities, user or contractor-installed programs or equipment, environment failure such as a power loss, or user error.

**processor controller.** Hardware that provides support and diagnostic functions for the central processors.

**processor operations.** The part of SA z/OS that monitors and controls processor (hardware) operations. Processor operations provides a connection from a focal-point system to a target system. Through NetView on the focal-point system, processor operations automates operator and system consoles for monitoring and recovering target systems. Also known as ProcOps.

**processor operations control file.** Named by your system programmer, this file contains configuration and customization information. The programmer records the name of this control file in the processor operations file generation panel ISQDGP01.

**Processor Resource/Systems Manager (PR/SM).** The feature that allows the processor to use several operating system images simultaneously and provides logical partitioning capability. See also *LPAR*.

**ProcOps.** Processor operations.

**ProcOps Service Machine (PSM).** The PSM is a CMS user on a VM host system. It runs a CMS multitasking application that serves as "virtual hardware" for ProcOps. ProcOps communicates via the PSM with the VM guest systems that are defined as target systems within ProcOps.

**product automation.** Automation integrated into the base of SA z/OS for the products DB2, CICS, IMS, OPC (formerly called *features*).

**program to program interface (PPI).** A NetView function that allows user programs to send or receive data buffers from other user programs and to send alerts to the NetView hardware monitor from system and application programs.

**protocol.** In SNA, the meanings of, and the sequencing rules for, requests and responses used for managing the network, transferring data, and synchronizing the states of network components.

**proxy resource.** A resource defined like an entry type APL representing a processor operations target system.

**PR/SM.** Processor Resource/Systems Manager.

**PSM.** ProcOps Service Machine.

**PU.** Physical unit.

**R**

**remote system.** A system that receives resource status information from an SA z/OS focal-point system. An SA z/OS remote system is defined as part of the same SA z/OS enterprise as the SA z/OS focal-point system to which it is related.

**requester.** A requester is a workstation software, which enables users to log on to a domain, that is, to the server(s) belonging to this domain, and use the resources in this domain. After the log on to a domain, users can access the shared resources and use the processing capability of the server(s). Because the bigger part of shared resources is on the server(s), users can reduce hardware investment.

**resource.** (1) Any facility of the computing system or operating system required by a job or task, and including main storage, input/output devices, the processing unit, data sets, and control or processing programs. (2) In NetView, any hardware or software that provides function to the network. (3) In SA z/OS, any z/OS application, z/OS component, job, device, or target system capable of being monitored or automated through SA z/OS.

**Resource Access Control Facility (RACF).** A program that can provide data security for all your resources. RACF protects data from accidental or deliberate unauthorized disclosure, modification, or destruction.
resource group. A physically partitioned portion of a processor. Also known as a side.

Resource Monitoring Facility (RMF) Monitor III. A program that measures and reports on the availability and activity of system hardware and software resources, such as processors, devices, storage, and address spaces. RMF can issue online reports about system performance problems as they occur.

Resource Object Data Manager (RODM). A data cache manager designed to support process control and automation applications. RODM provides an in-memory data cache for maintaining real-time data in an address space that is accessible by multiple applications. RODM also allows an application to query an object and receive a rapid response and act on it.

resource token. A unique internal identifier of an ESCON resource or resource number of the object in the IODF.

restart automation. SA z/OS-provided automation that monitors subsystems to ensure that they are running. If a subsystem fails, SA z/OS attempts to restart it according to the policy in the automation control file.

Restructured Extended Executor (REXX). An interpretive language used to write command lists.

return code. A code returned from a program used to influence the issuing of subsequent instructions.

REXX. Restructured Extended Executor.

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

RMF. Resource Measurement Facility.

RODM. Resource Object Data Manager.

S

SAF. Security Authorization Facility.

SA z/OS. System Automation for z/OS

SA z/OS customization dialogs. An ISPF application through which the SA z/OS policy administrator defines policy for individual z/OS systems and builds automation control data and RODM load function files.

SA z/OS customization focal point system. See focal point system.

SA z/OS data model. The set of objects, classes and entity relationships necessary to support the function of SA z/OS and the NetView automation platform.

SA z/OS enterprise. The group of systems and resources defined in the customization dialogs under one enterprise name. An SA z/OS enterprise consists of connected z/OS systems running SA z/OS.

SA z/OS focal point system. See focal point system.

SA z/OS policy. The description of the systems and resources that make up an SA z/OS enterprise, together with their monitoring and automation definitions.

SA z/OS policy administrator. The member of the operations staff who is responsible for defining SA z/OS policy.

SA z/OS satellite. If you are running two NetViews on an z/OS system to split the automation and networking functions of NetView, it is common to route alerts to the Networking NetView. For SA z/OS to process alerts properly on the Networking NetView, you must install a subset of SA z/OS code, called an SA z/OS satellite on the Networking NetView.

SA z/OS SDF focal point system. See focal point system.

SCA. In SA z/OS, system console A, the active system console for a target hardware. Contrast with SCB.

SCB. In SA z/OS, system console B, the backup system console for a target hardware. Contrast with SCA.

screen. Deprecated term for display panel.

screen handler. In SA z/OS, software that interprets all data to and from a full-screen image of a target system. The interpretation depends on the format of the data on the full-screen image. Every processor and operating system has its own format for the full-screen image. A screen handler controls one PS/2 connection to a target system.

SDF. Status Display Facility.

SDLC. Synchronous data link control.

SDSE. System Display and Search Facility.

secondary system. A system is a secondary system for an application if it is defined to automation on that system, but the application is not normally meant to be running there. Secondary systems are systems to which an application can be moved in the event that one or more of its primary systems are unavailable. SA z/OS does not start the application on its secondary systems.

server. A server is a workstation that shares resources, which include directories, printers, serial devices, and computing powers.

Glossary 397
**service language command (SLC).** The line-oriented command language of processor controllers or service processors.

**service processor (SVP).** The name given to a processor controller on smaller System/370 processors.

**service period.** Service periods allow the users to schedule the availability of applications. A service period is a set of time intervals (service windows), during which an application should be active.

**service threshold.** An SA z/OS policy setting that determines when to notify the operator of deteriorating service for a resource. See also alert threshold and warning threshold.

**session.** In SNA, a logical connection between two network addressable units (NAUs) that can be activated, tailored to provide various protocols, and deactivated, as requested. Each session is uniquely identified in a transmission header by a pair of network addresses identifying the origin and destination NAUs of any transmissions exchanged during the session.

**session monitor.** The component of the NetView program that collects and correlates session-related data and provides online access to this information. The successor to NLDN.

**shutdown automation.** SA z/OS-provided automation that manages the shutdown process for subsystems by issuing shutdown commands and responding to prompts for additional information.

**side.** A part of a partitionable CPC that can run as a physical partition and is typically referred to as the A-side or the B-side.

**Simple Network Management Protocol (SNMP).** An IP-based industry standard protocol to monitor and control resources in an IP network.

**single image.** A processor system capable of being physically partitioned that has not been physically partitioned. Single-image systems can be target hardware processors.

**single-image (SI) mode.** A mode of operation for a multiprocessor (MP) system that allows it to function as one CPC. By definition, a uniprocessor (UP) operates in single-image mode. Contrast with physically partitioned (PP) configuration.

**SLC.** Service language command.

**SMP/E.** System Modification Program Extended.

**SNA.** Systems Network Architecture.

**SNA network.** In SNA, the part of a user-application network that conforms to the formats and protocols of systems network architecture. It enables reliable transfer of data among end users and provides protocols for controlling the resources of various network configurations. The SNA network consists of network addressable units (NAUs), boundary function components, and the path control network.

**SNMP.** Simple Network Management Protocol (a TCP/IP protocol). A protocol that allows network management by elements, such as gateways, routers, and hosts. This protocol provides a means of communication between network elements regarding network resources.

**solicited message.** An SA z/OS message that directly responds to a command. Contrast with unsolicited message.

**SSCP.** System services control point.

**SSI.** Subsystem interface.

**start automation.** SA z/OS-provided automation that manages and completes the startup process for subsystems. During this process, SA z/OS replies to prompts for additional information, ensures that the startup process completes within specified time limits, notifies the operator of problems, if necessary, and brings subsystems to an UP (or ready) state.

**startup.** The point in time at which a subsystem or application is started.

**status.** The measure of the condition or availability of the resource.

**status focal-point system.** See focal—point system.

**status display facility (SDF).** The system operations part of SA z/OS that displays status of resources such as applications, gateways, and write-to-operator messages (WTORs) on dynamic color-coded panels. SDF shows spool usage problems and resource data from multiple systems.

**steady state automation.** The routine monitoring, both for presence and performance, of subsystems, applications, volumes and systems. Steady state automation may respond to messages, performance exceptions and discrepancies between its model of the system and reality.

**structure.** A construct used by z/OS to map and manage storage on a coupling facility. See cache structure, list structure, and lock structure.

**subgroup.** A named set of systems. A subgroup is part of an SA z/OS enterprise definition and is used for monitoring purposes.

**SubGroup entry.** A construct, created with the customization dialogs, used to represent and contain policy for a subgroup.
**subsystem.** (1) A secondary or subordinate system, usually capable of operating independent of, or asynchronously with, a controlling system. (2) In SA z/OS, an z/OS application or subsystem defined to SA z/OS.

**subsystem interface.** The z/OS interface over which all messages sent to the z/OS console are broadcast.

**support element.** A hardware unit that provides communications, monitoring, and diagnostic functions to a central processor complex (CPC).

**support processor.** Another name given to a processor controller on smaller System/370 processors; see service processor.

**SVP.** Service processor.

**switches.** ESCON directors are electronic units with ports that dynamically switch to route data to I/O devices. The switches are controlled by I/O operations commands that you enter on a workstation.

**switch identifier.** The switch device number (swchdevn), the logical switch number (LSN) and the switch name

**symbolic destination name (SDN).** Used locally at the workstation to relate to the VTAM application name.

**synchronous data link control (SDLC).** A discipline for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or nonswitched links. The configuration of the link connection may be point-to-point, multipoint, or loop. SDLC conforms to subsets of the Advanced Data Communication Control Procedures (ADCCP) of the American National Standards Institute and High-Level Data Link Control (HDLC) of the International Standards Organization.

**SYSINFO Report.** An RMF report that presents an overview of the system, its workload, and the total number of jobs using resources or delayed for resources.

**SysOps.** System operations.

**sysplex.** A set of z/OS systems communicating and cooperating with each other through certain multisystem hardware components (coupling devices and timers) and software services (couple data sets).

In a sysplex, z/OS provides the coupling services that handle the messages, data, and status for the parts of a multisystem application that has its workload spread across two or more of the connected processors, sysplex timers, coupling facilities, and couple data sets (which contains policy and states for automation).

A parallel sysplex is a sysplex that includes a coupling facility.

**sysplex application group.** A sysplex application group is a grouping of applications that can run on any system in a sysplex.

**sysplex couple data set.** A couple data set that contains sysplex-wide data about systems, groups, and members that use XCF services. All z/OS systems in a sysplex must have connectivity to the sysplex couple data set. See also couple data set.

**Sysplex Timer.** An IBM unit that synchronizes the time-of-day (TOD) clocks in multiple processors or processor sides. External Time Reference (ETR) is the z/OS generic name for the IBM Sysplex Timer (9037).

**system.** In SA z/OS, system means a focal point system (z/OS) or a target system (MVS, VM, VSE, LINUX, or CF).

**System Automation for z/OS.** The full name for SA z/OS.

**System Automation for OS/390.** The full name for SA OS/390, the predecessor to System Automation for z/OS.

**system console.** (1) A console, usually having a keyboard and a display screen, that is used by an operator to control and communicate with a system. (2) A logical device used for the operation and control of hardware functions (for example, IPL, alter/display, and reconfiguration). The system console can be assigned to any of the physical displays attached to a processor controller or support processor. (3) In SA z/OS, the hardware system console for processor controllers or service processors of processors connected using SA z/OS. In the SA z/OS operator commands and configuration dialogs, SC is used to designate the system console for a target hardware processor.

**System Display and Search Facility (SDSF).** An IBM licensed program that provides information about jobs, queues, and printers running under JES2 on a series of panels. Under SA z/OS you can select SDSF from a pull-down menu to see the resources’ status, view the z/OS system log, see WTOR messages, and see active jobs on the system.

**System entry.** A construct, created with the customization dialogs, used to represent and contain policy for a system.

**System Modification Program/Extended (SMP/E).** An IBM licensed program that facilitates the process of installing and servicing an z/OS system.

**system operations.** The part of SA z/OS that monitors and controls system operations applications and subsystems such as NetView, SDSF, JES, RME, TSO, RODM, ACF/VTAM, CICS, IMS, and OPC. Also known as SysOps.
system services control point (SSCP). In SNA, the focal point within an SNA network for managing the configuration, coordinating network operator and problem determination requests, and providing directory support and other session services for end users of the network. Multiple SSCPs, cooperating as peers, can divide the network into domains of control, with each SSCP having a hierarchical control relationship to the physical units and logical units within its domain.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks.

System/390 microprocessor cluster. A configuration that consists of central processor complexes (CPCs) and may have one or more integrated coupling facilities.

terminal access facility (TAF). (1) A NetView function that allows you to log onto multiple applications either on your system or other systems. You can define TAF sessions in the SA z/OS customization panels so you don’t have to set them up each time you want to use them. (2) In NetView, a facility that allows a network operator to control a number of subsystems. In a full-screen or operator control session, operators can control any combination of subsystems simultaneously.

terminal emulation. The capability of a microcomputer or personal computer to operate as if it were a particular type of terminal linked to a processing unit to access data.

threshold. A value that determines the point at which SA z/OS automation performs a predefined action. See alert threshold, warning threshold, and error threshold.

time of day (TOD). Typically refers to the time-of-day clock.

Time Sharing Option (TSO). An optional configuration of the operating system that provides conversational 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 command that issues a command or command processor (list of commands) at a specified time or time interval.

TOD. Time of day.

token ring. A network with a ring topology that passes tokens from one attaching device to another; for example, the IBM Token-Ring Network product.

TP. Transaction program.

transaction program. In the VTAM program, a program that performs services related to the processing of a transaction. One or more transaction programs may operate within a VTAM application program that is using the VTAM application program interface (API). In that situation, the transaction program would request services from the applications program using protocols defined by that application program. The application program, in turn, could request services from the VTAM program by issuing the APPCCMD macro instruction.

transitional automation. The actions involved in starting and stopping subsystems and applications that
have been defined to SA z/OS. This can include issuing commands and responding to messages.

**translating host.** Role played by a host that turns a resource number into a token during a unification process.

**trigger.** Triggers, in combination with events and service periods, are used to control the starting and stopping of applications in a single system or a parallel sysplex.

**TSO.** Time Sharing Option.

**TSO console.** From this 3270-type console you are logged onto TSO or ISPF to use the runtime panels for I/O operations and SA z/OS customization panels.

**TSO/E.** TSO Extensions.

**U**

**UCB.** The unit control block; an MVS/ESA data area that represents a device and that is used for allocating devices and controlling I/O operations.

**unsolicited message.** An SA z/OS message that is not a direct response to a command. Contrast with solicited message.

**user task.** An application of the NetView program defined in a NetView TASK definition statement.

**Using.** An RMF Monitor III definition. Jobs getting service from hardware resources (processors or devices) are using these resources. The use of a resource by an address space can vary from 0% to 100% where 0% indicates no use during a Range period, and 100% indicates that the address space was found using the resource in every sample during that period. See also Workflow.

**V**

**view.** In the NetView Graphic Monitor Facility, a graphical picture of a network or part of a network. A view consists of nodes connected by links and may also include text and background lines. A view can be displayed, edited, and monitored for status information about network resources.

**Virtual Storage Extended (VSE).** An IBM licensed program whose full name is Virtual Storage Extended/Advanced Function. It is an operating system that controls the execution of programs.

**Virtual Telecommunications Access Method (VTAM).** An IBM licensed program that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability. Its full name is Advanced Communications Function for the Virtual Telecommunications Access Method. Synonymous with ACF/VTAM.

**VM/ESA.** Virtual Machine/Enterprise Systems Architecture.

**VM Second Level Systems Support.** With this function, Processor Operations is able to control VM second level systems (VM guest systems) in the same way that it controls systems running on real hardware.

**volume.** A direct access storage device (DASD) volume or a tape volume that serves a system in an SA z/OS enterprise.

**volume entry.** A construct, created with the customization dialogs, used to represent and contain policy for a volume.

**volume group.** A named set of volumes. A volume group is part of a system definition and is used for monitoring purposes.

**volume group entry.** An construct, created with the customization dialogs, used to represent and contain policy for a volume group.

**Volume Workflow.** The SA z/OS Volume Workflow variable is derived from the RMF Resource Workflow definition, and is used to measure the performance of volumes. SA z/OS calculates Volume Workflow using:

\[
\text{Volume } \text{Using} = \frac{\text{accumulated}}{\text{Using}} \times 100
\]

\[
\text{Workflow } \% = \frac{\text{accumulated} + \text{accumulated}}{\text{Using} \times \text{Delay}}
\]

The definition of **Using** is the percentage of time when a job has had a request accepted by a channel for the volume, but the request is not yet complete.

The definition of **Delay** is the delay that waiting jobs experience because of contention for the volume. See also **Address Space Workflow**.

**VSE.** Virtual Storage Extended.

**VTAM.** Virtual Telecommunications Access Method.

**W**

**warning threshold.** An application or volume service value that determines the level at which SA z/OS changes the associated icon in the graphic interface to the warning color. See **alert threshold**.

**workflow.** See **Address Space Workflow** and **Volume Workflow**.
**workstation.** In SA z/OS workstation means the *graphic workstation* that an operator uses for day-to-day operations.

**write-to-operator (WTO).** A request to send a message to an operator at the z/OS operator console. This request is made by an application and is handled by the WTO processor, which is part of the z/OS supervisor program.

**write-to-operator-with-reply (WTOR).** A request to send a message to an operator at the z/OS operator console that requires a response from the operator. This request is made by an application and is handled by the WTO processor, which is part of the z/OS supervisor program.

**WTO.** Write-to-Operator.

**WTOR.** Write-to-Operator-with-Reply.

**WWV.** The US National Institute of Standards and Technology (NIST) radio station that provides standard time information. A second station, known as WWVB, provides standard time information at a different frequency.

**X**

**XCF.** Cross-system coupling facility.

**XCF couple data set.** The name for the sysplex couple data set prior to MVS/ESA System Product Version 5 Release 1. See also *sysplex couple data set.*

**XCF group.** A set of related members that a multisystem application defines to XCF. A member is a specific function, or instance, of the application. A member resides on one system and can communicate with other members of the same group across the sysplex.

**XRF.** Extended recovery facility.

**Numerics**

**390-CMOS.** Processor family group designator used in the SA z/OS processor operations documentation and in the online help to identify any of the following S/390 CMOS processor machine types: 9672, 9674, 2003, 3000, or 7060. SA z/OS processor operations uses the OCF facility of these processors to perform operations management functions. See *OCF-based processor.*
Index

Special characters
- *EMPTY* sample database 300
- *MULTISYS* sample database 300
- *SAP_HA* sample database 301
- *SYSPLEX* sample database 301
- *WS401_HA* sample database 301

A
- ABENDS (JES3) 194
- ACF 10, 257
- ACFFQRY 73
- ACORESTART 167
- Action Message Retention Facility 9
- actions on policy databases 295
- Adjacent Domain policy definition 284
- ADJACENT NETVIEW policy item 283
- alternate adjacent policy definition 284
- alternate configuration support 378
- alternate configurations 378
- alternate data sets 97
- AMC 10
- AMC file 315, 324
- AMRF 9
- anchors
  - heartbeat 98
  - in RODM 98
  - Lost&Found 98
  - specifying 99
  - sysplex 98
- AOCCLONE IDs 42
- AOCCLONESec system automation symbols 42
- AOFADMON 118
- AOFAJMON, see INGPJMON 119
- AOFAPMON 118
- AOFARCAT 275
- AOFATMON 118
- AOFCPSM 118
- AOFMINORCHECK 39
- AOFOFF sample 275
- AOFOFFPO sample updating 316
- AOFOUXMON 118
- AOFWRKxx 274
- AOFWRKxx operators 278
- APPC automation 9
- APPLGROUP INFO policy item 128, 130
- application (entry type) 4
- application class 147
- Application Defaults 6
- Application Defaults policy item 111
- APPLICATION DEFAULTS policy item 111
- Application Defaults policy object 272
- application group 4
  - nature 126
  - nature of 50
- Sysplex Application Group 49
- System Application Group 49
- application group resource name 50
- APPLICATION GROUPS policy item 93, 110
- APPLICATION INFO policy item 152, 154
- application instance 147
- application monitor routine 118, 148, 158
- Application policy object 145
  - creating a new 145
  - policy items 151
- application resource name 48
- application resources
  - automatically linking 54
  - generating 52
- application type 147
- ApplicationGroup policy object 125
  - creating a new 125
  - policy items 128
- APPLICATIONS policy item 128, 130
- ARM
  - MOVE group for 354
  - ARM element name 333
  - ASCB chaining 148
  - with application automation 161
- ASCH automation 9
- assigning
  - clone IDs 42
  - system automation symbols 42
- AT build, automated
  - concurrent user access 367
- ATTEND TIME policy item 261
- AUTBASE 274
- AUTCON 275
- AUTGSS 275
- AUTHB 275
- AUTHW1 275
- AUTJES 274
- AUTLOG 274
- AUTMON 275
- AUTMSG 274
- AUTNET1, AUTNET2 274
- auto message classes 83
- Auto Msg Classes 83
- AUTO MSG CLASSES policy item 87
- Auto Operators 6
  - Auto Operators policy item 111
  - AUTO OPERATORS policy item 111
  - Auto Operators policy object 273
- AUTO1 274
- AUTO2 274
- automated AT build
  - concurrent user access 367
- automated function 169, 179, 182, 183
- Automatic Restart Manager 333
  - defining element names 156
  - element name 353
  - MOVE group for 354
  - using clone values 156
- automation
  - goal driven 23
- automation (continued)
  - table policy definition 119
- Automation automation flag 35
- automation concepts 23
- automation configuration 11, 324
- ACF 324
- AMC file 324
- automation control file 324
- automation manager configuration file 324
- automation console policy item 110
- AUTOMATION CONSOLE policy item 110, 116
- automation control file 10, 324
- building 317
- distributing 324
- migrating 296, 299
- automation flag
  - Automation 35
  - Initstart 36
  - Recovery 36
  - Restart 37
  - Shutdown 36
  - Start 36
  - types 35
- automation flags 34
  - automation manager flags 38
  - extended 154
  - global automation flag 38
  - hold flag 38
  - minor resource flags 39
  - resource status 38
  - setting 37
  - setting application defaults 273
- automation flags policy definition 152
- AUTOMATIONフラグ policy item 152, 162, 270, 271, 273
- automation info 152
- AUTOMATION INFO policy item 152, 157
- automation manager configuration file 10, 315, 324
  - building 317
  - distributing 324
- automation manager flags 38
- automation manager global automation flag 38
- automation name 50, 127
- automation networks 347
  - definition 282
- automation operators 273
- automation operators policy definition 274
- AUTOMATIONオプション policy item 271
- automation policy
  - building 10
  - distributing 10
  - automation procedure 9
  - automation resource name 127
  - automation setup policy definition 110
documents, licensed xviii

drain, JES2 190
DSIOFF data set 274
DSIOFF member 316
DSIPARM
description 388
dual NetView 115

E

E-T DATA policy item 293
element name 353
Automatic Restart Manager 156
element names
in Automatic Restart Manager 353
type 3
enterprise definition 11, 82
Enterprise policy object 82, 221
policy items 82
entry name creation 336
entry name selection
concurrent user access 366
entry type
introduction 3
short description 3
entry type selection
concurrent user access 366
environment definition, system defaults 271
ENVIRONMENT policy item 271, 272
error messages
BUILD 322
error thresholds 31
event 39, 40
events 4
scenario 70
events and conditions 40
Events policy object
creating a new 221
policy items 222
exit names policy definition 120
extended automation flags 154
extended recovery facilities 247
external shutdown 161
external startup 161
EXTERNALLY 134
EXTSTART status 353

F

FALLBACK 188
FALLBACK status
support 188
using 188
fast path 13
feature keywords
cross reference with customization
dialog 360
FINAL shutdown 180
focal point
name 87
focal point NetView domains policy
definition 284
focal point services 282
focal point system definition 348
FORCE shutdown 180
FORCEDOWN 134
FORWARD policy item 284
frequent threshold 32
FULL SESSIONS policy item 284
functional names for auto operators 274
G

GAT&DOMID 275
gateway
inbound 349
outbound 349
gateway monitor time policy
definition 118
gateway policy definition 284
GATEWAY policy item 284
gateway sessions
defining 349
GATOPER 275
GENERATED RESOURCES policy
item 129, 143, 154, 188
generic routine 9
global automation flag 38
goal driven automation 23
graphic workstation description 394
graphical interface 8
GROUP INFO policy item 92, 94
Group policy object 90
new, creating 90
grouping
scenario 24
support 23
grouping resources 8
groups 4
groups and resources 23
GSNOPER 275
SHUTDOWN 279
STARTUP 279
GUESTS AND SYSTEMS policy
item 236
H

HASPARENT 134
HASPARENT relationship 29
HASPASSIVEPARENT 135
HASPASSIVEPARENT relationship 29
HBOPER 275
HEALTHSTATE policy item 238
heartbeat
explanation 107
heartbeat interval 106
missing heartbeat delay 107
heartbeat anchor 98
heartbeat interval 106
help information 369
hold flag 38
HWMCA
description 390
HWOPER 275
I

IMMED shutdown 180
importing
policy database data 327
IMS automation 247
IMS CONTROL policy item 208, 209
IMS extended recovery facilities 247
IMS RESOURCE NAMES policy
item 250
IMS STATE/ACTION policy item 248
IMS state/action table 247
IMS STATUS FILE policy item 249
IMS subsystems
policy items 208
IMS XRF resources 247, 250
IMS XRF RESOURCES policy item 112
IMS XRF status files 247, 249
inbound gateway 349
includes 6
INCLUDES policy item 112
Includes policy object 292
infrequent threshold 32
ING,CUSTOM,AOFTABL 314, 375
INGAUTO 38
INGDLG 12
INGEO 390
INGEVENT 39
INGMTSYS 119
INGPJMONG 119
INGPOST command 98
INGREQ 23
INGSEND PARMS policy item 82, 85
INGSET 23
INGTHRES 32
INGXSG 98
initialized target system 121
INITSEL 12
Initstart automation flag 36
interested operator list 122
invoking the customization dialog 12
IPL
MVS target system 123
VM target system 124
IPL INFO 113
IPL INFO policy item 122
ISPF temporary data sets 376
ISQMTSYS, 119
IXCARM macro invocations 353
IXCMAIPU 353
J

JES2
automation procedures 9
recovery parameters 191
spool recovery
automation 9
establishing 190
from full condition 190
JES2 DRAIN 190, 193
JES2 SPOOLFULL 190
JES2 SPOOLSHORT 190, 192
JES2 subsystems
policy items 190
JES3
ABENDS 194
automation procedures 9
MONITOR 194
recovery parameters 195
message processing (continued)
    scenario 74
    user defined data 172
    with user-defined data 74
message retrieval tool, LookAt xviii
messages
    policy definition 153
    response 123, 124
MESSAGES USER DATA policy item 153, 165, 263
MIGRATE command
    Policy Data Base Selection 296
    migrating
        automation control file 296, 329
migration
    entry name creation 336
    include files 339
    job for batch migration 345
    product automation definition 339
    selective migration 331
migration functions 327
    minor resource flags 39
    as subsidiary to major resource 264
    for applications 153
MINOR RESOURCE FLAGS policy item 153, 186, 264, 269
minor resources 269
    AOFMINORCHECK 39
missing heartbeat 107
    processing 107
    missing heartbeat delay 107
    missing heartbeat processing 107
MONITOR (JES3) 194
MONITOR INFO policy item 237, 238
monitor resource 5
Monitor Resource
    policy object 237
monitoring
tape mounts 260
MONITORING PERIOD policy item 245, 246
monitoring periods for CICS 246
Monitoring, Sysplex Timer 96
MONOPER 275
MOVE 50, 126
MOVE group 60
MOVE group for ARM 354
MOVED 188
moved and restructured data 355
MOVED status
    Automatic Restart Manager 353
    automation 353
    using 188
MPF definitions 83, 89
MPFLSTSA file 315
MPFLSTSA member 10
    building 317
    distributing 324
MVCOPER 274
MVS Component 5
MVS Component policy item
    to choose sets to be part of system object 111
MVS COMPONENT policy item 111
MVS Component policy object
dump data sets 9
    using 262
MVS console policy definition 278
MVS job 148
MVS messages, responding to 263
MVS Target System IPL Information panel 123
MVS_COMP DEFAULTS policy item 111
MVS_COMP Defaults policy object 269
MVS_CONS policy 275
MVS_DUMP
    MVS Component policy object 262
thresholds policy 263

N
naming restrictions 318
nature
    BASIC 50
    MOVE 50
    SERVER 50
nature of an application group 50
nature of application groups
    BASIC 126
    MOVE 126
    SERVER 126
NBUF policy definition 265
NETOPER 274
NetView
    domain name 115
    nomination before BUILD 10
    operator definition file 316
    operator profile 316
    Operators 278
    policy item 110
    RMTCMD command 283
    specifying domain name 115
NetView automation table 10, 315
    building 317
    distributing 324
    updating 76
    NETVIEW policy item 109, 115
    Network 6
    Network policy item 111
    NETWORK policy item 111
    Network policy object 282
    networks automation
        definition process 347
        NMC DEFINITIONS panel 93
        NMC DEFINITIONS policy item 93
        NMC focal point system 389
        NNT sessions 6
        and minor resources 110
        definition 287
        NNT SESSIONS policy item 110
        NNT Sessions policy object 286
        NONMVS job 148
        NORM shutdown 180
        NORMCONS policy definition 267
        notification forwarding 348
        notification operator 280
        notification operators 111, 280
        Notify Operators 6
        Notify Operators policy item 111
        NOTIFY OPERATORS policy item 111
        notify operators policy object 279

K
keywords for features 360

L
licensed documents xviii
link definitions (CICS) 243, 244
link monitor periods (CICS) 243, 246
LINK TO CLASS policy item 151
LINK TO INSTANCES policy item 152
load profile name 121
LOCATE 297
logging policy database
    modifications 303
LOGOPER 274
LOGREC
    automation policy definition 9
    thresholds policy 263
LookAt message retrieval tool xviii
Lost&Found anchor 98
LPARS AND SYSTEMS policy item 233

M
MAKEAVAILABLE 133
MAKEUNAVAILABLE 133
MCFLAGS policy definition 117
MEMBER OF policy item 154
    for application groups 144
    for application groups 129
    for applications 189
message
    forwarding path, defining 347
    message class 280
    message classes 83, 281, 282
message IDs
    reserved 167
message prefix 158
message processing 166
    CMD processing 168
    code processing 170
    reply processing 169

406 System Automation for z/OS: System Automation for z/OS V2R3.0 Defining Automation Policy
O
OPC automation 251
OPC CONTROL policy item 217
OPC controller details 251, 253
policy items 253
OPC special resources 251, 255
OPC subsystems
policy items 217
OPC system details 251, 252
OPC SYSTEM DETAILS policy item 112
OPC workstation domains 251, 256
OPCA DOMAINDID policy item 256
OPCA PCS policy item 253, 254
OPCAO ENVIRON policy item 252
operator
definition file 316
profile 316
operator console
interested operator 122
OPERATORS policy item 274
outbound gateway 349
autotask, defining 349
output data set 320
overview 1

P
panelid 7
password mask 285
PLEXOPER 275
Policy Data Base Selection panel 297
policy database 11
adding to index 297
available actions 295
converting 331
creating 297
creating, new 297
data, importing 327
initial conversion of 355
logging modifications 303
reports 307
samples 300
sending to remote nodes 377
using existing 295
policy database selection
concurrent user access 365
policy definitions
concurrent user access 365
policy item 3, 7
APPLICATION GROUPS 99
AUTOMATIC OPTIONS 271
COPY 19
DESCRIPTION 18
ENVIRONMENT 272
WHERE USED 20
Policy Item
IEADMCCXX SYMBOLS 103
JOB DEFINITIONS 100
JOB/ASID DEFINITIONS 101
LOCAL PAGE DATA SET 99
NMC DEFINITIONS 98
RESOURCE DEFINITIONS 102
policy items
create or modify 371
general policy items 18
policy object 3, 7
policy object (continued)
creating, new 14
policy objects
Application 145
Application Defaults 272
ApplicationGroup 125
Auto Operators 273
copying 19
copying definitions 19
deleting 16
Enterprise 82, 221
Group 90
Includes 292
Monitor Resource 237
MVS Component 262
MVSCOMP Defaults 269
Network 282
NNT Sessions 286
Notify Operators 279
Processor 229
Product Automation 241
Resident CLISTS 288
Service Periods 223
Status Details 289
SubGroup 103
Sysplex Application Group 125, 128
System 105
System Application Group 128
System Defaults 270
Tape Attendance 260
Timeout Settings 258
Timers 256
Triggers 226
User E-T Pairs 292
PolicyDB 298, 329
POSTOPER 275
POSTSTART 178
PPIOPER 275
preference value
default 127
scenarios 142, 143
preference values 141
preferences 65
PREPAVAILABLE 134
PREPUNAVAILABLE 134
primary data sets 97
primary JES policy definition 117
primary menu
customization dialog 2, 12
primary system association 187
priority of requests 226
processor control file 315
PROCESSOR INFO policy item 231
PROCESSOR OPERATIONS policy item 120
PROCESSOR OPS INFO policy item 83, 86
processor policy item 113
PROCESSOR policy item 112
Processor policy object 229
creating a new 229
policy items 230
policy items for PSM 234
processors 5
profilename 42
product automation 5, 396
CICS 243
product automation (continued)
DB2 242
IMS 247
Product Automation
policy object 241
profile information before BUILD 10
propagation of requests 29
proxy resource 119
PSM 230
PSM INFORMATION policy item 234
R
RECOPER 275
recovering WTOBUF 265
Recovery automation flag 36
relationship condition 137
relationship type 133
relationships 132, 165
policy item for applications 152
relationships concept 28
RELATIONSHIPS policy item 129, 132,
152, 165, 238
reply processing 169
reply shutdown processing 183
REPORT 296
reports
about ApplicationGroups 307
about Applications 307
about generated resources 307
request 23
calculation of schedules 28
INGREQ 23
priority 23
request propagation 29
reserved message IDs 167
Resident CLISTS 6
RESIDENT CLISTS policy item
with minor resources 111
RESIDENT CLISTS policy item 110
Resident CLISTS policy object 288
resource
dependent 28, 129, 132, 153, 238
supporting 28, 129, 132, 153, 238
resource name
WLM 157
resource states
automation flags 38
description 38
list 38
SDF 39
RESOURCE THRESHOLDS policy item 200, 205, 209, 243
CICS 203
DB2 207
resources
explanation 46
grouping 8
scenario 25, 45
resources and groups 23
RESOURCES policy item 129, 139
Restart automation flag 37
restart option policy definition 160
RMF 9
RMTCMD command 283
RODM
anchors 98

Index 407
S
SA z/OS enterprise components 81
SA z/OS topology manager 326
SAF ENVIRONMENT policy item 285
sample policy databases 300
*EMPTY 300
*MULTISYS 300
*SAP_HA 301
*SYSPLEX 301
*WAS401_HA 301
satisfactory target 140
schedule automatic calculation 28
using 69
schedule override 69
Screens policy item 111
SCREENS policy item 111
SDF
automation control file entry 290
highlight 291
priority policy definition 291
root name policy definition 120
SDF DETAILS policy item 290
secondary system association 187
select migration objects 331
SEND COMMAND OPERS policy item 82, 83
sending policy databases 377
SERVER 50
SERVER group 67
service period 28, 129, 225, 238
linking to application 152
using 69
SERVICE PERIOD policy item 129, 131, 152, 164, 238
service periods 4
Service Periods policy object 223
service window 28, 225
SERVICE WINDOW policy item 224
SESSIONS policy item 287
setting automation flags 37
settings menu 370
Settings option 369
SHUTDOWN
GSOPER 279
Shutdown automation flag 36
Shutdown condition 39, 40, 41, 227, 229
SHUTDOWN condition for a trigger 228
shutdown delay policy definition 161
SHUTDOWN policy item 153, 179
shutdown processing command 181, 182
reply 183
SHUTOPER 275
single NetView 115
SMF data sets 9
SMFDUMP
MVS Component policy object 262
thresholds policy 263
SPECIAL RESOURCE policy item
OPC 255
SPECIAL RESOURCES policy item 253
SPOOLFULL 190
SPOOLFULL (JESS) 194
SPOOLSHORT 190
SPOOLSHORT(JESS) 194
standard Groups 92
policy items 92
start
start on IPL 159
start on recycle 160
start timeout policy definition 160
Start automation flag 36
starting the customization dialog 12
STARTUP 178
GSOPER 279
startup condition 39, 40, 41, 227
STARTUP condition for a trigger 228
startup condition list 229
startup phases
POSTSTART 177
PRESTART 177
STARTUP 177
STARTUP policy item 153, 175
STATE ACTION TABLE policy item 200, 203, 209
state/action table for CICS 243
STATE/ACTION TABLE policy item 243
state/action tables (CICS) 243
state/action tables (IMS) 247
state/action tables for CICS 243
Status Detail 6
Status Details
display facility 289
policy object 289
status forwarding path 107
SUBGROUP INFO policy item 104
SubGroup policy object 103
creating a new 103
policy items 104
subgroups 4
SUBGROUPS policy item 92, 94
subsystem
shutdown 153
startup 152, 153
subsystem name 146
subtype 148
supporting resource 28, 129, 132, 153, 238
symbol descriptions 83, 88
SYSLOG
data sets 9
MVS Component policy object 262
SYSLOG thresholds policy definition 263
SYSNAME naming restrictions 318
sysname policy item 119
SYSOPER 274
SysOps NetView 115
SysOps NetView Network Name 115
sysplex
creating 46, 47
sysplex anchor 98
Sysplex Application Group 49, 125, 126
Sysplex Application Group (continued)
generating Sysplex Application Group resources 51
Sysplex Application Group policy object 125
dsplex group 90
dsplex Groups 92
policy items 92
Sysplex Policy Definition panel 93
SYSPLEX policy item 93, 95
Sysplex Timer Monitoring 96
System Application Group 49, 126
generating System Application Group resources 55
System Application Groups 128
system association policy definition 154
SYSTEM ASSOCIATION policy item 154, 187
system automation symbols 42
system categories 187
system console interested operator 122
System Defaults 6
System Defaults policy item and minor resources 111
SYSTEM DEFAULTS policy item 111
System Defaults policy object 270
SYSTEM INFO policy item 109, 113
system monitor time policy definition 118
system operations configuration files 315, 324
building 317
distributing 324
System policy object 105
creating a new 105
policy items 108
SYSTEMS policy item 95
for Groups 92
for SubGroups 105
systems, definitions 4
T
TAF 283
TAF full-screen sessions
defining 350
TAF full-screen sessions policy definition 284
tape attendance 5
Tape Attendance policy item with minor resources 111
TAPE ATTENDANCE policy item 111
Tape Attendance policy object 260
tape mount, monitoring 260
tape policy 260
target
docal point name 87
TARGET SYSTEM INFO 113
TARGET SYSTEM INFO policy item 121
target system information
initialized target system 121
load profile name 121
time zones 121
target system time zones 121
task
message monitor 87
task (continued)
target control 87
Terminal Access Facility 283, 350
threshold
critical 31, 32
for applications 33
for MVS Components 34
for z/OS components 31
frequent 32
infrequent 32
MVS/COMP 32
THRESHOLD policy items 264
thresholds
setting up 31
THRESHOLDS policy item 153, 185, 270, 271, 273
TIMEAUTO 38
timeout settings 5
Timeout Settings policy object 258
TIMEOUT VALUES policy item 259
timeout variable 258
timers 5
TIMERS policy item 257
Timers policy object 256
To Domain policy definition 284
TRANSIENT job 148
trigger 39, 40
link to an application group 129
linking to application 152
restriction 42
TRIGGER policy item 129, 131, 152, 164
trigger policy object 4
triggers
scenario 70
triggers and conditions 40
Triggers policy object 226
TSO/E automation 9

W
Where Used panel 17, 21
WHERE USED policy item 20
WLM resource name 157
workstation domains
OPC 256
WORKSTATION DOMAINS policy item 112
WTOBUF
automation policy definition 9
recovery 262
routines 264
WTOBUF RECOVERY policy item 264, 265

X
XCF group name
INGXSG, default 98
INGXSGxy 98
XCFOPER 275
XCFOPER2 275
XCFOPER3 275
XRF 247
XRF STATUS FILES policy item 112

Z
z/OS job name defined 149

U
UCB (unit control block) 401
unset condition 222, 223
updating
AOOPFPO sample 316
updating the NetView automation table 76
user defined data processing 172
User E-T Pairs 6
description 110
policy object 292
scenario 72
USER E-T PAIRS policy item 110
user-written functions 377
USS CONTROL policy item 219
USS subsystems
policy items 218

V
VIEW 297
VM Target System IPL Information panel 124
VTAM automation 9
VTAMLST 284
Readers’ Comments — We’d Like to Hear from You

System Automation for z/OS
Defining Automation Policy
Version 2 Release 3

Publication No. SC33-7039-06

Overall, how satisfied are you with the information in this book?

<table>
<thead>
<tr>
<th>Overall satisfaction</th>
<th>Very Satisfied</th>
<th>Satisfied</th>
<th>Neutral</th>
<th>Dissatisfied</th>
<th>Very Dissatisfied</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
</tbody>
</table>

How satisfied are you that the information in this book is:

<table>
<thead>
<tr>
<th></th>
<th>Very Satisfied</th>
<th>Satisfied</th>
<th>Neutral</th>
<th>Dissatisfied</th>
<th>Very Dissatisfied</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accurate</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>Complete</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>Easy to find</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>Easy to understand</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>Well organized</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>Applicable to your tasks</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
</tbody>
</table>

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? □ Yes □ No

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.

Name

Company or Organization

Phone No.

Address
Readers’ Comments — We’d Like to Hear from You

IBM Deutschland Entwicklung GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany
Spine information: