

System Automation for OS/390



Defining Automation Policy

Version 2 Release 1

System Automation for OS/390



Defining Automation Policy

Version 2 Release 1

Note!

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

Fifth Edition (December 2001)

This edition applies to System Automation for OS/390 (Program Number 5645-006) Version 2, Release 1, 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

Internet

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

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

© Copyright International Business Machines Corporation 1996, 2001. All rights reserved.

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

Contents

Figures	vii
--------------------------	------------

Tables	xi
-------------------------	-----------

Notices	xiii
--------------------------	-------------

Web Site Disclaimer	xiii
Programming Interface Information	xiii
Trademarks	xiv

About This Book	xv
----------------------------------	-----------

Who Should Use This Book	xv
SA OS/390 User Groups	xv
Where to Find More Information	xvi
The System Automation for OS/390 Library	xvi
Related Product Information	xvii
Using LookAt to look up message explanations	xix
Accessing licensed books on the Web	xx

Summary of Changes	xxi
-------------------------------------	------------

What's New in This Book for SA OS/390 2.1	xxi
What's New in SA OS/390 2.1	xxi
Notes on Terminology	xxiii

Chapter 1. Overview	1
--------------------------------------	----------

How to Define Automation Policy: An Outline	1
Customization Dialog Introduction	2
Policy Objects and Policy Items	6
Considerations for the SA OS/390 Graphical Interface	9
Grouping Resources Logically	9
Considerations for Automation	10
Building and Distributing SA OS/390 Automation Policy	11

Chapter 2. Using the Customization Dialog	13
--	-----------

How to Start the Customization Dialog	14
How to Use a Fast Path	16
How to Use the Customization Dialog	16
Concurrent User Access	17
How to Cancel Input on a Panel	17
Creating a New Policy Database	17
Using Sample SA OS/390 Policy Databases	20
Using Existing Databases	22
Policy Information	24
How to Create New Policy Objects	25
Delete Option on All Entry Name Selection Panels	26
Policy Items Found on Many Selection Panels	27

Chapter 3. Automation Concepts	31
---	-----------

Goal Driven Automation	31
Grouping Support	31
A Grouping Scenario	32

Automatic Calculation of Schedules (Service Periods)	36
Adaptive Scheduling	36
Relationship Support	36
How SA OS/390 Uses Error Thresholds	39
Setting up a Threshold	39
Automation Flags	42
Types of Automation Flags	43
Setting Automation Flags	45
Automation Manager Flags	46
Relationship Between Automation Flags and Resource Status	46
Automation Flags and Minor Resource Flags	47
Event and Trigger Support	47

Chapter 4. Scenarios on How to Use the Customization Dialog	51
--	-----------

How to Work with Resources	51
Step 1: How to Create the Policy Objects that Make Up the Sysplex	51
Step 2: How to Create the Applications	53
Step 3: How to Create the Application Groups	54
How to Use Schedules (Service Periods)	77
Schedule Override	77
Early Start	78
How to Use Triggers and Events	78
How to Use User E-T Pairs	80
Step 1: Specifying User Data in the Customization Dialog	80
Step 2: Access the User Data	81
How to Implement Message Processing with User-Defined Data	82
Step 1: How to Specify the User-Defined Data	82
Step 2: How to Specify the Commands	84
Step 3: How to Update the NetView Message Automation Table	84
Step 4: How to Write the REXX Exec	84

Chapter 5. Defining Automation Policy	87
--	-----------

Enterprise Policy Object	88
Policy Items for an Enterprise	88
More About Policy Item SEND COMMAND OPERS	89
More About Policy Item PROCESSOR OPS INFO	91
Group Policy Object	92
Creating a New Group	92
Policy Items for Standard Groups	93
Policy Items for Sysplex Groups	94
More About Policy Item SYSPLEX	95
SubGroup Policy Object	98
Creating a New SubGroup	98
Policy Items for SubGroups	99
System Policy Object	99
Creating a New System	99
Policy Items for Systems	104

More About Policy Item NETVIEW	108
More About Policy Item AUTOMATION CONSOLE	109
More About Policy Item AUTOMATION SETUP	111
More About Policy Item NOTIFY OPERATORS	113
More About Policy Item PROCESSOR OPERATIONS	117
ApplicationGroup Policy Object	132
Creating a New ApplicationGroup	133
Policy Items for ApplicationGroups	135
More About Policy Item APPLGROUP INFO	137
More About Policy Item APPLICATIONS	138
More About Policy Item RESOURCES	139
More About Policy Item GENERATED RESOURCES	140
More About Policy Item MEMBER OF for ApplicationGroups	141
Application Policy Object	141
Creating a New Application	141
Policy Items for Applications	147
More About Policy Item AUTOMATION INFO	150
More About Policy Item RELATIONSHIPS	157
More About Policy Item MESSAGES/USER DATA	163
More About Policy Item STARTUP	169
More About Policy Item SHUTDOWN	173
More About Policy Item MEMBER OF for Applications.	177
More About Policy Item SYSTEM ASSOCIATION.	177
Defining JES Subsystems	179
Policy Items for JES2 Subsystems.	181
Policy Items for JES3 Subsystems.	184
Policy Items for CICS Subsystems	185
Policy Items for DB2 Subsystems	190
Policy Items for IMS Subsystems	191
Policy Items for OPC Subsystems.	199
Events Policy Object	201
Creating a New Event	201
Policy Items for Events	202
More About Policy Item CONDITION	202
Service Periods Policy Object	203
Triggers Policy Object	205
More About Policy Item CONDITION	206
Processor Policy Object	208
Creating a New Processor	209
Policy Items for Processors	210
CP States for Screen-Oriented Processors	211
More about the CONNECTIONS Policy Items	213
Channel Path IDs	216
Communications Path Policy Object	218
Creating a New Communications Path	218
Policy Items for Communications Path	218
Processor Operations PS/2 Communications Path	219
NetView Connection	227
Communications Path Commands	228
Product Automation Policy Object	229
Defining Automation for DB2 Components	229
Defining Automation for CICS Components	230
Defining Automation for IMS Components	235

Defining Automation for OPC Components	240
Timers Policy Object	246
Timeout Settings Policy Object.	248
Tape Attendance Policy Object.	250
MVS Component Policy Object	252
More About the THRESHOLD Policy Items	254
More About the WTOBUF RECOVERY Policy Item	255
More About Policy Item MESSAGES/USER DATA	258
More About Policy Item MINOR RESOURCE FLAGS	259
MVSCOMP Defaults Policy Object	260
System Defaults Policy Object	261
Application Defaults Policy Object	262
Auto Operators Policy Object	263
More About the OPERATORS Policy Item.	264
Structure of AOFWRKxx Operators	269
Auto Msg Classes Policy Object	269
Network Policy Object	270
NNT Sessions Policy Object	273
Resident CLISTs Policy Object.	275
Status Details Policy Object.	276
More About Policy Item SDF DETAILS.	277
Communications Task Policy Object	279
Creating a New Communications Task	279
Policy Items for Communications Task	280
More About Policy Item CONTROL PARAMETERS	280
More About Policy Item CONTROLLED PS/2S	282
Communications Task Commands	282
Includes Policy Object	286
User E-T Pairs Policy Object	286
More About Policy Item E-T DATA	286

Chapter 6. Building and Distributing Control Files. 289

How to Define Data Sets for Build Processing	289
How to Build Data and Control Files from the Build Functions Menu	289
Building the Processor Control File	290
Usage Notes.	291
LAN Passthru Function	292
Building an Automation Control File and an Automation Manager Configuration File	292
System Operations Control Files Build Options	293
Distributing the System Operations Control Files	298
Building NetView and VTAM Information Used by Processor Operations	301
Generate NetView and VTAM Information	301
Building PS/2 Profile Information	303
What Are PS/2 Profiles?.	303
Contents of PS/2 Profiles	304
Generating PS/2 Profiles	305
Customizing the PS/2 Computer for the Standalone Function	306

Chapter 7. Migration Functions. 309

Migrate Automation Control File	309
Entry Name Creation.	314

Migration of Include Files and Product Automation Definitions	317
Generating a Job for Batch Migrate	324
Merge TSCF Configuration	324

Chapter 8. Automating Networks 327

Automation Network Definition Process	327
Defining a Focal Point System	328
Defining Gateway Sessions	329
Defining Adjacent NetView Paths	334
Defining Automatically Initiated TAF Full-Screen Sessions	335
Defining Automatically Initiated User NNT Sessions	336

Chapter 9. How System Operations Coordinates with Automatic Restart Manager 339

Defining an ARM Element Name	339
Defining a MOVE Group for Automatic Restart Manager	340

Chapter 10. Conversion Function 341

Deleted Entry Types and Policy Items	341
Initial Policy Database Conversion	341
Converted Data	342
Selective Policy Database Conversion	348

Appendix A. Concurrent Customization by Multiple Users 349

Concurrent User Access Capabilities	349
Policy Definitions Usage	349
Policy Data Base Selection	349
Entry Type Selection	350
Entry Name Selection	350
Changing Links	350

Appendix B. More about Using the Customization Dialog 351

Getting Help	351
------------------------	-----

Changing Your Customization Dialog Environment	351
Extending Policy Definitions	353
Defining Data Sets for Batch Processing	356
Define ISPF Temporary Data Sets for Batch Build	357
Creating Reports about Policy Databases	358
User-written Functions	360
Alternate Configuration Support	361
How to Define Alternate Configurations for Systems	362
How to Build Alternate Configurations	362
Restrictions	362

Appendix C. DB2 Automation for System Automation for OS/390 365

Product Overview	365
Line Mode Functions	365
Planning, Installation, and Defining Automation Policy	366
Planning Requirements	366
IMS	366
CICS	366
Installation	366
Automation Control File (ACF)	366
NetView Message Automation Table (AT)	366
Defining Automation Policy	367
Creating Your PDB	367
Tailoring Your DB2 ACF Entries	367
DB2 Automated Functions – Line Command Functions	370
Command Handler	371
Command Requests	372
Maintenance Start	373
Terminate Threads	376
Start/Stop Tablespace	378
Event-Driven Functions	379
Connection Monitoring	380
Critical Event Monitoring	383

Glossary 389

Index 409

Figures

1. Panel Flow of the Customization Dialog	7	40. Define New Entry Input Panel for WBSRVAPG	75
2. Entry Type Selection Panel	8	41. Final Select Resources Panel for WBSRVAPG	77
3. Entry Name Selection Panel for Groups	8	42. Scenario with Triggers, Events and Service	
4. Policy Selection Panel for Sysplex Groups	9	Period	80
5. Customization Dialog Primary Menu	14	43. UET Entry-Type Selection Panel.	81
6. Policy Data Base Selection Panel	18	44. UET Keyword-Data Specification Panel	81
7. Create a New Policy Database Panel	18	45. Message Processing Panel.	83
8. Entry Type Selection Panel	25	46. User Defined Data Panel	83
9. Request a List of Instances	27	47. CMD Processing Panel.	84
10. Links of Application Class to Instances	27	48. Entry Type Selection Panel	87
11. Where Used Panel	29	49. Policy Selection Panel for Enterprises	88
12. Example of Aggregating Applications into		50. Operator Profile for Sending Commands	89
Application Groups	32	51. Processor Operations Information Panel	91
13. Application Groups	34	52. Define New Entry Panel for a Sysplex Group	93
14. Grouping of Applications into Nested		53. Policy Selection Panel for Groups	94
ApplicationGroups	35	54. Policy Selection Panel for Sysplex Groups	95
15. Relationships and Conditions	37	55. Sysplex Policy Definition Panel for Sysplex	
16. Relationship in a Sysplex	38	Groups	96
17. Relationship between Applications and		56. Define New Entry Panel for a Subgroup	98
Application Groups	38	57. Policy Selection Panel for Subgroups	99
18. Thresholds Specification Panel	40	58. Define New Entry Panel for a System	100
19. Thresholds Definition Panel	41	59. Policy Selection Panel for Systems	104
20. Component Definition: MVSDUMP Panel	42	60. NetView Information Panel	109
21. Flag Automation Specification Panel	43	61. Environment Definition: CONSOLE (Route	
22. How a Trigger uses Events	48	Codes) Panel.	110
23. Triggers, Events and Conditions.	49	62. Environment Setup Panel	111
24. Current Configuration with Sysplex		63. Operator Notification Panel	114
Application Group CICS AOR	57	64. Operator Notification Message Classes Panel	115
25. Current Configuration with Applications		65. Processor Operations Target Definition Panel	
CICSAOR1...3	59	for Screen-Oriented Processors	117
26. Resources for the CICS AOR Sysplex		66. Target System Basic Characteristics Panel for	
Application Group	59	9021, 9121, 3090, and 308x Processors	118
27. Resources for the CICS Sysplex Application		67. Target System Basic Characteristics Panel for	
Group	61	OCF-Based Processors in (Mode	119
28. Resources for the CICS Sysplex Application		68. Target System Basic Characteristics Panel for	
Group	62	an OCF-Based Processor in Basic Mode	120
29. Current Configuration with Application group		69. Target System Basic Characteristics Panel for	
DB2	64	9674	121
30. Current Configuration with applications linked		70. Target System Basic Characteristics Panel for	
to DB2	65	4381	122
31. Select APG-Resource for Grouping Panel for		71. Target System IPL Information Panel for	
DB2 Application group.	65	Screen-Oriented Processors	123
32. Select Resources Panel for DB2 Application		72. MVS Target System IPL Information Panel	124
group	66	73. VM Target System IPL Information Panel	125
33. Select Resources Panel for Application group		74. Target System Connections Panel	126
NET	67	75. Processor Operations Target System Time	
34. Select Resources Panel for DWHAPG with		Zone Panel	127
valid APG-Resources	68	76. Processor Operations Interested Operator List	
35. Select Resources Panel for DWHAPG2	70	Panel	128
36. Select Resources Panel for DWHAPG2	71	77. Processor Operations CP States Panel for	
37. Select Resources Panel for DWHAPG with		9021, 9121, and 3090 Processors	129
valid APG-Resources	72	78. CP LPAR States Panel.	130
38. Select Resources Panel for DWHAPG with		79. Processor Operations CP States Panel for a	
Selected Preferences.	73	308x Processor	131
39. Select Resources Panel for DWHAPG with		80. Processor Operations CP States Panel for a	
Converted Preferences	74	4381 Processor	132

81. Define New Entry Panel for a Sysplex Application Group	133	133. Target Hardware CP States Panel	212
82. Policy Selection Panel for System Application Groups	136	134. Connections	213
83. Application Group Information Panel	138	135. Operator Console Connections	215
84. Applications for ApplicationGroup Panel for System Application Groups	138	136. CHPID Assignments	217
85. Generated Resources Panel	140	137. Defining a New SA OS/390 Communications Path Entry	218
86. Member of Panel for ApplicationGroups	141	138. Policy Selection Panel for PS/2 Communications Path.	219
87. Define New Entry Panel for Applications	142	139. Policy Selection Panel for NetView Connection Communications Path	219
88. Applications Defined as Instance Objects and of Type STANDARD	147	140. PS/2 Characteristics	220
89. Application Automation Definition Panel	151	141. Processor Operations Port Configurations Panel	221
90. Relationship Selection List Panel	157	142. 3270 Port Description Panel	222
91. Define Relationship Panel	158	143. ASCII Port Description Panel	226
92. Example for Strong Chaining	161	144. NetView Connection Path Definition Panel	228
93. Message Processing Panel	163	145. Entry type selections for Product Automation Panel	229
94. CMD Processing Panel	165	146. Policy Selection Panel for DB2 System Defaults	230
95. Reply Processing Panel	167	147. Policy Selection Panel for CICS State/Action Tables	231
96. Code Processing Panel	168	148. CICS State/Action tables Panel	232
97. User Defined Data Panel.	169	149. Policy Selection Panel for CICS Link Definitions	232
98. Using STARTUP Policy	169	150. CICS connection links Panel	233
99. Subsystem Startup Processing Panel	170	151. Policy Selection Panel for CICS Link Monitor Periods	234
100. Startup Command Processing Panel	171	152. Link monitoring service period Panel	235
101. Subsystem Shutdown Processing Panel	173	153. Policy Selection Panel for IMS State/Action Table	236
102. Shutdown Command Processing Panel for INIT and FINAL	175	154. IMS state action tables Panel	237
103. Shutdown Command Processing Panel for NORM, IMMED, and FORCE	176	155. Policy Selection Panel for IMS XRF Status File	237
104. Shutdown Reply Processing Panel for NORM, IMMED and FORCE	177	156. IMS Status file specification Panel.	238
105. Member of Panel for Applications	177	157. Policy Selection Panel for IMS XRF Resource	239
106. System Association Panel	178	158. IMS Resource names Panel	239
107. Policy Selection Panel for JES2 Subsystems	182	159. Policy Selection Panel for OPC System Detail	241
108. Environment Definition: JES2 Panel	183	160. OPC System details Panel	241
109. SPOOLFULL Processing Panel for JES2	184	161. Policy Selection Panel for OPC Controller Details	242
110. Policy Selection Panel for JES3 Subsystems	185	162. OPC Controller details Panel	243
111. Policy Selection Panel for CICS Subsystems	186	163. Policy Selection Panel for OPC Special Resources	244
112. CICS control specifications	187	164. OPC special resource definitions Panel	245
113. CICS Resource thresholds selection	189	165. Policy Selection Panel for OPC Workstation Domain	245
114. Policy Selection Panel for DB" Subsystems	190	166. Policy Selection Panel for Timers	246
115. DB2 control entries Panel	191	167. Timer specification Panel.	247
116. Policy Selection Panel for IMS Subsystems	192	168. Policy Selection Panel for <i>Timeout Settings</i>	248
117. IMS control specifications Panel	193	169. Environment Definition: TIMEOUT Panel	249
118. IMS Environ specifications Panel	195	170. Policy Selection Panel for Tape Attendance	251
119. Policy Selection Panel for OPC Subsystems	200	171. Environment Definition: TAPES Panel	251
120. OPC control specifications Panel	200	172. Policy Selection Panel for MVS Component	253
121. Entry Name Selection Panel for Events	201	173. Environment Definition: WTOBUF Overflow Panel	255
122. Define New Event Panel	201	174. Policy Selection Panel for MVS Component	256
123. Policy Selection Panel for Events	202	175. Console Mode After WTOBUF Recovery Panel	257
124. Unset Condition Panel for Events	202	176. Message Processing Panel	258
125. Policy Selection Panel for Service Periods	203	177. Minor Resource Selection Panel	260
126. Service Windows Panel for Service Periods	204		
127. Policy Selection Panel for Triggers	206		
128. Trigger Conditions Panel.	207		
129. STARTUP Condition for Trigger Panel	208		
130. Defining a New SA OS/390 Processor Entry	209		
131. Policy Selection Panel for Processors (OCF-based)	210		
132. Policy Selection Panel for Processors (Screen-Oriented)	210		

178. Policy Selection Panel for MVSCOMP Defaults	261	203. Build Functions Menu Panel	290
179. Policy Selection Panel for System Defaults	262	204. Processor Control File Generation Panel	291
180. Policy Selection Panel for Application Defaults	263	205. ACF and AMC Build Parameters Panel	294
181. Policy Selection Panel for Auto Operators	264	206. SA OS/390 Automation Configuration	299
182. Automation Operator Definition Panel	266	207. Generate NetView and VTAM Information Panel	301
183. Automation Operator NetView Userids Panel	267	208. System Migration Definitions Panel	311
184. Automation Operator: MVSCONS Panel	268	209. System Object Migration Panel.	313
185. Policy Selection Panel for Auto Msg Classes	270	210. Entry Name Migration Panel	314
186. Policy Selection Panel for Network	271	211. UET Keyword-Data specification for entry name HSMMAIL	318
187. Policy Selection Panel for NNT Sessions	274	212. ACF Definitions for CICS	319
188. NNT Sessions Definition Panel.	274	213. Monitoring Service Period	322
189. NNT Sessions Definition Panel.	275	214. Merge TSCF Configuration Panel	324
190. Policy Selection Panel for Resident CLISTS	276	215. Focal Point Forwarding Definitions for Systems	328
191. Policy Selection Panel for Status Details	277	216. GATEWAY Definitions Panel	331
192. Status Display Facility Details Panel	277	217. SAF Environment Definition Panel	333
193. Define New Entry Panel for a Communications Task.	279	218. Example Adjacent NetView Path	334
194. Policy Selection Panel for Communications Task	280	219. PDB Convert Option Panel	342
195. Communications Task Control Parameters Panel	280	220. SA OS/390 Customization Dialog Settings Menu	352
196. Controlled PS/2 List Panel	282	221. Policy Definition Entry Selection Panel	353
197. Processor Operations PS/2 Distribution Panel	283	222. Define Policy Selection Panel	354
198. Unassigned PS/2 List Panel.	284	223. SA OS/390 Data Set Customization Details Panel	356
199. Processor Operations - Task-PS/2 Assignment Panel	285	224. SA OS/390 Report Selection Menu	358
200. Policy Selection Panel for User E-T Pairs	286	225. Policy Data Base Report Parameters Panel	359
201. UET Entry-Type Selection Panel	287	226. List of Generated Resources.	360
202. UET Keyword-Data Specification Panel	288	227. Send Policy Data Base Panel	361

Tables

1. System Automation for OS/390 Library	xvi	9. Correlation of Card Number Fields (for ASCII Card Type)	222
2. Related Products Books	xvii	10. Operator Console Types Supported	224
3. Standard Application Set List	22	11. SA OS/390 Mapping of NetView Automation Operator to SA OS/390 Automated Function	264
4. Quick Retrieval for Entry Types	87	12. Creation of Entry Names	314
5. SA OS/390 Message Classes	116	13. Cross Reference of SA OS/390 1.3 Feature Keywords to SA OS/390 2.1 Dialog Entries	345
6. Relationship Conditions	162		
7. Mapping of Resources Used to I/O Address	214		
8. Correlation of Card Number Fields (for 3270 Card Type)	221		

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 OS/390.

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
BookMaster	CICS
DB2	DFSMS/MVS
ES/9000	ESCON
IBM	IMS
Library Reader	Micro Channel
MQSeries	Multiprise
MVS	MVS/ESA
MVS/XA	Netfinity
Netfinity Manager	OpenEdition
OS/2	OS/390
Parallel Sysplex	RACF
RMF	Sysplex Timer
System/370	System/390
Tivoli	Tivoli Enterprise Console
VM/ESA	VSE/ESA
VTAM	3090

The following terms are trademarks of other companies:

HeathKit

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 System Automation for OS/390 (SA OS/390). 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 OS/390's automation.

Who Should Use This Book

Installing, maintaining and using SA OS/390 is done by personnel from the user groups defined in "SA OS/390 User Groups". This book is primarily intended for automation programmers responsible for defining and maintaining the automation policy.

SA OS/390 User Groups

There are the following primary SA OS/390 user groups:

- **System programmers** are responsible for:

- **product installation**

- This is basically taking the product from tape and moving it into system libraries.

- **system configuration**

- This is basically allocating data sets and configuring individual systems to run the product. At the end of this process, a ready to use installation of the customization dialog and NetView is available to be used by further SA OS/390 user groups. Additional work may be required to help set up processor operations and the NMC workstation.

- **maintenance**

- This is the task of applying APARs to the base code and performing whatever system actions are required afterwards.

- **Automation programmers** are responsible for:

- **policy migration**

- If your enterprise uses policy for earlier SA OS/390 releases, this information needs to be migrated to the new release.

- **policy definitions and distribution**

- Using the customization dialog, automation programmers will, with help from specialist system programmers and operators, define the automation policy for the enterprise. This includes the definition of all automated resources and the individual definitions required for each resource. Thus they are creating the knowledge base that SA OS/390 will use to run their applications and systems. As the creation of this knowledge base can be done centrally on a system where the customization dialog is installed, the policy files that are created need to be distributed to the appropriate target systems before they can be used.

- **automation customization**

- This covers writing and modification of customer automation procedures that run alongside SA OS/390. Existing automation procedures may need modifying to take account of changes that have been made with the current SA OS/390 release.

workstation setup and configuration

This task will also often fall upon the automation programmers. It is concerned with getting a workstation set up to run the SA OS/390 workstation, the NMC workstation and for the workstation user to be able to issue commands from it.

- **Operators** are the end users of the product, as it aims to simplify their jobs. They are responsible for:

ensuring application availability

This basically means starting applications when they need to be started, picking them up if they break down and stopping them when required.

monitoring systems and applications

Operators are responsible for detecting any problems that might occur with the systems or with the application software running upon them. When they find one, they will generally try a few standard fixes and then contact a system programmer.

IPLs and shutdowns

Operators are responsible for ensuring the orderly shutdown of systems and applications when they need to be stopped, and for their successful restarting afterwards.

special operations

This covers all of the things that the operators are asked to do that are not a part of their scheduled (or planned) activities. Examples might be keeping an application available until later in the day, changing the time that a particular event will happen, recycling applications to pick up fixes, or issuing special commands after an application has been started.

Where to Find More Information

The System Automation for OS/390 Library

The following table shows the information units in the System Automation for OS/390 library:

Table 1. System Automation for OS/390 Library

Title	Order Number
<i>System Automation for OS/390 General Information</i>	GC33-7036
<i>System Automation for OS/390 Licensed Program Specifications</i>	SC33-7037
<i>System Automation for OS/390 Planning and Installation</i>	SC33-7038
<i>System Automation for OS/390 Customizing and Programming</i>	SC33-7035
<i>System Automation for OS/390 Defining Automation Policy</i>	SC33-7039
<i>System Automation for OS/390 User's Guide</i>	SC33-7040
<i>System Automation for OS/390 Messages and Codes</i>	SC33-7041
<i>System Automation for OS/390 Operator's Commands</i>	SC33-7042
<i>System Automation for OS/390 Programmer's Reference</i>	SC33-7043
<i>System Automation for OS/390 CICS Automation Programmer's Reference and Operator's Guide</i>	SC33-7044
<i>System Automation for OS/390 IMS Automation Programmer's Reference and Operator's Guide</i>	SC33-7045
<i>System Automation for OS/390 OPC Automation Programmer's Reference and Operator's Guide</i>	SC23-7046

The System Automation for OS/390 books (except Licensed Program Specifications) are also available on CD-ROM as part of the following collection kits:

IBM Online Library OS/390 Collection (SK2T-6700)

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

This softcopy collection includes the IBM Library Reader, a program that enables you to view online documentation.

SA OS/390 Homepage

For the latest news on SA OS/390, visit the SA OS/390 homepage 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 OS/390 base program.

Table 2. Related Products Books

Title	Order Number
<i>ISPF User's Guide</i>	SC34-4484
<i>ISPF Dialog Management Guide and Reference</i>	SC34-4266
<i>MVS/ESA MVS Configuration Program Guide and Reference</i>	GC28-1817
<i>MVS/ESA Planning: Dynamic I/O Configuration</i>	GC28-1674
<i>MVS/ESA Support for the Enterprise Systems Connection</i>	GC28-1140
<i>MVS/ESA Planning: APPC Management</i>	GC28-1110
<i>MVS/ESA Application Development Macro Reference</i>	GC28-1822
<i>MVS/ESA SP V5 System Commands</i>	GC28-1442

Table 2. Related Products Books (continued)

Title	Order Number
<i>MVS/ESA SPL Application Development Macro Reference</i>	GC28-1857
<i>OS/390 Hardware Configuration Definition: User's Guide</i>	SC28-1848
<i>OS/390 Information Roadmap</i>	GC28-1727
<i>OS/390 Information Transformation</i>	GC28-1985
<i>OS/390 Introduction and Release Guide</i>	GC28-1725
<i>OS/390 JES Commands Summary</i>	GX22-0041
<i>OS/390 Licensed Program Specifications</i>	GC28-1728
<i>OS/390 Printing Softcopy Books</i>	S544-5354
<i>OS/390 Starting Up a Sysplex</i>	GC28-1779
<i>OS/390 Up and Running!</i>	GC28-1726
<i>Planning for the 9032 Model 3 and 9033 Enterprise Systems Connection Director</i>	SA26-6100
<i>Resource Access Control Facility (RACF) Command Language Reference</i>	SC28-0733
<i>S/390 MVS Sysplex Overview – An Introduction to Data Sharing and Parallelism</i>	GC23-1208
<i>S/390 MVS Sysplex Systems Management</i>	GC23-1209
<i>S/390 Sysplex Hardware and Software Migration</i>	GC23-1210
<i>S/390 MVS Sysplex Application Migration</i>	GC23-1211
<i>S/390 Managing Your Processors</i>	GC38-0452
<i>Tivoli/Enterprise Console User's Guide Volume I</i>	GC31-8334
<i>Tivoli/Enterprise Console User's Guide Volume II</i>	GC31-8335
<i>Tivoli/Enterprise Console Event Integration Facility Guide</i>	GC31-8337
<i>Tivoli for OS/390 NetView V1R3 Administration Reference</i>	SC31-8222
<i>Tivoli for OS/390 NetView V1R3 Application Programmer's Guide</i>	SC31-8223
<i>Tivoli for OS/390 NetView V1R3 APPN Topology and Accounting Agent Guide</i>	SC31-8224
<i>Tivoli for OS/390 NetView V1R3 Automation Guide</i>	SC31-8225
<i>Tivoli for OS/390 NetView V1R3 AON Customization Guide</i>	SC31-8662
<i>Tivoli for OS/390 NetView V1R3 AON User's Guide</i>	GC31-8661
<i>Tivoli for OS/390 NetView V1R3 Bridge Implementation</i>	SC31-8238
<i>Tivoli for OS/390 NetView V1R3 Command Reference Vol. 1</i>	SC31-8227
<i>Tivoli for OS/390 NetView V1R3 Command Reference Vol. 2</i>	SC31-8227
<i>Tivoli for OS/390 NetView V1R3 Customization Guide</i>	SC31-8228
<i>Tivoli for OS/390 NetView V1R3 Customization: Using Assembler</i>	SC31-8229
<i>Tivoli for OS/390 NetView V1R3 Customization: Using Pipes</i>	SC31-8248
<i>Tivoli for OS/390 NetView V1R3 Customization: Using PL/I and C</i>	SC31-8230
<i>Tivoli for OS/390 NetView V1R3 Customization: Using REXX and the NetView Command List Language</i>	SC31-8231
<i>Tivoli for OS/390 NetView V1R3 Data Model Reference</i>	SC31-8232
<i>Tivoli for OS/390 NetView V1R3 Installation and Administration Guide</i>	SC31-8236
<i>Tivoli for OS/390 NetView V1R3 Messages and Codes</i>	SC31-8237

Table 2. Related Products Books (continued)

Title	Order Number
<i>Tivoli for OS/390 NetView V1R3 MultiSystem Manager User's Guide</i>	SC31-8607
<i>Tivoli for OS/390 NetView V1R3 NetView Graphic Monitor Facility User's Guide</i>	GC31-8234
<i>Tivoli for OS/390 NetView V1R3 NetView Management Console User's Guide</i>	GC31-8665
<i>Tivoli for OS/390 NetView V1R3 User's Guide</i>	SC31-8241
<i>Tivoli for OS/390 NetView V1R3 Planning Guide</i>	GC31-8226
<i>Tivoli for OS/390 NetView V1R3 RODM and GMFHS Programmer's Guide</i>	SC31-8233
<i>Tivoli for OS/390 NetView V1R3 Security Reference</i>	SC31-8606
<i>Tivoli for OS/390 NetView V1R3 SNA Topology Manager and APPN Accounting Manager Implementation Guide</i>	SC31-8239
<i>Tivoli Management Platform Reference Guide</i>	GC31-8324
<i>TSO/E REXX/MVS User's Guide</i>	SC28-1882
<i>TSO/E REXX/MVS Reference</i>	SC28-1883
<i>VM/XA SP GCS Command and Macro Reference</i>	SC23-0433
<i>VSE/SP Unattended Node Support</i>	SC33-6412
<i>VTAM Messages and Codes</i>	SC31-6493
<i>VTAM Network Implementation Guide</i>	SC31-6404
<i>VTAM Network Implementation Guide</i>	SC31-6434

Using LookAt to look up message explanations

LookAt is an online facility that allows you to look up explanations for z/OS messages, system abends, and some 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/servers/eserver/zseries/zos/bkserv/lookat/lookat.html>

or from anywhere in z/OS where you can access a TSO command line (for example, TSO prompt, ISPF, z/OS UNIX System Services running OMVS).

To find a message explanation on the Internet, go to the LookAt Web site and simply enter the message identifier (for example, IAT1836 or IAT*). You can select a specific release to narrow your search. You can also download code from the *z/OS Collection*, SK3T-4269 and the LookAt Web site so you can access LookAt from a PalmPilot (Palm VIIx suggested).

To use LookAt as a TSO command, you must have LookAt installed on your host system. You can obtain the LookAt code for TSO from a disk on your *z/OS Collection*, SK3T-4269 or from the LookAt Web site. To obtain the code from the LookAt Web site, do the following:

1. Go to <http://www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/lookat.html>.
2. Click the **News** button.
3. Scroll to **Download LookAt Code for TSO and VM**.

4. Click the ftp link, which will take you to a list of operating systems. Select the appropriate operating system. Then select the appropriate release.
5. Find the **lookat.me** file and follow its detailed instructions.

To find a message explanation from a TSO command line, simply enter: **lookat message-id**. LookAt will display the message explanation for the message requested.

Note: Some messages have information in more than one book. For example, IEC192I has routing and descriptor codes listed in *z/OS MVS Routing and Descriptor Codes*. For such messages, LookAt prompts you to choose which book to open.

Accessing licensed books on the Web

OS/390 licensed documentation in PDF format is available on the Internet at the IBM Resource Link Web site at:

<http://www.ibm.com/servers/resourceLink>

Licensed books are available only to customers with a OS/390 license. Access to these books requires an IBM Resource Link Web userid and password, and a key code. With your OS/390 order you received a memo that includes this key code.

To obtain your IBM Resource Link Web userid and password log on to:

<http://www.ibm.com/servers/resourceLink>

To register for access to the z/OS licensed books:

1. Log on to Resource Link using your Resource Link userid and password.
2. Click on **User Profiles** located on the left-hand navigation bar.
3. Click on **Access Profile**.
4. Click on **Request Access to Licensed books**.
5. Supply your key code where requested and click on the **Submit** button.

If you supplied the correct key code you will receive confirmation that your request is being processed. After your request is processed you will receive an e-mail confirmation.

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

To access the licensed books:

1. Log on to Resource Link using your Resource Link userid and password.
2. Click on **Library**.
3. Click on **zSeries**.
4. Click on **Software**.
5. Click on **OS/390**.
6. Access the licensed book by selecting the appropriate element.

Summary of Changes

This section summarizes the changes for SA OS/390 2.1 as follows:

- “What’s New in This Book for SA OS/390 2.1”
- “What’s New in SA OS/390 2.1”

What’s New in This Book for SA OS/390 2.1

This book contains information previously presented in *System Automation for OS/390 Customization*, GC28-1566-01, which supports the customization of SA OS/390, Release 3.

This edition has been restructured in a way that the information not directly pertaining to the use of the customization dialog has been moved to other places in the SA OS/390 library, mainly into *System Automation for OS/390 Customizing and Programming*. This fact is reflected in the new title of this documentation.

How to define automation policy for all SA OS/390 entry types is now consolidated in one chapter: “Chapter 5. Defining Automation Policy” on page 87.

The following documentation has been added:

- “Chapter 3. Automation Concepts” on page 31 explains the automation concepts which you need to understand before using the customization dialog. Especially, the new concepts of generated resources, grouping support and new relationships are presented. Not all of the information contained here is new, but existing conceptual information has been consolidated here.
- “Chapter 4. Scenarios on How to Use the Customization Dialog” on page 51 contains scenarios that help to understand how to apply SA OS/390 automation concepts using in concrete situations.
- “Chapter 5. Defining Automation Policy” on page 87 contains the documentation of the new *Product Automation* policy object which has been introduced to integrate automation for the following products:
 - DB2
 - CICS
 - IMS
 - OPC

New policy items for entry type *Application* have been introduced for applications of these types accordingly.

- “Appendix C. DB2 Automation for System Automation for OS/390” on page 365 contains automation information for DB2 applications.

What’s New in SA OS/390 2.1

A summary of the new SA OS/390 2.1 functionality is presented in the following list:

- The main design change of SA OS/390 2.1 compared to earlier releases is the implementation of *goal driven automation*. The new automation concepts

comprise for example, enhanced grouping support and enhanced relationship support. These are described in the new chapter “Chapter 3. Automation Concepts” on page 31.

The new concepts involve major changes in defining automation policy for entry types *Application* and *ApplicationGroup*, like new policy items and new entry fields in the affected customization dialog panels. For a summary of such changes, select option 5. *Changes for this Release* from the *HELP* menu of the action bar.

Also a changed behaviour of automation results from the goal driven automation in contrast to the earlier command driven automation. As this information is important when planning to use SA OS/390, you can find it in *System Automation for OS/390 Planning and Installation*.

- The entry point to the customization dialog consists of a new primary panel called *Customization Dialog Primary Menu*, now fully enabling ISPF nested options and stacked commands. See “How to Start the Customization Dialog” on page 14.
- For the automation of the products *DB2*, *CICS*, *IMS* and *OPC*, a new *Product Automation* entry type has been introduced into the *Entry Type Selection* panel. “Product Automation Policy Object” on page 229 contains the related documentation.

Also, new policy items for entry type *Application* have been introduced for applications of these types. These are described in “Application Policy Object” on page 141.

- The command PERSONALIZE has changed to the *SETTINGS* option which is selectable from the *Customization Dialog Primary Menu*.
- New report facilities are implemented and described in “Creating Reports about Policy Databases” on page 358.
- Multi-User support is extended to application groups.
- A new *User-written Functions Selection Menu* panel is added which allows you to add installation specific functions and includes one example to send an OS/390 data set to a specific destination. This panel can be reached from the *Customization Dialog Primary Menu* using option *U (User-defined selections)*.
- The entry type *Includes* is no longer supported. It is only available for downward compatibility and will be removed in the next release. If *Includes* policy objects are found by SA OS/390 2.1 load, an error message is issued.
- An enhanced COPY function is implemented. See “Policy Items Found on Many Selection Panels” on page 27.
- An enhanced service period definition is provided and described in “Service Periods Policy Object” on page 203.
- A variety of new relationships/dependencies can be defined on SA OS/390 2.1. Information about these can be found in “Relationship Support” on page 36 (conceptual information) and “More About Policy Item RELATIONSHIPS” on page 157 (how to define relationships).
- Command texts, wherever they can be entered, may now spread over two input lines.
- Monitoring granularity is individualized by making the monitoring routine and monitoring interval definable multiple times (on application resource basis, not only once on system basis).
- Users can now specify flexible start-up commands for applications. Therefore, a new policy item *STARTUP* has been defined for entry type *Application*. This is described in “More About Policy Item STARTUP” on page 169.

- Policy item SHUTDOWN of *Applications* is enhanced with an additional shutdown phase: FINAL. For more information refer to “More About Policy Item SHUTDOWN” on page 173.
- Automation programmers can specify message prefixes for an application. Refer to “More About Policy Item AUTOMATION INFO” on page 150 for more information.
- The policy item RESTART for *Applications* is no longer available. The specification of restart commands is now possible under policy item MESSAGES/USER DATA with message ID ACORESTART. For more information refer to “More About Policy Item MESSAGES/USER DATA” on page 163.
- The policy item MESSAGES/USER DATA for *Applications* now allows the specification of free format user data when defining messages. A new message type USER is introduced. For this type, any user data is allowed as is allowed under the specification of *User E-T Pairs*. This means it is no longer necessary to specify additional application data under *User E-T Pairs*, but it can be specified directly in the application where the data belongs to. For more information refer to “More About Policy Item MESSAGES/USER DATA” on page 163.
- The policy item MINOR RESOURCES is renamed to MINOR RESOURCE FLAGS and is moved from entry type *System* to entry type *MVS Component*. Minor resources need to be defined only once and then you can link them to various systems. For more information refer to “MVS Component Policy Object” on page 252.
- For entry type *Auto Operators*, the policy item OPERATORS is enhanced in a way that you can specify SYSCLONE variables and the &DOMAIN. variable. For entry type *Network*, the policy item FORWARDS is enhanced in a way that you can specify the &DOMAIN. variable. For more information refer to “More About the OPERATORS Policy Item” on page 264 and “Network Policy Object” on page 270.
- The possibility to specify a command has been moved from the policy item THRESHOLDS to policy item MESSAGES/USER DATA for entry type *MVS Component*.
- Enhanced migration and conversion support is provided with SA OS/390 2.1. For example, you can use SA OS/390 2.1 to automatically convert a SA OS/390 1.3 or earlier policy database to the new format. See “Chapter 7. Migration Functions” on page 309 and “Chapter 10. Conversion Function” on page 341 for more information.
- As the processing of automation in a sysplex has been divided up between an automation manager and multiple automation agents (refer to *System Automation for OS/390 User’s Guide*), the build process has been redesigned. For more information refer to “Chapter 6. Building and Distributing Control Files” on page 289.

The following items have been removed from SA OS/390:

- I/O exception monitoring
- service level monitoring

Notes on Terminology

MVS:

References in this book to “MVS” refer either to the MVS/ESA™ product or to the MVS™ element of OS/390.

PS/2:

The *Processor Operations PC* often is referred to as *Processor Operations PS/2* in both SA OS/390 panels and SA OS/390 documentation.

NetView

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

Resources, Applications and Subsystems

These three terms are used in the SA OS/390 library to designate automated entities. The correct term to be used depends from the context in which it is used. Refer to the *Introduction* chapter in *System Automation for OS/390 Operator's Commands* for the definition of these terms.

Chapter 1. Overview

Before you can use SA OS/390 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 OS/390 customization dialog to provide information to SA OS/390 such as:

- Which resources (like 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 which automation, like 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 automation handles certain tasks and events during system operation.

The customization dialog leads the SA OS/390 policy administrator through the process of defining resources and automation for the enterprise. From this information SA OS/390 automatically extracts the required bits 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 OS/390 to work within your enterprise. Making this determination is part of the process of planning to use SA OS/390. There are many considerations for planning to use SA OS/390 of which you should be aware. For this purpose, also refer to the planning instructions given in *System Automation for OS/390 Planning and Installation*.

Note: Do not use the customization dialog on a Kanji terminal as they will not work. If you have your terminal defined as a 3278KN 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" gives an outline of the process of creating enterprise automation policy.
- "Customization Dialog Introduction" on page 2 introduces all entry types.
- "Considerations for the SA OS/390 Graphical Interface" on page 9 provides some information on how to customize the monitoring aspect of SA OS/390.
- "Considerations for Automation" on page 10 provides some information on how to customize the automation aspect of SA OS/390.
- "Building and Distributing SA OS/390 Automation Policy" on page 11 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:

Overview

1. At the configuration focal point, where you have the customization dialog installed, you create the information into the policy database using the ISPF input panels of the customization dialog. The description of the customization dialog comprises the major part of this documentation.
2. Then use the BUILD function of the customization dialog to produce the required output, the system operations control files (ACF and AMC file).
3. Then you distribute the generated output to the sysplexes or stand-alone systems that you want to automate.

Refer to “Chapter 6. Building and Distributing Control Files” on page 289 and Figure 206 on page 299 for more information on building and distributing the generated output files.

Customization Dialog Introduction

All resources of your enterprise, such as systems or applications, or other objects that you use to automate your enterprise data processing, like components of your OS/390® operating system, are called *entry types*. A complete list of SA OS/390 entry types is listed on the *Entry Type Selection* panel (see Figure 8 on page 25), which is the first panel you will see after you decided with which enterprise, or, respectively, with which policy database you are going to work.

For each entry type you can invoke a series of panels to create, modify or delete instances of a certain entry type. These instances are called *policy objects* and are referred to by their *entry name*. 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 presented in “Chapter 4. Scenarios on How to Use the Customization Dialog” on page 51.

(1/ENT) Enterprise

An Enterprise is the top level of your SA OS/390 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

(2/GRP) Group

A group is the largest subdivision of an SA OS/390 enterprise. An SA OS/390 enterprise may have several groups, each consisting of systems and/or subgroups. Groups can be used to represent a sysplex. 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 OS/390. The system may be MVS, VM, VSE, TPF, 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 an OS/390 subsystem or job that runs on a system in the SA OS/390 enterprise. SA OS/390 monitors applications through the messages they issue to the system, MVS commands, and SA OS/390-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 OS/390 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 hardware on which a system runs. It can be a single-image or physically partitioned processor. The processors supported by SA OS/390 fall into two categories.

Screen-oriented processor: A processor that interacts with human operators through line-mode display consoles. For System/390[®] and System/370[™] processors, the display consoles use either an IBM[®] 3270 data stream or an ASCII data stream. The screen-oriented processors supported by SA OS/390 as targets are 9021, 9121, 3090[™], 308x and 4381 processors.

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

Overview

(11/CMP) Communications Path

A communications path is a set of communications facilities that SA OS/390 processor operations functions use to connect a focal point system to a target system. There must be at least one active communications path connected to the target system by means of communications ports.

There are two types of communications paths: a NetView connection and a personal computer (PC). The type of communications path which is used depends upon the type of processor, the operating system running on the target system, and whether the path is to a system console (SC) or operator console (OC), as shown below.

Note:

The *Processor Operations PC* often is referred to as *Processor Operations PS/2* in both SA OS/390 panels and SA OS/390 documentation.

Path	Screen-oriented processors	OCF-based processors
PC	o SC for all processors	o OC for MVS/XA, VM, VSE, TPF
	o OC for MVS,VM,VSE,TPF	o Optional OC for MVS/ESA
NVC	Not supported	o SC
		o Via console integration - Optional OC for MVS/ESA - OC console for LINUX - OC for coupling facility

(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 OS/390 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 OS/390 data sets, resources and facilities.

(34/MDF) MVSCOMP Defaults

Sets defaults for OS/390 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/AMC) Auto Msg Classes

Defines tasks to which message classes are assigned.

(39/NTW) Network

Defines Gateway, adjacent NetView domains, and automation focal points.

(40/NNT) NNT Sessions

Defines NetView to NetView sessions (user IDs, passwords, NetView domains).

(41/RES) Resident CLISTs

Automation CLISTs which are to be preloaded in order to improve performance.

(42/SCR) Status Details

Defines NetView Status Display facility (SDF) settings.

(43/CMT) Communications Task

A communications task is an SA OS/390 user task responsible for all communications with a number of personal computers. Communications tasks receive commands from target control tasks and send inbound messages to the message monitoring task. Each communications task can control from 1 to 64 personal computers.

(98/ICL) Includes

User edited automation control file fragments to be included in the

Overview

automation control file. No more support will be provided for this entry type. It is available only for clean up purposes and will not be respected for automation.

(99/UET) User E-T Pairs

User-defined entry/type data for the automation control file. Used for user-defined extensions to automation policy.

The normal procedure to define automation policy for a certain policy object is to select this object via its name. Then you get a list of so-called *policy items* specific for the entry type you are working with.

Policy Objects and Policy Items

Using entry type *Group* as an example, Figure 1 on page 7 outlines the panel flow in the customization dialog for defining automation policy for the entry types. The labels (for example, AOFGETYP, AOFGENAM) are the panel IDs that are displayed after you enter the command `PANELID` into the command line.

In this example, the arrow denoted with *1a*) denotes the panel flow required if you need to create a new *Group* entry (policy object), while *1b*) denotes the flow if you want to define the automation policy for an existing policy object. Hence, a policy object is an instance of a certain entry type.

In case of *1b*), after selecting an existing policy object and pressing the ENTER key, you will see the policy items available for this policy object. Policy item examples for policy objects from entry type *Group* are *DESCRIPTION*, *GROUP INFO*. If you select policy items, you will get specific panels where you can define values for the object's automation policy.

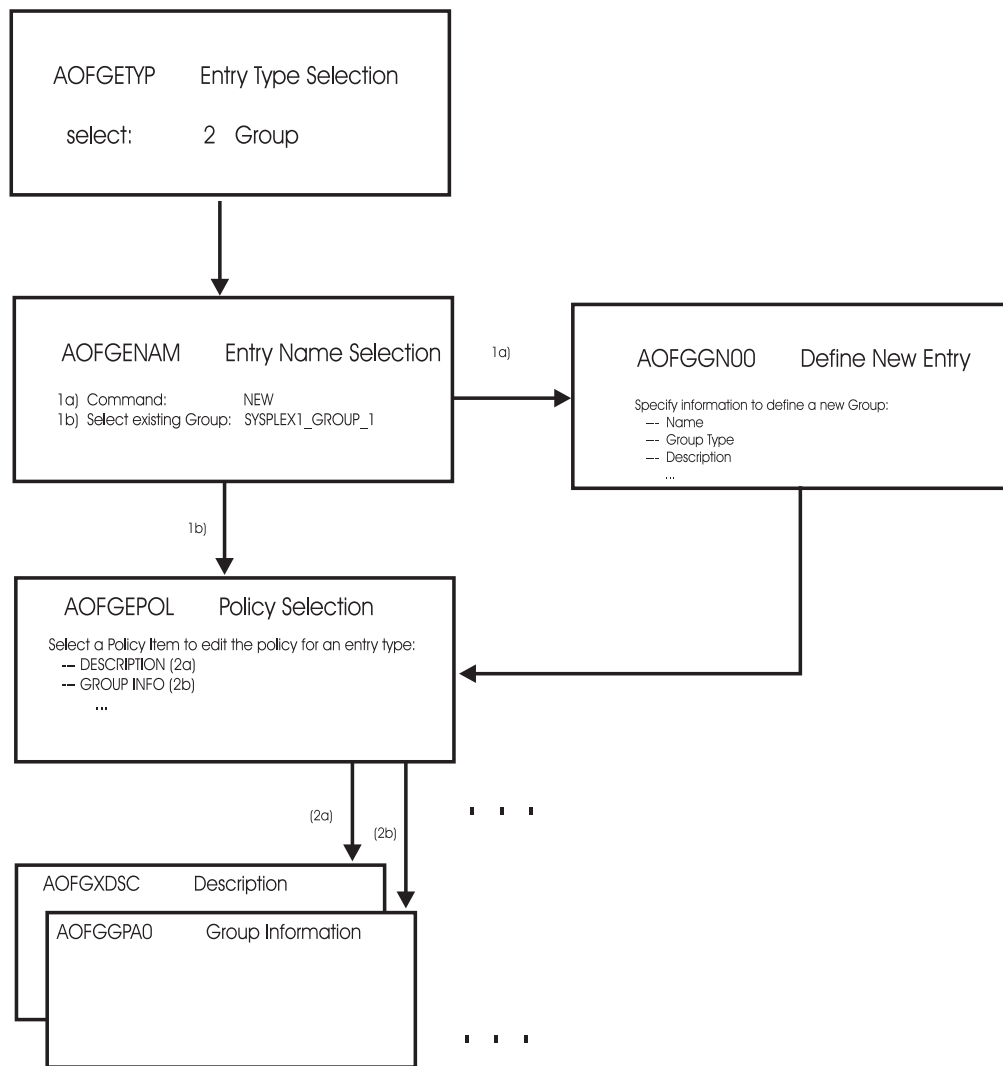


Figure 1. Panel Flow of the Customization Dialog

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 OS/390 enterprise.

For example, if you choose option 2 on the *Entry Type Selection* panel (Figure 2 on page 8) to specify *Group* policy, Figure 3 on page 8 is displayed.

Overview

```

MENU  COMMANDS  HELP
-----
AOFGETYP                      Entry Type Selection
Option ==>>

1 ENT      Enterprise          30 TMR      Timers
2 GRP      Group              31 TMO      Timeout Settings
3 SBG      SubGroup           32 TPA      Tape Attendance
4 SYS      System             33 MVC      MVS Component
5 APG      ApplicationGroup   34 MDF      MVSCOMP Defaults
6 APL      Application        35 SDF      System Defaults
7 EVT      Events             36 ADF      Application Defaults
8 SVP      Service periods    37 AOP      Auto Operators
9 TRG      Triggers           38 AMC      Auto Msg Classes
10 PRO     Processor          39 NTW      Network
11 CMP     Communications Path 40 NNT      NNT Sessions
                                     41 RES      Resident CLISTS
                                     42 SCR      Status Details
                                     43 CMT      Communications Task

20 PRD     Product Automation

                                     98 ICL      Includes
                                     99 UET      User E-T Pairs

(*) Multi-User-Capable
F1=HELP    F2=SPLIT  F3=END     F4=RETURN  F5=RFIND   F6=RCHANGE
F7=UP      F8=DOWN   F9=SWAP   F10=LEFT   F11=RIGHT  F12=RETRIEVE
More:      +

```

Figure 2. Entry Type Selection Panel

Figure 3 shows the *Group* policy objects *SYSPLEX1* and *SYSPLEX2*, and Figure 4 on page 9 shows the policy items for the selected policy object *SYSPLEX1*.

```

COMMANDS  ACTIONS  VIEW  HELP
-----
AOFGENAM                      Entry Name Selection
Command ==>>>                                SCROLL==>> PAGE

Entry Type : Group                      PolicyDB Name : DATABASE_NAME
                                         Enterprise Name : YOUR_ENTERPRISE

Action  Entry Name          C Short Description
S       SYSPLEX1           S Sysplex with SYSTEM1, SYSTEM2, SYSTEM3
        SYS1SYS2           S Sysplex with SYS1 and SYS2

```

Figure 3. Entry Name Selection Panel for Groups

If you select an existing group, the *Policy Selection* panel is displayed. This panel gives you a list of the policy items for a group. If you enter S for SELECT beside any policy item, the appropriate policy input panel is displayed. Figure 4 on page 9 shows the *Policy Selection* panel for groups, listing the policy items for the selected policy object *SYSPLEX1*.

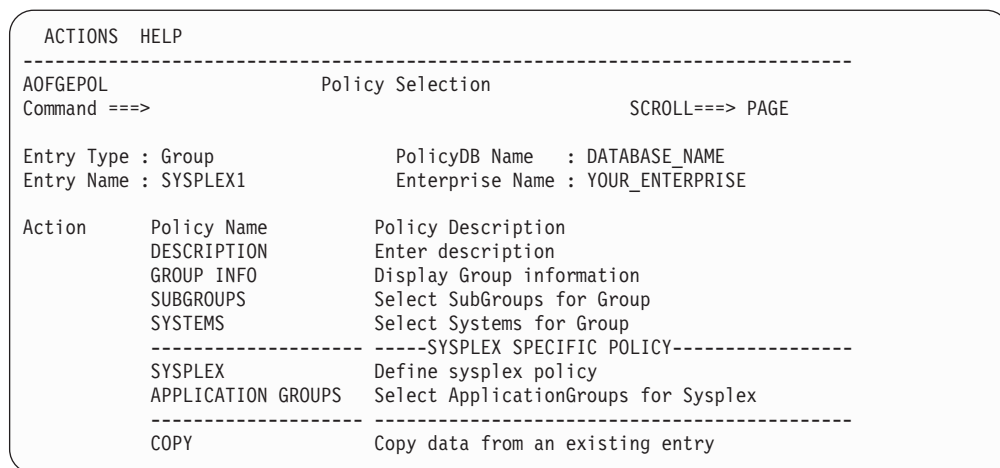


Figure 4. Policy Selection Panel for Sysplex Groups

Considerations for the SA OS/390 Graphical Interface

The customization dialog leads the SA OS/390 policy administrator through several panels to define the systems and other automated resources of the enterprise. The resulting *enterprise policy* provides the information which SA OS/390 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 OS/390. The automation manager component of SA OS/390 monitors the resources and their status. SA OS/390 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 OS/390 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 OS/390 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 OS/390 enterprise.

The grouping of resources allowed by SA OS/390 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. Applications can be part of more than one application group, and systems can be part of more than one subgroup or group.

Overview

For information on how to set up the graphical interface on the NMC workstation refer to *System Automation for OS/390 Planning and Installation*. For information on how to use this graphical interface refer to *System Automation for OS/390 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 OS/390 enterprise uses SA OS/390 routines to automate operator functions for its own applications and OS/390 components. The programs that automate these functions are called *SA OS/390 automation procedures*. *SA OS/390 automation procedures* invoke SA OS/390 provided *common routines* and *generic routines*. You can customize your automation by writing your own *automation procedures* which can also invoke the SA OS/390 provided routines.

SA OS/390 provides automation procedures that are internally invoked according to your entries in the NetView message automation table and that manage the following:

- 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-OS/390 resources
- Any other applications
- Special automation functions are also provided to manage CICS[®], IMS[™], DB2[™], and to integrate with OPC

SA OS/390 provides automation procedures that enable recovery of the following OS/390 components and data sets:

- SYSLOG data sets
- LOGREC data sets
- System Management Facility (SMF) data sets
- Write-to-operator (WTO) buffers
- JES spools
- OS/390 dump data sets
- Action Message Retention Facility (AMRF) buffers
- sysplex resources, for example, coupling facility

With SA OS/390 you can also define policy for the automated startup and shutdown of processors and for the automated IPL and shutdown of operating systems.

The above lists represent applications and OS/390 components for which SA OS/390 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 OS/390 policy administrator uses the customization dialog to set the way in which enterprise resources are to be handled by SA OS/390 when predefined

conditions are met. In addition to invoking provided automation procedures, SA OS/390 can be used to trigger user-written automation procedures.

Each entry type can be used to specify multiple sets of objects of the same entry type. 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.

Building and Distributing SA OS/390 Automation Policy

For automation of your enterprise you may create:

- one policy database for each sysplex **and** one policy database for each stand-alone system or
- you may combine multiple sysplexes into one policy database or
- you may even create one policy database for the complete enterprise, which is recommended.

Once you have entered all data necessary to define your enterprise monitoring and automation policy into the policy database, you use the BUILD function of the customization dialog to build the files needed by SA OS/390. These include:

- The automation control file (ACF) for each set of definitions, and a master include file for each system
- automation manager configuration file (AMC), that contains an image of the automated systems in a sysplex or of a stand-alone system
- A processor control file
- NetView and VTAM[®] information
- Processor operations PS/2 profile information

The automation control definitions are needed if system operations automation functions are desired. If processor operations automation functions (monitoring processors, communications tasks, and communications paths) are desired, the processor operations control file, NetView and VTAM definitions, and processor operations PS/2 profile information must also be built.

Refer to “Chapter 6. Building and Distributing Control Files” on page 289 for more information on the SA OS/390 build function.

Each of these files (with the exception of the processor operations control file, which is required only on the SA OS/390 focal point system) must be distributed to other systems in the SA OS/390 enterprise:

- The automation control files must be distributed to each system in the SA OS/390 enterprise where SA OS/390 system operations functions are to run. In general, this is all of the OS/390 systems in the SA OS/390 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 it is recommended, if possible, to use a shared data set for the whole enterprise.

- If you want to use the processor operations stand-alone function, then the PS/2 profile information should be distributed to each processor operations PC that should provide that function.

Overview

Chapter 2. Using the Customization Dialog

The SA OS/390 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 OS/390 Planning and Installation*.

The customization dialog provides panels on which you create and work with SA OS/390 automation policy definitions. The task of defining and building automation policy is performed on the SA OS/390 configuration focal point system from where it is distributed to the systems and sysplexes that you want to automate (see also Figure 206 on page 299).

The combined definition of all the resources is called an *enterprise definition* or also an *automation configuration*. SA OS/390 enterprise definition data entered in the customization dialog is stored in an OS/390 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 OS/390 provides sample policy databases containing sample enterprise definitions. These are described in "Using Sample SA OS/390 Policy Databases" on page 20. Choose the sample policy database that is closest to the enterprise that you wish to define. The samples provided with SA OS/390 include:

- A single system enterprise
- A multi-system enterprise with four systems
- A multi-system enterprise with four systems configured in an OS/390 sysplex
- An enterprise with a four-system sysplex with Automatic Restart Manager (ARM)

The sample policy database is just a starting point: you will need to modify the resource definitions and automation policy specifications to match your enterprise configuration and automation requirements. Some policy default values inherited from the samples will not exactly meet your needs. So if you plan to use a sample database it is advisable to create a personal defaults database using a sample database or the default database as a model, review the default values inherited from the sample automation policy and change them as necessary. You can then use your personal defaults database as a model for your production databases. Once you complete the definition of your new policy database, it can be used as a model for the definition of other new policy databases.

You can create many different SA OS/390 enterprise definitions. However, SA OS/390 uses only one enterprise definition at a time. Each SA OS/390 enterprise definition is stored in a different policy database. The customization dialog enables you to edit, copy, delete, and build SA OS/390 enterprise definitions after you create them.

Using the Customization Dialog

You select entry types from the list presented on the *Entry Type Selection* panel (Figure 8 on page 25) and certain objects from these entry types to specify the way enterprise resources are to be handled by SA OS/390 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 OS/390 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 OS/390 for automation only you must still use the customization dialog to create policy for the resources you want to automate.

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 OS/390 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 OS/390 displays its *Customization Dialog Primary Menu* as shown in Figure 5.

Note: The installation can use the keyword INITSEL for the start-up exec INGDLG to automatically provide an initial selection on the *Customization Dialog Primary Menu* (see Figure 5). 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.

```
-----
MENU  OPTIONS  HELP
-----
                                SA OS/390 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 Migrate           Migrate files into a Policy Data Base
6 Merge             Merge a TSCF configuration 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
F1=HELP   F2=SPLIT   F3=END     F4=RETURN   F5=RFIND   F6=RCHANGE
F7=UP     F8=DOWN    F9=SWAP    F10=LEFT    F11=RIGHT  F12=RETRIEVE
```

Figure 5. Customization Dialog Primary Menu

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

Using the Customization Dialog

- 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 8 on page 25.
- Type a '?' into 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 6 on page 18. 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 17) or add an existing policy database to your display list.

If other users want to view the policy database you have just created, they must add it to their PolicyDB list using the ADD command.

Additionally, you can select the following options from the *Customization Dialog Primary Menu*:

0 Settings

With this option, you reach the *Customization Dialog Settings Menu*, see "Changing Your Customization Dialog Environment" on page 351.

1 Open

With this option, you open the policy database of your choice. See the description above.

2 Build

With this option, you reach the *Build Functions Menu*, see "Chapter 6. Building and Distributing Control Files" on page 289.

3 Report

With this option, you reach the *Report Selection Menu*, see "Creating Reports about Policy Databases" on page 358.

4 Policies

With this option, you reach the *Policy Data Base Selection* panel with the list of existing policy databases, see "Creating a New Policy Database" on page 17.

5 Migrate

With this option, you reach the *System Migration Definitions* panel, see "Migrate Automation Control File" on page 309.

6 Merge

With this option, you reach the *Merge TSCF Configuration* panel, see "Merge TSCF Configuration" on page 324.

U User

This option invokes the *User-written Functions Selection Menus*. This menu allows you to add installation specific functions to the customization dialog. For more information refer to "User-written Functions" on page 360.

X Exit Terminate the customization dialog

Using the Customization Dialog

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
- If you want to start with defining a new application group for example, just type:
APG; 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*:
=APG; S CICS_APG

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

How to Use 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 Actions column and press the ENTER key, this action, the default action, is used. For example, entering “S” under Actions 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 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 on 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 command verifies input, saves the data, and exits the current panel.
- Pressing the ENTER key verifies input and re-displays the current panel.

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** and **APG**. For other entry types, only one user at a time can access a policy database. See also “Concurrent User Access” and “Appendix A. Concurrent Customization by Multiple Users” on page 349 for further information.

How to Cancel Input on a Panel

To cancel your input on the current panel, type the CANCEL command 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.

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 OS/390 customization dialog). Initially the *Policy Data Base Selection* panel shows no policy databases. Otherwise it shows all available policy databases. Figure 6 on page 18 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 request the “Name and data set name” option (option 2) from the VIEW pull-down.

Using the Customization Dialog

```

MENU  COMMANDS  ACTIONS  VIEW  HELP
-----
A0FGPDB          Policy Data Base Selection
Command ==>>>                                SCROLL==>> PAGE

Action   PolicyDB Name      Enterprise Name/Data Set Name
        DATABASE_NAME  YOUR_ENTERPRISE
        RESOURCE_PDB  'HART.DOCU.PDB'
        SH_POLICYDB   MY_ENTERPRISE
        SH_POLICYDB   'HART.RESOURCE.PDB'
        SH_POLICYDB   SH_ENTERPRISE
        SH_POLICYDB   'HART.SH.POLICYDB'

F1=HELP   F2=SPLIT   F3=END   F4=RETURN   F5=RFIND   F6=RCHANGE
F7=UP     F8=DOWN    F9=SWAP  F10=LEFT    F11=RIGHT  F12=RETRIEVE

```

Figure 6. Policy Data Base Selection Panel

To create a new database, type N (New) on the command line. You will see a display similar to Figure 7.

```

COMMANDS  HELP
-----
A0FGPDBN          Create a New Policy Database
Command ==>>>

To define a new policy database, specify the following information:
PolicyDB Name. . . . . _____
Enterprise Name. . . . .
Data Set Name. . . . .

Managed storage. . . . . NO          YES          NO
Management class . . . . .          Blank for default management class *
Storage class. . . . .              Blank for default storage class *
Volume serial. . . . .              Blank for authorized default volume
Data class . . . . .                Blank for default data class *
Space units. . . . . CYLINDERS      CYLS TRKS BLKS KB MB
Primary quantity . . . . . 1        1 to 999 - In above units
Secondary quantity . . . . . 1      0 to 999 - In above units
Directory blocks . . . . . 50      1 to 999 or blank - Required for PDS
Record format. . . . . : FB
Record length. . . . . : 80
Block size . . . . . 3120
Data Set Name type . . . . . PDS     LIBRARY PDS *
Device Type. . . . . SYSDA

* Used only if Managed storage = YES

Specify one of the existing policy databases to serve as the model
for the policy database that is being created:
Model PolicyDB name. . . *DEFAULTS      PolicyDB name or "?"
                                           for list of names

```

Figure 7. 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 OS/390 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 OS/390 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

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

specifies the management class that should be used to obtain the data management related information for the allocation of the data set.

Storage class

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

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

specifies the primary space allocation in space units. The default is 1.

Secondary quantity

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

is FB and cannot be changed.

Using the Customization Dialog

Record length

is 80 and cannot be changed.

Blocksize

defines the block size for the data set. It must be a multiple of 80.

Data Set Name type

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.

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 OS/390 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 7 on page 18 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 OS/390 displays a *Command Progress Display* panel. When database creation is finished, the *Entry Type Selection* panel is displayed for the new database. See Figure 8 on page 25 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 OS/390 Policy Databases

SA OS/390 provides sample enterprise organizations implemented as SA OS/390 policy databases. Sample databases are installed when the SA OS/390 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:

*DEFAULTS

This sample is bound for experienced users who are going to build a new policy database from scratch and contains only a few default entries which are required in most policy databases. Any policy databases you create should contain these values or locally modified equivalents. They cover

things such as automation operators and resident CLISTs. This is the database you use to model your own databases if you want to start with no defined systems, applications or other resources.

***MULTISYS - Multiple OS/390 Images, Non Sysplex**

This sample consists of four OS/390 systems in the standard SA OS/390 test configuration, that is:

- Focal Point
- Backup Focal Point
- Target 1
- Target 2

All images are connected through SA OS/390 gateways with active status forwarding and associated command facilities. The standard application set is defined and linked to each system.

A sample screen-oriented processor and 2 sample OCF-based processors are defined, along with communications paths to each of the processors and a sample communications task.

Also included is a sample class definition of processor operations including IXC message automation.

System symbolic variables like &SYSNAME or &SYSPLEX are used for the main definitions.

***SYSPLEX - Multiple OS/390 Images, Sysplex**

This sample is a four system configuration defined as a sysplex group with two subgroups of two systems each. Each system has the standard application set linked to it. All mandatory sysplex-specific details are defined, as well as sysplex *ApplicationGroups* and sysplex *Applications*. A coupling facility is also defined.

As with the *MULTISYS sample, processors, communications path, and a communications task are defined.

Also included is a sample class definition of processor operations including IXC message automation.

System symbolic variables like &SYSNAME or &SYSPLEX are used for the main definitions.

***PRODUCTS - Samples for CICS, DB2, IMS, OPC Products**

This sample policy database has been derived from the *SYSPLEX sample. It also contains a sysplex with up to four systems and additionally contains the definitions of the following four products:

- for CICS: Class/Instance definitions for five CICS versions/types.
- for IMS: Class/instance definitions for IMS version 6/7 for all the different region types.
- for OPC: Class/Instance definitions for OPC 2.3 CONTROLLER/TRACKER/SERVER.
- for DB2: Class/Instance definitions for DB2 for all the different region types.

Using the Customization Dialog

Table 3 lists the application names that form the *standard application set*.

Table 3. Standard Application Set List

Application Name	Description
APPC	APPC/MVS automation policy
ASCH	ASCH (the APPC/MVS transaction scheduler) automation policy
GMFHS	Graphic Monitor Facility Host Subsystem
IO_OPERATIONS	I/O operations functions main component
JES2	JES2 automation policy
JES3	JES3 automation policy
LLA	Link-list lookaside
NETV_SA	SA OS/390 Automation NetView
NETV_SSI	NetView subsystem interface
PROCESSOR_OPERATIONS	Processor operations functions
RMF™	Resource Measurement Facility
RMFMON3	RMF Monitor III
RODM	Resource Object Data Manager
TRANSIENT1 TRANSIENT2	These are samples of <i>transient</i> applications which have special properties for automation.
TSO	This is generally a required application as well as being an example of a VTAM dependent one.
VLF	Virtual library facility
VTAM	VTAM

The standard application set contains a number of applications that are required to be started in order for SA OS/390 to be fully functional. The applications are:

- APPC
- ASCH
- PROCESSOR_OPERATIONS
- IO_OPERATIONS

The sample policy databases that link the standard application set contain the information that is required to automate the startup (and shutdown) of the SA OS/390 components and their prerequisite functions.

Using Existing Databases

If you have already created one or more policy databases, they are listed in the *Policy Data Base Selection* panel. Databases created with the NEW command automatically appear in your policy database list. In addition you can add databases created with other user IDs using the ADD command. 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.

Using the Customization Dialog

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 8 on page 25.

(N) NEW

This option lets you create a new policy database as described in “Creating a New Policy Database” on page 17. The *Create a New Policy Database* panel is displayed.

(E) EDIT

This option lets you change the data set name of your policy database and the build output data set name. To change the data in the database itself, you must OPEN it.

(D) DELETE

This option lets you delete the policy database data set, and all references to the policy database. SA OS/390 will display a panel asking you to confirm your request.

SA OS/390 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 OS/390 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 OS/390 data set.

(B) BUILD

This option displays the *Build Functions Menu*. You can build various configuration files. This function is explained in “Chapter 6. Building and Distributing Control Files” on page 289.

(G) MIGRATE

This option will bring you to the panel shown in Figure 208 on page 311 which allows you to migrate an automation control file to a policy database of the current release of SA OS/390.

You will find more information about this option in “Chapter 7. Migration Functions” on page 309.

Using the Customization Dialog

(MR) MERGE

This option merges a TSCF configuration into a policy database. SA OS/390 will display panel *Merge TSCF Configuration* shown in “Merge TSCF Configuration” on page 324.

(R) REPORT

This option displays the *Report Selection Menu* for a selected policy database. With this option, 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.

This function creates a member \$RPTPDB in the specified output data set, that needs to be processed by SCRIPT/VS.

Also, you can create a report for all objects in the policy database, that will not be used for building the system operations control files (unlinked entries and relationships).

Or, you can create a report for all automatically generated resources, which are the resources of type APL and APG.

For more information, refer to “Creating Reports about Policy Databases” on page 358.

(V) VIEW

This option toggles between two views. The summary view displays the policy database name and enterprise name for each policy database on one line. The extended view displays a second line for each policy database, giving the data set name.

(L) LOCATE

This option is entered only from the command line. Entering LOCATE lets you position yourself within the displayed list. If the entry is identical, the selected entry is the first in the list. If the entry is not identical, then the next higher (alphabetically) entry is the second in the list. This is the same as the ISPF LOCATE function.

Policy Information

This section and the remainder of this chapter contains information you need to know when working with whatever entry types in the *Entry Type Selection* panel in Figure 8 on page 25. For example some policy items are equal or some functions are available for all entry types.

Each major component of an SA OS/390 enterprise, and the enterprise itself, is described by a *policy object* of a given entry type. These policy objects correspond to the resources to be monitored or automated (application, system, processors) and the grouping of these resources (application group, Sysplex Application Group, subgroup, group, sysplex group). Figure 8 on page 25 shows the entry types definable by SA OS/390. From this panel you can choose an entry type and create policy objects of this entry type or edit automation policy of existing policy objects of the SA OS/390 enterprise. Automation policies (with the exception of communications tasks) apply to OS/390 systems only.

Note: The customization dialog allows concurrent update for entry types *Application* and *ApplicationGroup* (marked with an (*)) in Figure 8 on page 25).

Using the Customization Dialog

For other entry types, only one user at a time can access a policy database. See “Appendix A. Concurrent Customization by Multiple Users” on page 349 for further information.

Remember: Instances of a certain *entry type* are called *policy objects*.

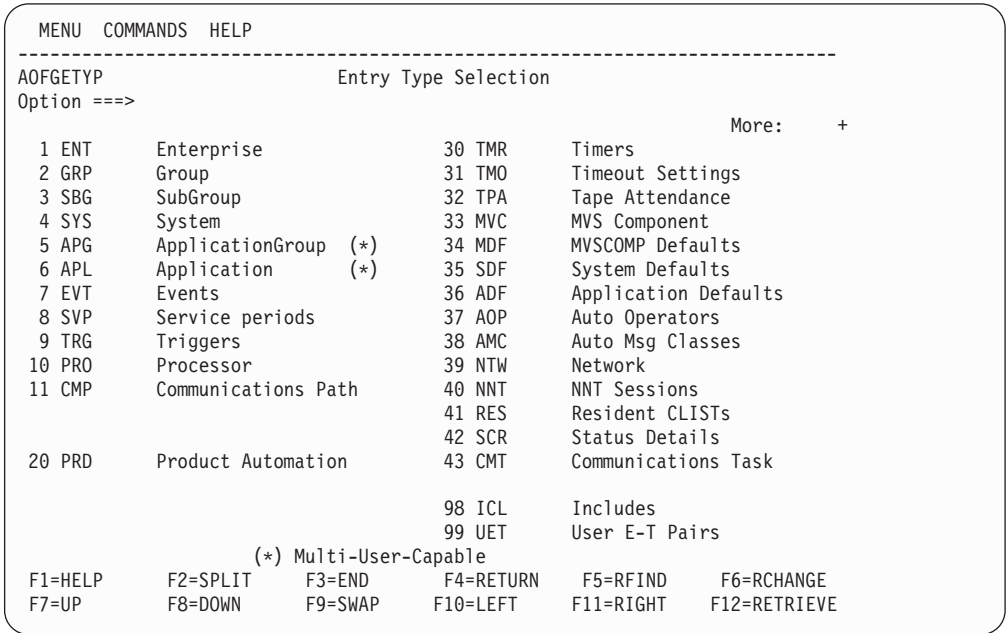


Figure 8. Entry Type Selection Panel

How to Create New Policy Objects

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

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 Group object, you type option 2 or *GRP* into the command line.
2. SA OS/390 displays the *Entry Name Selection* panel for the selected entry type. You can either select action *NEW* from the COMMANDS pull-down or type the command *new [entry_type_name]* into the panel’s command line. The entry type name is optional for this command. You can specify it later in the customization dialog.
3. SA OS/390 displays the *Define New Entry* panel for the selected entry type, where you now specify the required information. “Chapter 5. Defining Automation Policy” on page 87 contains several subchapters with more information on how to create new policy objects.

Delete Option on All Entry Name Selection Panels

Select the appropriate policy object from the *Entry Name Selection* panel, and select *Delete* from the ACTIONS pull-down menu. You need to confirm or cancel your deletion request.

If you want to delete an object that has links to one or more other objects, a panel is displayed that informs you of this fact and offers you to display a list of the referencing or referenced 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 possibility:

```
COMMANDS  HELP
-----
Delete Error
Command ==>

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

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

You can use the WHEREUSED function to list all the
referencing entries and, optionally, remove this
entry from each of them.

Display WHEREUSED list. . . YES 1      YES      NO
F1=HELP   F2=SPLIT  F3=END    F4=RETURN  F5=RFIND   F6=RCHANGE
F7=UP     F8=DOWN    F9=SWAP   F10=LEFT   F11=RIGHT
```

Entering YES here (**1**), will bring up the panel described in Figure 11 on page 29 where you can remove the application to be deleted from the application groups where it is contained.

Example 2:

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

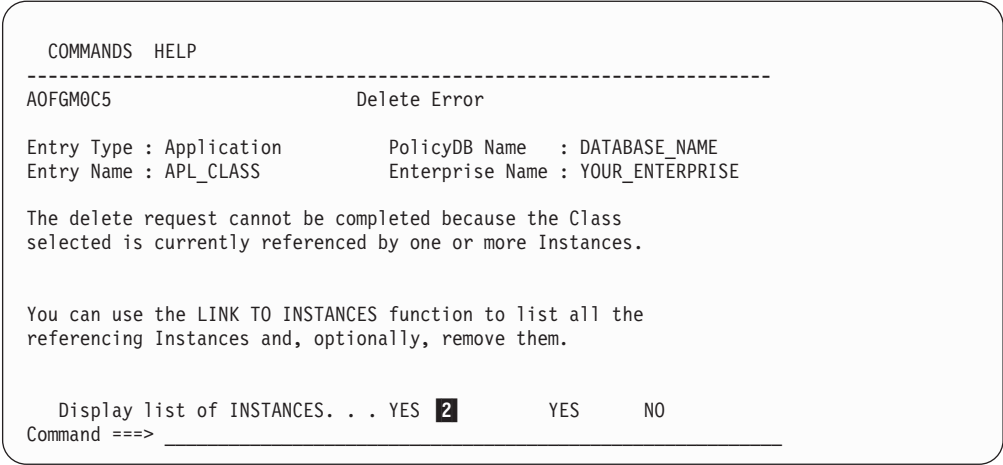


Figure 9. Request a List of Instances

Entering YES here (**2**), will bring up the panel shown in Figure 10.

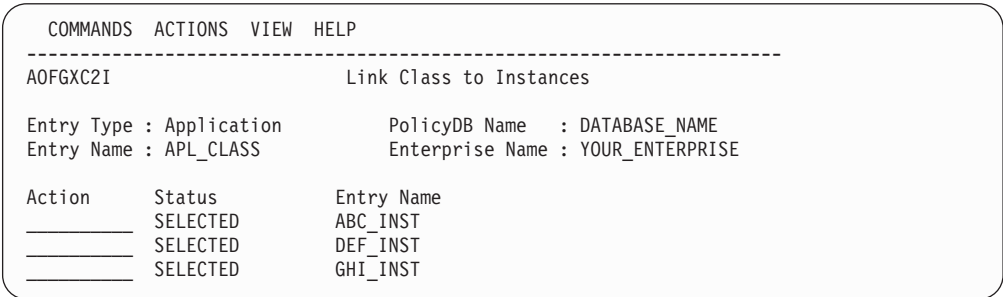


Figure 10. Links of Application Class to Instances

This panel (Figure 10) 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.

Policy Items Found on Many Selection Panels

The policy items described in this section are found in many policy definitions.

- **DESCRIPTION**

Both types of descriptions (short and extended, described below) 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.

Extended 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.

- **COPY**

Using the Customization Dialog

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 OS/390 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 OS/390 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 are returned to the *Policy Selection* panel for the new object. The **Copy complete** message shows that the copy has been successful.
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 and 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 92).
 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 "ApplicationGroup Policy Object" on page 132).
 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.
- **WHERE USED**

Using the Customization Dialog

When you select this option for a certain resource, SA OS/390 will display 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 OS/390 displays a list of all groups and subgroups in the SA OS/390 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 11).

Other examples for the usage of this policy item are:

- for *Service Periods*: to which applications are they linked?
- for *Processor*: which systems may run on this processor?
- for *System Defaults*: which OS/390 systems in the enterprise make use of this system defaults set.

COMMANDS	ACTIONS	VIEW	HELP

A0FGXWHU	Where Used		
Command ==>			SCROLL==> PAGE
Entry Type : Application		PolicyDB Name : DATABASE_NAME	
Entry Name : CICS		Enterprise Name : YOUR_ENTERPRISE	
Action	Status	Entry Name	Entry Type
	SELECTED	APG_FOC1	APG
	SELECTED	CICS_APG	APG
		CICS_APG2	APG
		COPY_APG	APG
		DEFAULTS	APG
		OPERATIONS	APG
		SAG_APPL	APG
		SYSTAPG	APG
		SYSTEM	APG

Figure 11. Where Used Panel

Chapter 3. Automation Concepts

This chapter describes various SA OS/390 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 36
- “Automation Flags” on page 42
- “How SA OS/390 Uses Error Thresholds” on page 39
- “Event and Trigger Support” on page 47

Goal Driven Automation

SA OS/390 implements goal driven automation.

One flavour of goal driven automation in SA OS/390 are events and triggers on the one hand and service periods on the other hand (see “Event and Trigger Support” on page 47 and “Automatic Calculation of Schedules (Service Periods)” on page 36). They basically allow you to define goals for applications or application groups (to be available or unavailable) and to specify external events that - additionally to the defined goals - need to be satisfied before an application/application group can be really started or stopped.

In SA OS/390 you can define the applications and application groups to be resources (so-called application resources and application group resource) that you want to automate.

Another flavour of goal driven automation is that automation programmers define the default behaviour or state of resources using the customization dialog so that SA OS/390 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 a request to start or stop it using the `INGREQ` command with appropriate startup or shutdown parameters. A request is executed by SA OS/390 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 OS/390.

Remember:

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 OS/390 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

Automation Concepts

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 OS/390 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 OS/390 to have the application group available.

Figure 12 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.

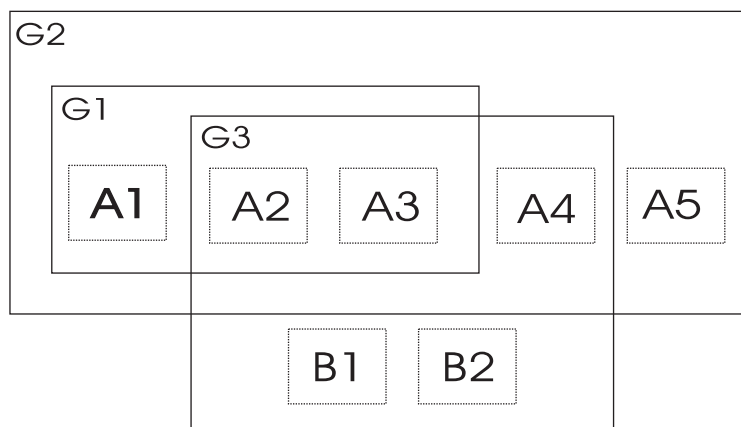


Figure 12. Example of Aggregating Applications into Application Groups

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 51.

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

- “Step 1: Identify Resources”
- “Step 2: Grouping the Resources” on page 33
- “Step 3: Nest Groups” on page 34
- “Step 4: Match Resources to Systems” on page 35

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: Grouping 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 Groups” on page 34).

Figure 13 on page 34 shows a possible aggregation of all components into groups:

Automation Concepts

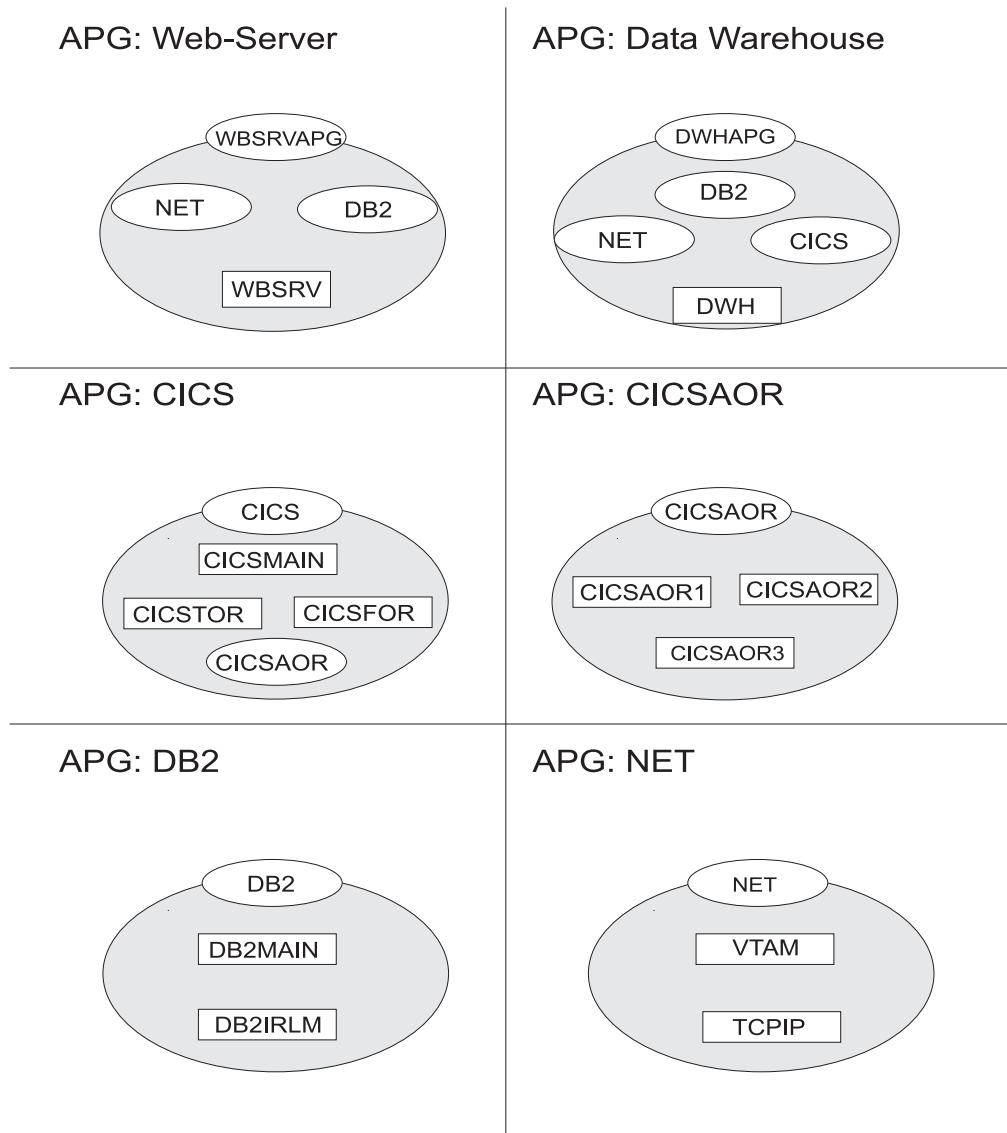


Figure 13. Application Groups

In Figure 13, 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 Groups

Figure 14 on page 35 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 like for example, *NET* and *DB2*.

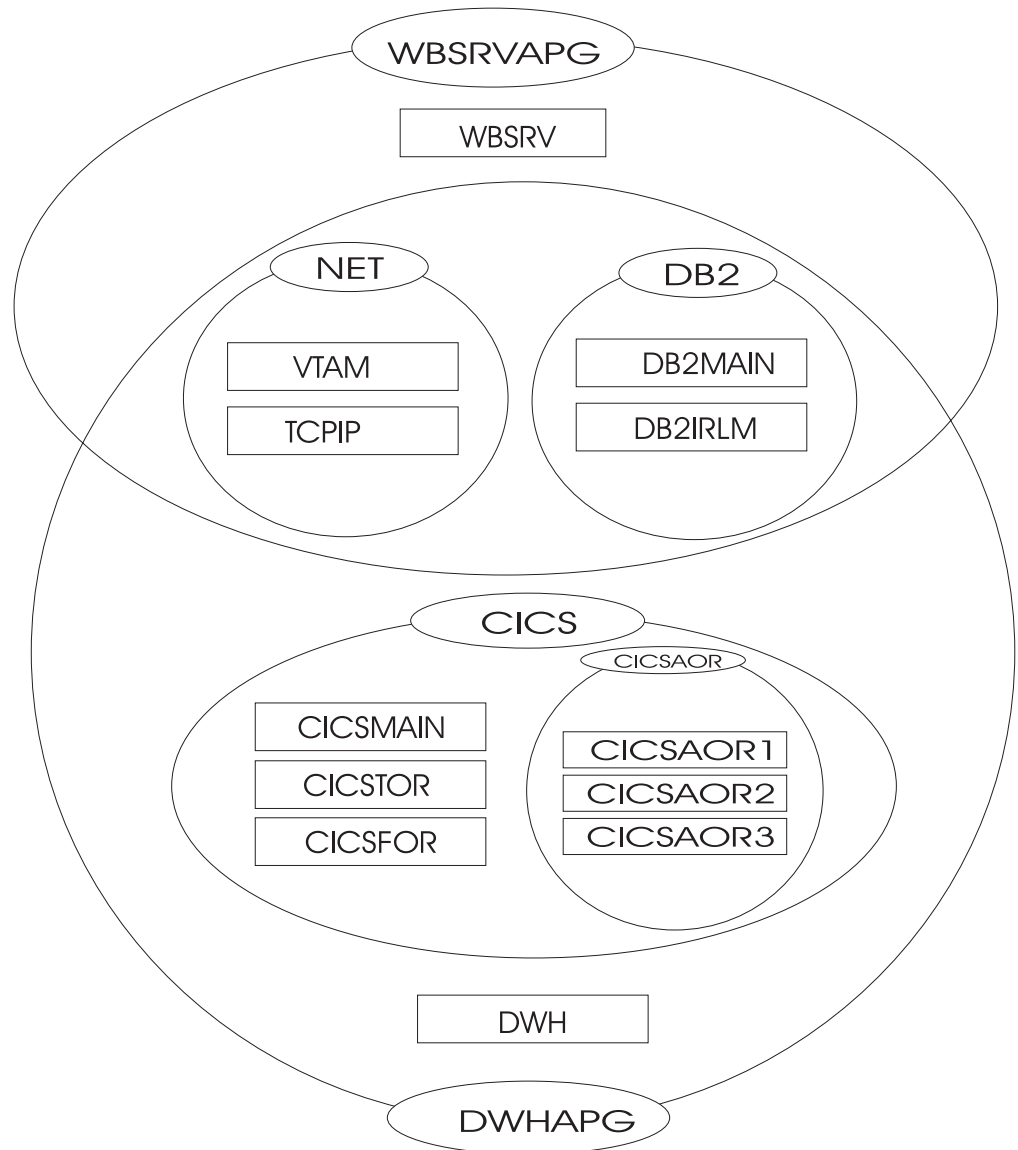


Figure 14. Grouping of Applications into Nested ApplicationGroups

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 OS/390 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.

Automation Concepts

- 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 51 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 or application groups. A service period is a set of service windows during which a resource (application or application group) 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 and application groups that they are linked to.

During the process of evaluating operator requests, SA OS/390 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 OS/390 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 OS/390 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 OS/390 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 OS/390 is the concept of *dependency relationships*. 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 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*. So a typical dependency

relationship that you define for a (dependent) resource *RES_DEP* would be: *MAKEAVAILABLE/WhenAvailable*, where the condition *WhenAvailable* applies to a supporting resource *RES_SUP* (see Figure 15). So the relationship defined in Figure 15 would read: Make the dependent resource *RES_DEP* available as soon as the supporting resource *RES_SUP* is available, but only if the desired goal for *RES_DEP* is: available.



Figure 15. Relationships and Conditions

So in the example from Figure 15, imagine that a request to become available is issued for *RES_DEP*. *RES_DEP* cannot be made available, unless *RES_SUP* is available. Now SA OS/390 implements the concept of request propagation. A request will be propagated along the dependency graph, so that finally the original request can be fulfilled. In our example, the request to *RES_DEP* will be propagated to *RES_SUP*, which can make itself available, because it is not dependent from anything else. Then, after *RES_SUP* is available, *RES_DEP* will become available. Thus the original request is persistent until it is withdrawn.

There is a special relationship *HASPARENT* that has been introduced to simplify relationship specifications. It is equivalent to a *MAKEAVAILABLE/WhenAvailable* dependency between the dependent and the supporting resource and a *MAKEUNAVAILABLE/WhenAssumedDown* dependency between the supporting resource to the dependent resource. A parent/child relationship from SA OS/390 1.3 will automatically be converted into a *HASPARENT* relationship from SA OS/390 2.1.

For each resource you should specify the appropriate relationship for starting and stopping it.

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 16 on page 38 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 16 on page 38.

Be careful not to define loops.

Automation Concepts

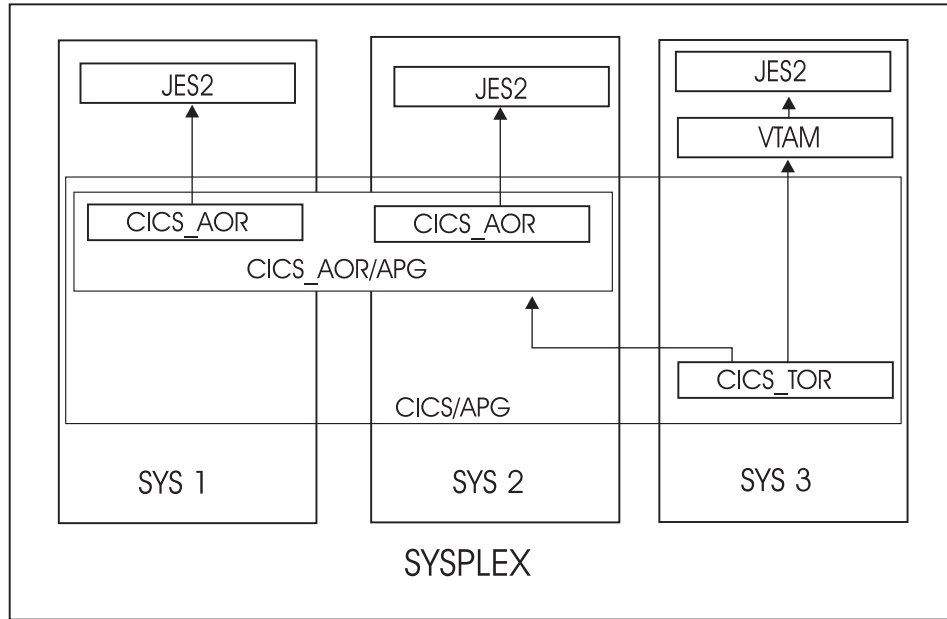


Figure 16. Relationship in a Sysplex

Figure 16 presents an example of grouping and relationships defined across a sysplex consisting of the systems *SYS1*, *SYS2* and *SYS3*.

In this figure, the arrows denote *HASPARENT* relationships. For example, *CICS_TOR* will be started as soon as one member of application group *CICS_AOR/CPG* is available.

If a group is a dependent resource, then all *MAKEAVAILABLE* relationships apply to the members of the group as well as to the group as a whole, for example, in Figure 17, resource **APL** will not start before resource **JES2** is available.

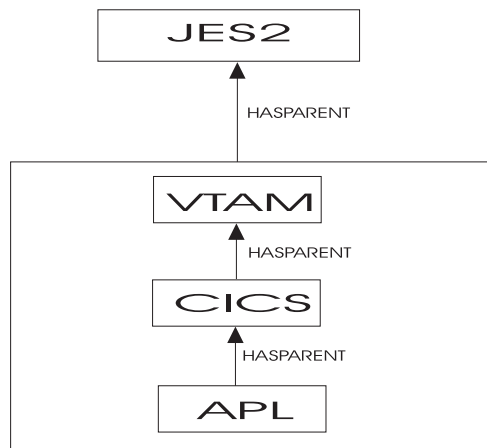


Figure 17. Relationship between Applications and Application Groups

For information on how to define relationships refer to “More About Policy Item RELATIONSHIPS” on page 157. During automation in your enterprise, conflicting relationships and requests could possibly occur. *System Automation for OS/390 User’s Guide* describes such a scenario and informs the operator how to handle such a situation.

How SA OS/390 Uses Error Thresholds

Error thresholds influence whether SA OS/390 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 OS/390 checks whether the number of occurrences of the condition reaches the critical threshold. If it is reached, SA OS/390 does not attempt restarting a resource. For applications, you can define a critical threshold for restarting these applications.

For OS/390 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 (command DISPTHRS) and to interactively set thresholds (Command INGTHRES) is provided in *System Automation for OS/390 User's Guide*.

Setting up a Threshold

SA OS/390 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 18 on page 40
- Application Defaults, you get a panel similar to Figure 18 on page 40
- MVSCOMP Defaults, you get a panel similar to Figure 18 on page 40
- Application, see Figure 19 on page 41
- MVS Component, see “Thresholds for MVS Components” on page 41

Thresholds can be displayed and modified using the INGTHRES command (see *System Automation for OS/390 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 OS/390 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 18 on page 40 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.

Automation Concepts

```

COMMANDS  HELP
-----
AOFPI1H1          Thresholds Specification
Command ==>>>

Entry Type : System Defaults      PolicyDB Name : DATABASE_NAME
Entry Name : SYSTEM_DEFAULTS_NAM Enterprise Name : YOUR_ENTERPRISE

Resource: System wide defaults

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

----- Levels -----
Resource          Critical      Frequent      Infrequent
                  Number  Interval    Number  Interval    Number  Interval
                  (hh:mm) (hh:mm)      (hh:mm) (hh:mm)      (hh:mm)
System wide defaults  4    01:00      3    02:00      2    12:00

```

Figure 18. 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 OS/390 notifies the operator and logs a message in the NetView log. When the number of occurrences of a situation reaches the critical threshold, SA OS/390 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 OS/390 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 OS/390 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 OS/390 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 OS/390.
- 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 OS/390 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 manyabend 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 OS/390 will display a panel as shown in Figure 19.

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.

```

COMMANDS  HELP
-----
AOFPIZTH          Thresholds Definition
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.

----- Levels -----
Resource          Critical          Frequent          Infrequent
                  Number Interval    Number Interval    Number Interval
                  (hh:mm)          (hh:mm)          (hh:mm)
APPC              4      01:00          3      02:00          2      12:00

```

Figure 19. Thresholds Definition Panel

Thresholds for MVS Components

You can define thresholds for components or functions of OS/390:

- MVSDUMP: for OS/390 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 OS/390 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.

Automation Concepts

```
COMMANDS  HELP
-----
AOFPIVD          Component Definition: MVSDUMP
Command ==>>>

Entry Type : MVS Component      PolicyDB Name  : TEST_PDB
Entry Name  : MVS_COMPONENT     Enterprise Name: ALL_PRIMARY

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

----- Threshold Levels -----
Resource          Critical          Frequent          Infrequent
Number Interval   Number Interval   Number Interval
                  (hh:mm)          (hh:mm)          (hh:mm)
MVSDUMP           3      01:00           2      01:00           1      01:00
```

Figure 20. Component Definition: MVSDUMP Panel

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 20, if a certain error occurs once per hour, it is considered to be infrequent. However, if an OS/390 dump (MVSDUMP) runs full more often than three times an hour, the critical stage is reached. In such a case, SA OS/390 will no longer recover the dump, but will notify the operator of this occurrence.

Automation Flags

One of the main tasks of customizing SA OS/390 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 OS/390 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 OS/390 option allowing a human operator to monitor and verify the flow of an automated operation.

SA OS/390 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” on page 43

- “Setting Automation Flags” on page 45
- “Relationship Between Automation Flags and Resource Status” on page 46
- “Automation Flags and Minor Resource Flags” on page 47

For more information on how to use automation flags and when SA OS/390 checks automation flags, refer to *System Automation for OS/390 Customizing and Programming*.

Types of Automation Flags

Figure 21 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 88 on page 147).

```

COMMANDS  ACTIONS  HELP
-----
A0FGFAS1          Flag Automation Specification
Command ==>>>

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

Resource: APPC
Enter level of automation desired.

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

Actions  Flag      Auto    Assist  Exits
Automation .          0
Recovery . .          0
Start. . . .          1
ShutDown . .          0
Initstart. .          2
Restart. . .          0

Enter or Display times to disable automation . . NO_____ Yes  No

```

Figure 21. Flag Automation Specification Panel

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

- **Automation**

Automation is the controlling flag for resource automation. During automated operations, SA OS/390 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 OS/390 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 OS/390 does not check any other flags.

The other flags which apply to each resource are:

- **Recovery**

This flag determines if SA OS/390 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

Automation Flags

off-loading. By automatically responding to error conditions, SA OS/390 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 flag 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 OS/390 initial status determination cycle. SA OS/390 conducts status determination cycles when:

- SA OS/390 is initialized after an IPL.
- SA OS/390 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 OS/390 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 OS/390 initialization, those applications that have a status of DOWN will be started if the Initstart flag setting is YES. SA OS/390 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 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 OS/390. By automatically responding to

the application requiring a restart, SA OS/390 automation decreases the time and system resources required to get an application up and running again.

Error thresholds influence whether SA OS/390 restarts resources. During a condition requiring restart, SA OS/390 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 OS/390 does not attempt restarting a resource. See “How SA OS/390 Uses Error Thresholds” on page 39 for more information.

The *Exits* column 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 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)
- *MVSCOMP 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 OS/390 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.

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 OS/390 Programmer's Reference* for more information.

The benefit of this automation hierarchy is that you can customize SA OS/390 to automate as many activities and resources as necessary in the OS/390 environment. You can choose to have SA OS/390 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.

Automation Flags

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 OS/390 provides an exit capability for automation flag processing. See *System Automation for OS/390 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 OS/390 User's Guide* or *System Automation for OS/390 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 only be viewed by the user, because it is set automatically by SA OS/390 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 OS/390 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 OS/390 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.

For information on which resource states for subsystems are used by SA OS/390 refer to *System Automation for OS/390 Customizing and Programming*.

Understanding how SA OS/390 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 OS/390 Customizing and Programming*.

SA OS/390 provides an exit for processing state changes. Refer to *System Automation for OS/390 Customizing and Programming* for more information.

Resource States and the Status Display Facility

SA OS/390 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 OS/390 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 OS/390 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 201
- “Triggers Policy Object” on page 205

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 OS/390 know that an event has occurred. One or more events are part of a trigger condition. You can define a condition to be either a *startup condition* or a *shutdown condition*.

Events are used to decide whether the goal for a resource should be realized. Even if all dependency relationships and conditions are fulfilled, but a resource’s startup condition is not fulfilled, because not all events of the startup condition are set, 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 OS/390 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 OS/390 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.

Event and Trigger Support

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

If a trigger is connected to an application group then, if the trigger startup condition for the application group is not satisfied, none of its members can be started. Also, if the trigger shutdown condition is not satisfied, 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.

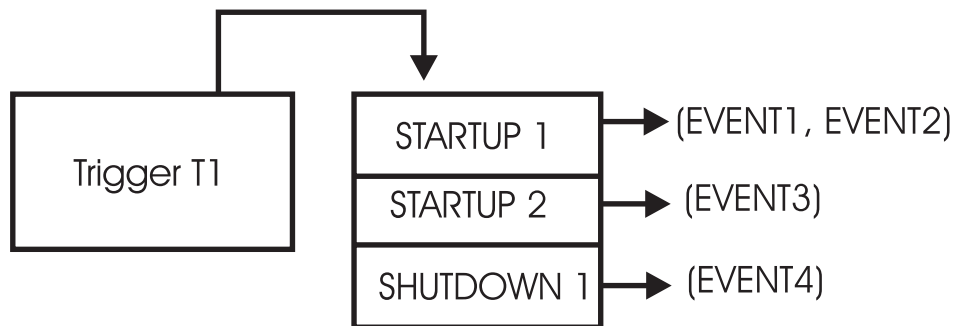


Figure 22. How a Trigger uses Events

So in Figure 22 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

So 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.

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

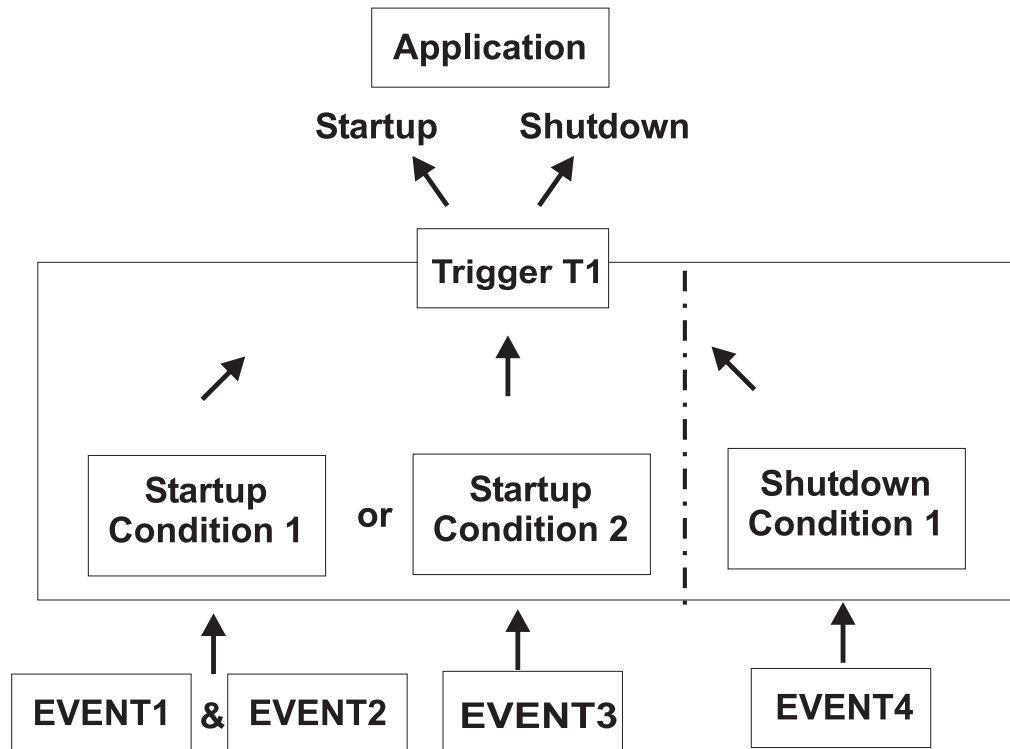


Figure 23. Triggers, Events and Conditions

Figure 23 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.

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

Event and Trigger Support

are set (and, for SA OS/390 1.3, the current time is outside of any service window for an eventually connected service period).

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

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 77
- “How to Use User E-T Pairs” on page 80
- “How to Implement Message Processing with User-Defined Data” on page 82

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 32.

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 53
- “Step 3: How to Create the Application Groups” on page 54

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 99 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.

Scenarios on How to Use the Customization Dialog

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 or APG are automatically generated when applications or application groups are linked to systems. Resources of other types are generated dynamically when the system operations control files are built.

Within the customization dialog, resources of type APL or APG 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 92 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 leave 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 51.
5. Select all three systems with an ‘S’ and press PF3 twice.
Now your environment contains a sysplex with three OS/390 systems.

Background:

Ways to Build a Sysplex

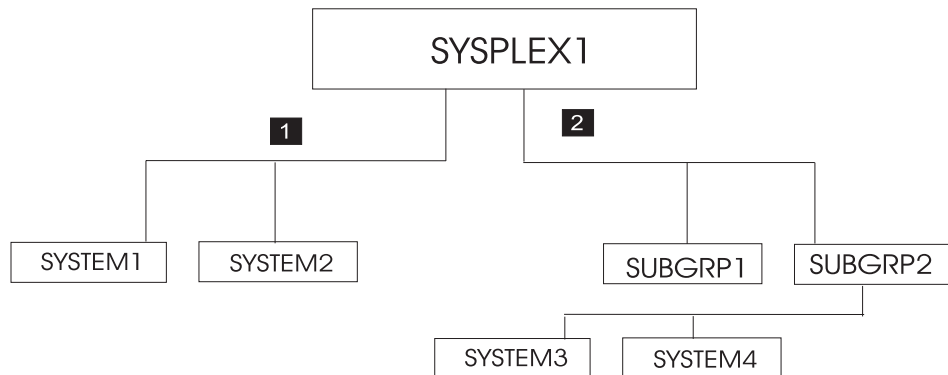
A Sysplex consists of OS/390 systems. In the customization dialog such a sysplex can be built by linking the (sysplex) group and its systems.

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).



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 141 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*.

Scenarios on How to Use the Customization Dialog

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.

Note that an application's subsystem name cannot be changed anymore.

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)
CICSMAN
CICSTOR
CICSFOR
CICSAOR1
CICSAOR2
CICSAOR3
DB2MAIN
DB2IRLM
TCPIP

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 behaviour of their members.

According to the example from "A Grouping Scenario" on page 32, 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.
- NET should run in the entire sysplex.
- CICS should run on SYSTEM3 and on SYSTEM2.

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 a sysplex.
- 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 133 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.

Scenarios on How to Use the Customization Dialog

Background:

Nature of an Application group

SA OS/390 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 ... 3) which all should be available. Therefore, you decide to take a BASIC Group.

5. Enter *CICSAOR* into the *Automation Name* field.

Background:

Application group Resource name

The *Automation Name* will be used as the first part of this application group's resource name. It is recommended to use the same name as the application group entry name and the Automation name for better identification of application groups and resources that will be derived from them.

An application group's Automation Name can be changed later on.

If no Automation Name is specified, no resource will be generated for the application group. Nevertheless, the application resources for all applications linked to this application group are created and will become available for automation.

6. Press PF3 to leave 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* pop-up window telling you that the resource has

Scenarios on How to Use the Customization Dialog

been created. This pop-up 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 CICSOR.

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
- if there is a link to a sysplex group.

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

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

The current configuration with application group CICSOR is now:

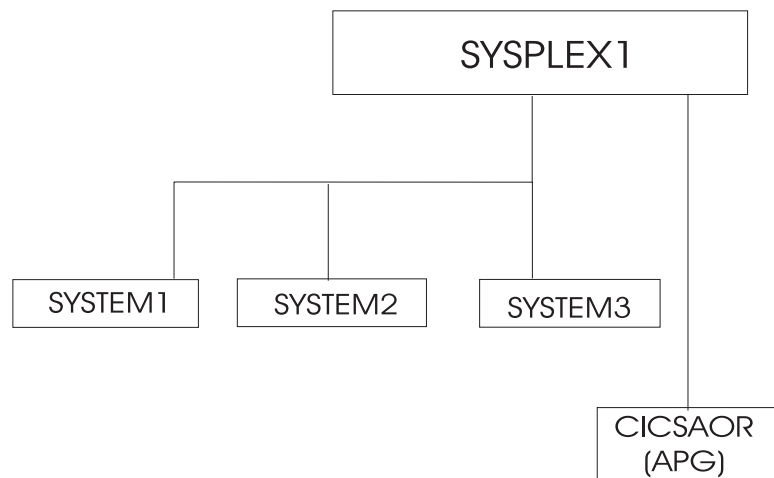


Figure 24. Current Configuration with Sysplex Application Group CICSOR

9. On the *Policy Selection* panel for CICSOR select the APPLICATIONS policy item.

The *Applications for ApplicationGroup* panel will be displayed with all twelve applications.

10. Select all of

CICSOR1
CICSOR2
CICSOR3

and press PF3 (END).

You are now back on the *Policy Selection* panel for CICSOR.

Scenarios on How to Use the Customization Dialog

Background: *Generating Application Resources*

Application Resources are generated when Applications are linked to Systems. This now is the case for CICSOR1, CICSOR2, CICSOR3: all now are linked to CICSOR, 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* pop-up window telling you that the resources have been created. For example, for CICSOR1, the following resources have been generated:

```
CICSOR1/APL/SYSTEM1  
CICSOR1/APL/SYSTEM2  
CICSOR1/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 CICSOR 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:

Scenarios on How to Use the Customization Dialog

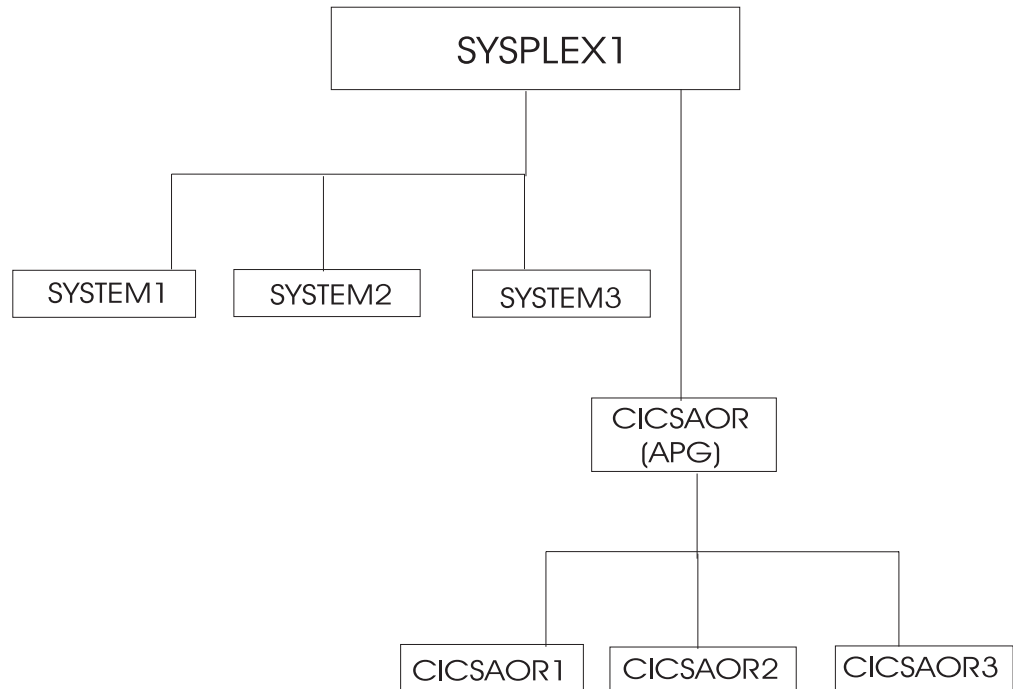


Figure 25. Current Configuration with Applications CICSAOR1...3

- 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 26.

```

COMMANDS  ACTIONS  HELP
-----
AOFGARS3          Select Resources
Command ==>>>          SCROLL==>> CSR

Entry Type : ApplicationGroup   PolicyDB Name : SCENARIO
Entry Name : CICSAOR           Enterprise Name : SCENARIO

Number of selected Resources . . . : 9
Behaviour. . . . . : ACTIVE

Update the following information:

Show valid APG-Resources . . . : NO      YES NO
Show selected Resources only . . : NO      YES NO
Show only Resources with String:

Action   Status   Resource Name
-----
         SELECTED  CICSAOR1/APL/SYSTEM1
         SELECTED  CICSAOR1/APL/SYSTEM2
         SELECTED  CICSAOR1/APL/SYSTEM3
         SELECTED  CICSAOR2/APL/SYSTEM1
         SELECTED  CICSAOR2/APL/SYSTEM2
         SELECTED  CICSAOR2/APL/SYSTEM3
         SELECTED  CICSAOR3/APL/SYSTEM1
         SELECTED  CICSAOR3/APL/SYSTEM2
         SELECTED  CICSAOR3/APL/SYSTEM3
  
```

Figure 26. Resources for the CICSAOR Sysplex Application Group

- 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

Scenarios on How to Use the Customization Dialog

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*.

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 behaviour 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 behaviour 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 Sysplex Application Group CICS

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

1. On the *Entry Type Selection* panel, enter option 5 or APG. To create the CICS Sysplex Application Group you have to repeat steps 1 through 10 from "Step 3.1: Create Sysplex Application Group CICSAOR" on page 55 accordingly.

On the *Applications for ApplicationGroup* panel, select applications

CICSMMAIN
CICSTOR
CICSFOR

On the *Select Resources* panel, deselect the generated resources on SYSTEM1.

Scenarios on How to Use the Customization Dialog

When you are done, the *Select Resources* panel should look like Figure 27 .

```
COMMANDS  ACTIONS  HELP
-----
A0FGARS3          Select Resources
Command ==>>>          SCROLL==>>> CSR

Entry Type : ApplicationGroup      PolicyDB Name : SCENARIO
Entry Name : CICS                  Enterprise Name : SCENARIO

Number of selected Resources . . . : 6
Behaviour. . . . . : ACTIVE

Update the following information:

Show valid APG-Resources . . . : NO      YES NO
Show selected Resources only . : NO      YES NO
Show only Resources with String:

Action   Status   Resource Name
         SELECTED CICSFOR/APL/SYSTEM1
         SELECTED CICSFOR/APL/SYSTEM2
         SELECTED CICSFOR/APL/SYSTEM3
         SELECTED CICSMAIN/APL/SYSTEM1
         SELECTED CICSMAIN/APL/SYSTEM2
         SELECTED CICSMAIN/APL/SYSTEM3
         SELECTED CICSTOR/APL/SYSTEM1
         SELECTED CICSTOR/APL/SYSTEM2
         SELECTED CICSTOR/APL/SYSTEM3
```

Figure 27. Resources for the CICS Sysplex Application Group

2. On the *Select Resources* panel from Figure 27, now change the field *Show valid APG-Resources* to YES. Then all existing APG-Resources on SYSTEM1, SYSTEM2 and SYSTEM3 and on Sysplex SYSPLEX1 are displayed, so CICSOR/APG will appear in the list.
3. Select CICSOR/APG.
Now seven resources are selected. You can check your resulting panel with Figure 28 on page 62.

Scenarios on How to Use the Customization Dialog

```

COMMANDS  ACTIONS  HELP
-----
AOFGARS3          Select Resources
Command ==>>>          SCROLL==>> CSR

Entry Type : ApplicationGroup      PolicyDB Name : SCENARIO
Entry Name : CICS                  Enterprise Name : SCENARIO

Number of selected Resources . . . : 7
Behaviour. . . . . : ACTIVE

Update the following information:

Show valid APG-Resources . . . : YES      YES NO
Show selected Resources only . . : NO      YES NO
Show only Resources with String:

Action   Status   Resource Name
-----
         SELECTED  CICSAOR/APG
         SELECTED  CICSFOR/APL/SYSTEM1
         SELECTED  CICSFOR/APL/SYSTEM2
         SELECTED  CICSFOR/APL/SYSTEM3
         SELECTED  CICSMAIN/APL/SYSTEM1
         SELECTED  CICSMAIN/APL/SYSTEM2
         SELECTED  CICSMAIN/APL/SYSTEM3
         SELECTED  CICSTOR/APL/SYSTEM1
         SELECTED  CICSTOR/APL/SYSTEM2
         SELECTED  CICSTOR/APL/SYSTEM3

```

Figure 28. Resources for the CICS Sysplex Application Group

4. 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.
5. 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, also start requests to application resources CICSFOR, CICSMAIN, CICSTOR (on SYSTEM2 and SYSTEM3) and to application group resource CICSAOR are automatically issued.

Step 3.3: 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 two application groups CICSAOR and CICS.
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.

Scenarios on How to Use the Customization Dialog

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* pop-up window telling you that two resources have been created:

DB2/APG/SYSTEM2
DB2/APG/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 29 on page 64.

Scenarios on How to Use the Customization Dialog

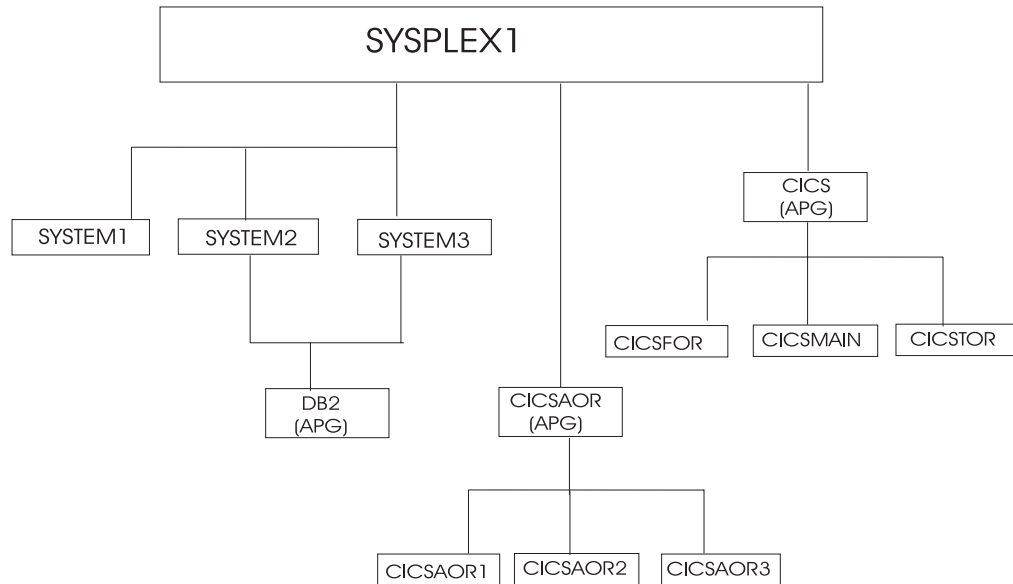


Figure 29. Current Configuration with Application group DB2

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.

So one resource is generated for each system-link of each application. The *Command Progress Display* pop-up window indicates that the following four resources are generated:

```
DB2IRLM/APL/SYSTEM2  
DB2IRLM/APL/SYSTEM3  
DB2MAIN/APL/SYSTEM2  
DB2MAIN/APL/SYSTEM3
```

Scenarios on How to Use the Customization Dialog

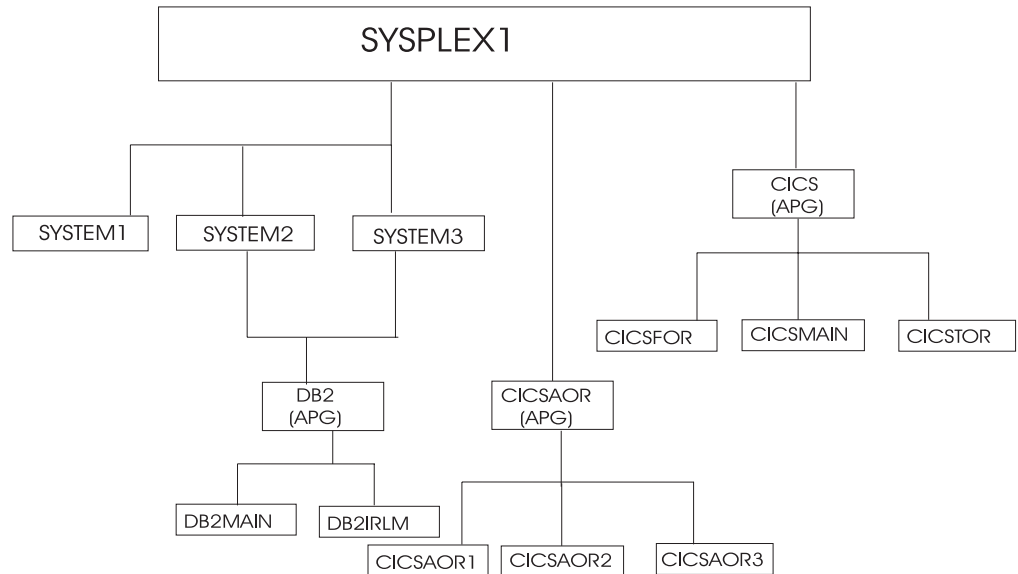


Figure 30. Current Configuration with applications linked to DB2

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

The *Select APG-Resource for Grouping* panel with two APG-Resources is shown. You can compare the resulting panel on your system with Figure 31.

```

COMMANDS  ACTIONS  HELP
-----
A0FGARS1      Select APG-Resource for Grouping
Command ==>>                                SCROLL==>> CSR

Entry Type : ApplicationGroup      PolicyDB Name : SCENARIO
Entry Name : DB2                  Enterprise Name : SCENARIO

Select APG-Resource(s) to modify group memberships

Action      Grouping-Resource      System      Selected Resources
DB2/APG/SYSTEM2      SYSTEM2      2
DB2/APG/SYSTEM3      SYSTEM3      2
  
```

Figure 31. Select APG-Resource for Grouping Panel for DB2 Application group

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 for which system you want to get the resources. This is done on the *Select APG-Resource for Grouping* panel by selecting one of the application group resources.

- On the *Select APG-Resource for Grouping* panel, select both APG-Resources
DB2/APG/SYSTEM2
DB2/APG/SYSTEM3

and press ENTER.

Scenarios on How to Use the Customization Dialog

The *Select Resources* panel with two APL-Resources for SYSTEM2 is displayed. You can compare the resulting panel on your system with Figure 32.

```

COMMANDS  ACTIONS  HELP
-----
A0FGARS3          Select Resources
Command ==>>>          SCROLL==>> CSR

Entry Type : ApplicationGroup      PolicyDB Name : SCENARIO
Entry Name : DB2                  Enterprise Name : SCENARIO

Number of selected Resources . . . : 2
Behaviour. . . . . : ACTIVE

Update the following information:

Show valid APG-Resources . . . : NO      YES NO
Show selected Resources only . . : NO      YES NO
Show only Resources with String:

Action   Status   Resource Name
-----
SELECTED DB2IRLM/APL/SYSTEM2
SELECTED DB2MAIN/APL/SYSTEM2

```

Figure 32. *Select Resources* Panel for DB2 Application group

Both applications should be started, thus no change on this panel is required.

11. On the *Select Resources* panel, Press PF3 (END) or CANCEL.

The *Select Resources* panel with two APL-Resources for SYSTEM3 is displayed (similar to the one shown in Figure 32). Both applications should be started, thus no change on this panel is required.

12. On the *Select Resources* panel with APL-Resources for SYSTEM3, press PF3 (END) twice now to return to the *Policy Selection* panel for the 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 complete now.

14. Now press PF3 until you are back on the *Entry Type Selection* panel.
15. Now the application group DB2 is complete: if a start request is issued against DB2 on SYSTEM2, also start requests to the application resources DB2MAIN and DB2IRLM are automatically issued for SYSTEM2. A start requests against application group DB2 on SYSTEM3 does the same for 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 33 on page 67.

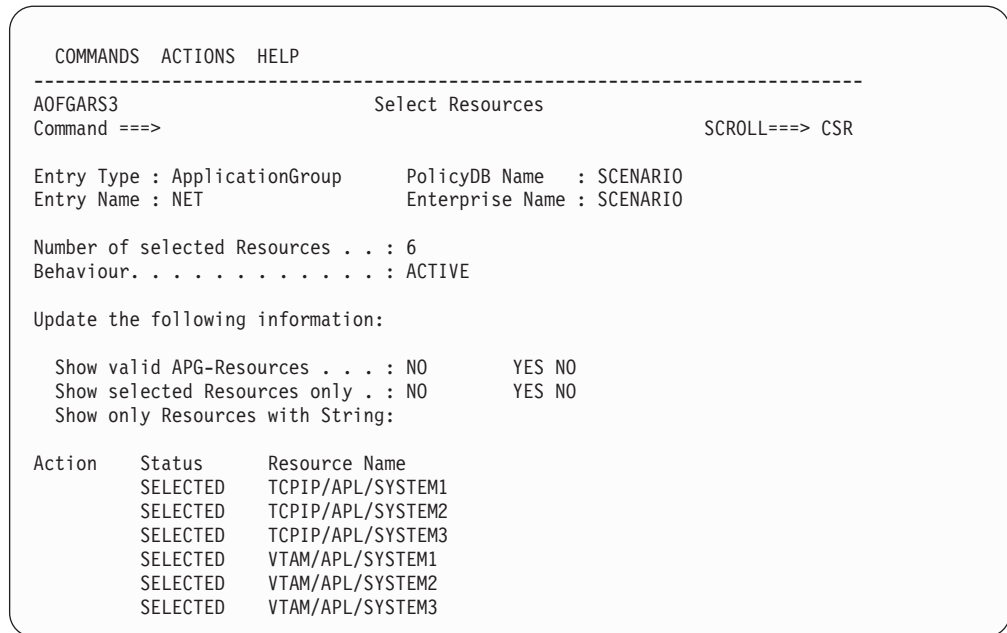


Figure 33. Select Resources Panel for Application group NET

Step 3.5: Create MOVE Group DWHAPG

The Data Warehouse Group DWHAPG should run on one system (SYSTEM1) and - if something goes wrong on SYSTEM1 - alternatively on SYSTEM2. So this application group will be 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 preparations 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 you leave the *Application Group Type* field at "SYSPLEX".

Scenarios on How to Use the Customization Dialog

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; so enter DWHAPG into the *Automation Name* field.

4. Press PF3 to leave the panel. The *Policy Selection* panel for Groups is displayed showing a message that the application group is 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* 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: you did not link any application 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 valid APG Resources* to YES.

The *Select Resources* panel will be displayed with all valid application group resources. Valid application group resources are Sysplex Application Group resources on the same sysplex (CICS/APG, CICSAOR/APG, NET/APG) and System Application Group resources on one of the systems in the Sysplex (DB2/APG/SYSTEM2 and DB2/APG/SYSTEM3). You can see this panel in Figure 34.

```

COMMANDS  ACTIONS  HELP
-----
AOFGARS2          Select Resources
Command ==>>>                                SCROLL==>> PAGE

Entry Type : ApplicationGroup      PolicyDB Name  : SCENARIO
Entry Name  : DWHAPG              Enterprise Name : SCENARIO

Number of selected Resources . . . : 0
Behaviour. . . . . : ACTIVE

Update the following information:
  Availability Target. . . . . 1      1 (for Move-APGs)

  Show valid APG-Resources . . . . YES  YES NO
  Show selected Resources only . . NO   YES NO
  Show only Resources with String.

Action   Preference  Resource Name
         Preference  CICS/APG
         Preference  CICSAOR/APG
         Preference  DB2/APG/SYSTEM2
         Preference  DB2/APG/SYSTEM3
         Preference  NET/APG
    
```

Figure 34. *Select Resources Panel for DWHAPG with valid APG-Resources*

However, the resulting panel from Figure 34 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 Sysplex Application Groups CICS/APG CICSAOR/APG and NET/APG are independent of such

Scenarios on How to Use the Customization Dialog

considerations. They must be started independently. The application group DB2 runs per definition on SYSTEM2 or SYSTEM3.

Our 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 cannot define these groups as System Application Groups because they need to have the Sysplex Application Groups CICS/APG, CICS/AOR/APG and NET/APG as members.

9. So press PF3 until you are back on panel *Entry Name Selection* and enter: NEW DWHAPG2 into the command line.

Or, as a fast path, specify =APG;NEW DWHAPG2.

The *Define New Entry* panel is displayed; the *Name* field is already filled with DWHAPG2.

10. Though DWHAPG2 should run on a single system, it needs to cross this system's boundaries due to its Sysplex Application Group components. Therefore, on the *Define New Entry* panel, you select type SYSPLEX.

You want to start all of DWHAPG2's components, thus leave its nature at BASIC.

For resource generation you need an automation name, so you enter DWHAPG2 into the related field and enter PF3 to leave the panel.

11. On the *Policy Selection* panel select the WHERE USED policy item and on the *Where Used* panel select sysplex group SYSPLEX1.
12. Press PF3 to leave the panel. You will see the *Command Progress Display* panel, telling you that the resource DWHAPG2/APG has been generated.
13. Select policy item APPLICATIONS. From the displayed list, select application DWH. Press PF3 to leave the panel.

As a result, the application resources

```
DWH/APL/SYSTEM1  
DWH/APL/SYSTEM2  
DWH/APL/SYSTEM3
```

are generated and linked into the application group resource DWHAPG2.

14. Select the RESOURCES policy item. The *Select Resources* panel with the three APL-Resources will be displayed.
15. You need to link the application group resource DWHAPG2, thus you change the field *Show valid APG-Resources* to YES.

As a result, six application group resources are displayed together with the three application resources (see also Figure 35 on page 70).

Scenarios on How to Use the Customization Dialog

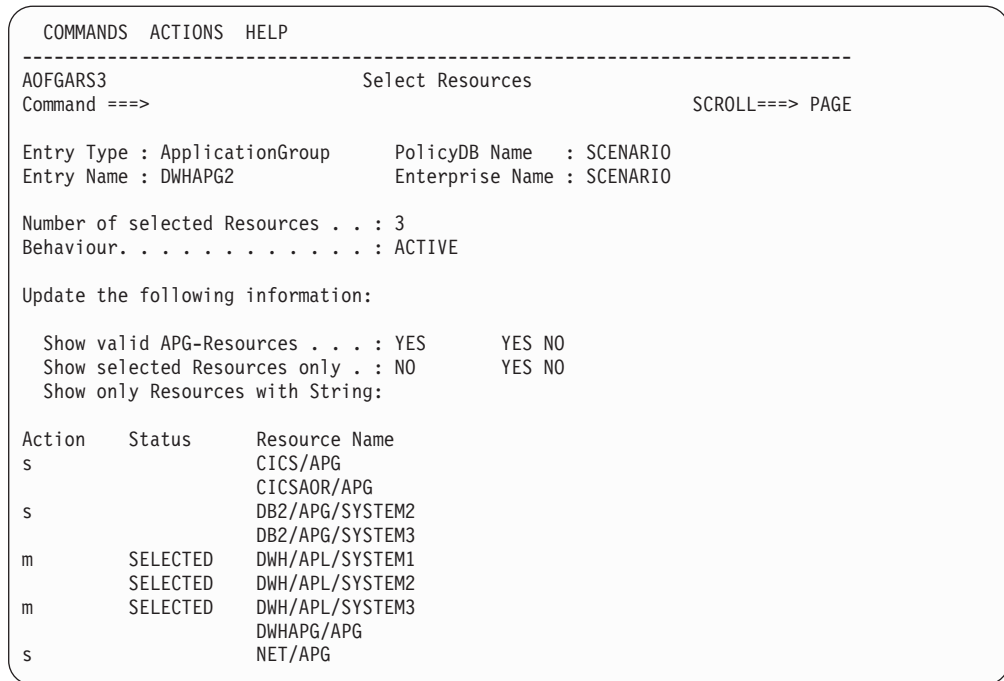


Figure 35. Select Resources Panel for DWHAPG2

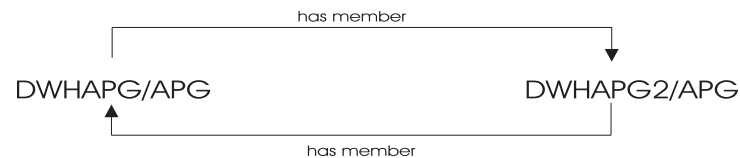
16. You want to use DWHAPG2 to start the application groups DWH and DB2 on SYSTEM2.
So you select CICS (CICS starts all CICS components), DB2 on SYSTEM2, DWH on SYSTEM2, and NET (see also Figure 35).

Background:

Defining Loops in Groups

When linking application group resources to application groups, you must be careful with your selections. You could also select CICSAOR/APG because it is a component you need. However CICSAOR will be started by CICS so you need not to care about it. (However selecting it here would do no harm because it would be started only once, either by control of DWHAPG2 or by control of CICS).

However, selecting DWHAPG/APG as member would finally result in a loop, when you link DWHAPG2/APG as member of DWHAPG/APG:



17. Change the field *Show selected Resources only* to YES to check the result. Then the *Select Resources* panel with one application resource and three application group resources selected is displayed (see Figure 36 on page 71).

Scenarios on How to Use the Customization Dialog

```

COMMANDS  ACTIONS  HELP
-----
A0FGARS3          Select Resources          SCROLL==> PAGE
Command ==>>>

Entry Type : ApplicationGroup      PolicyDB Name  : SCENARIO
Entry Name : DWHAPG2              Enterprise Name : SCENARIO

Number of selected Resources . . . : 4
Behaviour. . . . . : ACTIVE

Update the following information:

Show valid APG-Resources . . . : YES      YES NO
Show selected Resources only . : YES      YES NO
Show only Resources with String:

Action  Status  Resource Name
        SELECTED DWH/APL/SYSTEM2
        SELECTED CICS/APG
        SELECTED DB2/APG/SYSTEM2
        SELECTED NET/APG

```

Figure 36. Select Resources Panel for DWHAPG2

18. Press PF3 until you are back in the *Entry Name Selection* panel.
19. Now create application group DWHAPG3 in the same way as application group DWHAPG2 (as described in steps 9 on page 69 to 12 on page 69 and link it to Sysplex Application Group SYSPLEX1).
20. 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 upcoming *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 “Policy Items Found on Many Selection Panels” on page 27.

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

21. Select the RESOURCES policy item and link the resources for SYSTEM3. The result must look like Figure 36, with the following resources selected:

Action	Status	Resource Name
	SELECTED	DWH/APL/SYSTEM3
	SELECTED	CICS/APG
	SELECTED	DB2/APG/SYSTEM3
	SELECTED	NET/APG

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

22. Now you have all components for your Move Group DWHAPG, so select DWHAPG from the *Entry Name Selection* panel and then select its RESOURCES policy item.
23. *Select Resources* panel for DWHAPG is displayed.
Change the field *Show valid APG Resources* to YES.
The *Select Resources* panel with seven APG-Resources is displayed (see Figure 37 on page 72).

Scenarios on How to Use the Customization Dialog

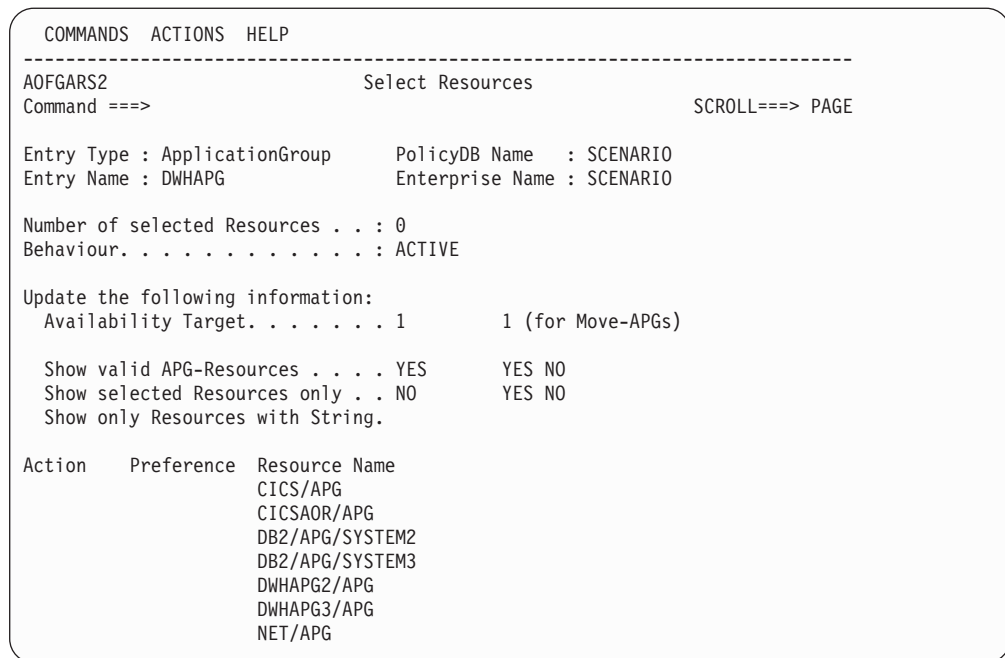


Figure 37. Select Resources Panel for DWHAPG with valid APG-Resources

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 SYSTEM 3. To define this, start with preference values 700 and 400. They have the following meaning:

- 700** Resource is started when the group is started. The resource is not stopped unless a very high preference (1000) alternative becomes available. (Use for the resource that normally runs).
- 400** 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 "Creating a New ApplicationGroup" on page 133.

24. Set preferences: Select the DWHAPG2/APG group on SYSTEM2 as primary (using action *P* in the *Action* column) and the DWHAPG3/APG group on SYSTEM3 as secondary (using preference value 400 in the *Preference* column). Figure 38 on page 73 shows the *Select Resources* panel with the selections

Scenarios on How to Use the Customization Dialog

made and Figure 39 on page 74 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 into *Preference* column or use the *Action* column for important values:

- PRIMARY...(P) - Sets "primary" preference = 700
- SECONDARY...(D) - Sets "secondary" preference = 400

```

COMMANDS  ACTIONS  HELP
-----
A0FGARS2          Select Resources
Command ==>>>                                SCROLL==>>> PAGE

Entry Type : ApplicationGroup      PolicyDB Name  : SCENARIO
Entry Name  : DWHAPG               Enterprise Name : SCENARIO

Number of selected Resources . . . : 0
Behaviour. . . . . : ACTIVE

Update the following information:
  Availability Target. . . . . 1          1 (for Move-APGs)

  Show valid APG-Resources . . . . YES    YES NO
  Show selected Resources only . . NO     YES NO
  Show only Resources with String.

Action   Preference  Resource Name
-----  -
         _____ CICS/APG
         _____ CICSAOR/APG
         _____ DB2/APG/SYSTEM2
         _____ DB2/APG/SYSTEM3
P        _____ DWHAPG2/APG
         400_       DWHAPG3/APG
         _____ NET/APG
  
```

Figure 38. *Select Resources Panel for DWHAPG with Selected Preferences*

Scenarios on How to Use the Customization Dialog

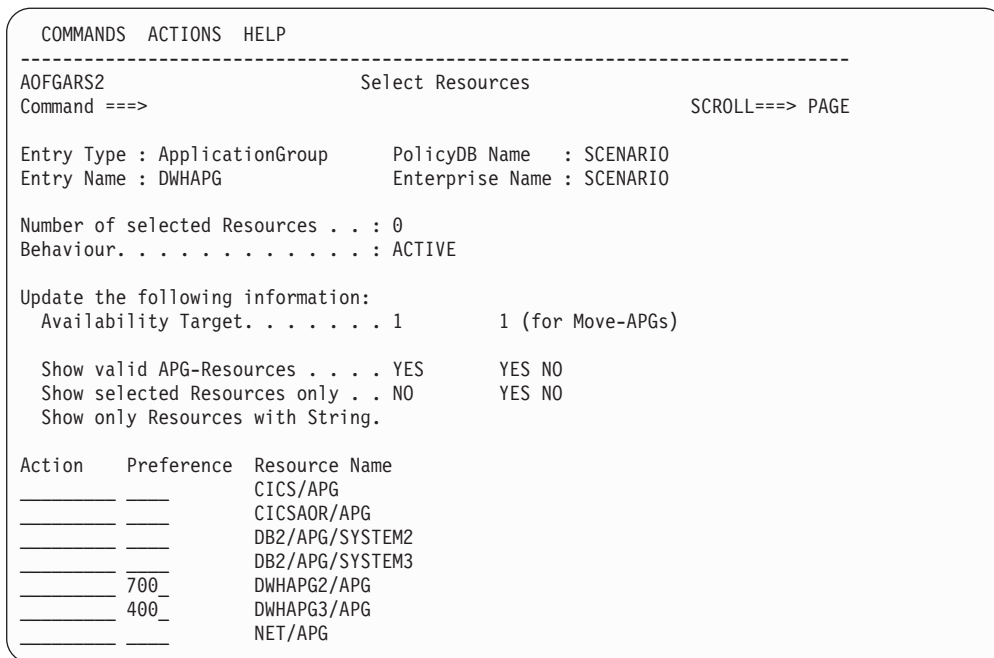


Figure 39. Select Resources Panel for DWHAPG with Converted Preferences

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

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

Scenarios on How to Use the Customization Dialog

The *Entry Name Selection* panel is displayed and shows seven application groups.

2. Enter NEW WBSRVAPG into 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*, you change the default preference to 444 as 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 40.

```

COMMANDS  HELP
-----
AOFGAN00          Define New Entry
Command ===>

To define a new entry, specify the following information:
Type . . . . . ApplicationGroup
Name . . . . . WBSRVAPG

Application Group Type . SYSPLEX          SYSTEM  SYSPLEX
Nature . . . . . SERVER                  BASIC MOVE SERVER
Default Preference . . . 444              0 to 1000, *DEF
-----

Automation Name . . . . WBSRVAPG          Name
Automatically link Application-Resources
  into APG . . . . . YES                  YES      NO

Behaviour. . . . . ACTIVE                  ACTIVE  PASSIVE

Short Description . . .
Extended Description . .
. .
. .
. .
. .

```

Figure 40. Define New Entry Input Panel for WBSRVAPG

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* pop-up window telling you that the resource WBSRVAPG/APG has been created.

Scenarios on How to Use the Customization Dialog

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/MVS1  
WBSRV/APL/MVS2  
WBSRV/APL/MVS3
```

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 field *Show valid APG-Resources* to YES, you get the complete list of selectable resources.

You 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:

- you set the Availability Target to "2",
- select mandatory components (WBSRV/APL/SYSTEM1 and NET/APG) with a value of 1000
- deselect WBSRV/APL/SYSTEM2 and WBSRV/APL/SYSTEM3 (action *m*)
- select DB2 (DB2/APG/SYSTEM2 and DB2/APG/SYSTEM3) with a value between 501 and 999.

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 switch the field *Show selected Resources only* to YES, your definitions should look like the sample screen shown in Figure 41 on page 77.

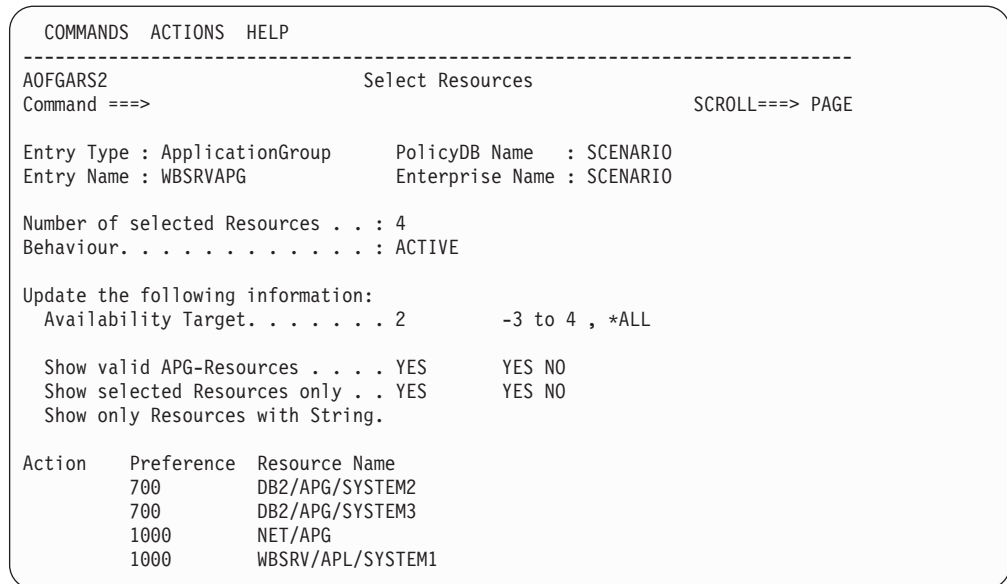


Figure 41. Final Select Resources Panel for WBSRVAPG

- The WBSRVAPG application group is complete now. Therefore, press PF3 in the *Select Resources* panel, select policy item APPLGROUP INFO in the upcoming *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 32. 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.

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

Scenarios on How to Use the Customization Dialog

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 OS/390 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 OS/390.

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 `AOR_MOVE/APG`

Scenarios on How to Use the Customization Dialog

- 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 OS/390 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 OS/390 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 205.
How to link a trigger to an application is described in chapter “Policy Items for Applications” on page 147 under policy item TRIGGER.
- Define a service period *SERV1* 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 203.

Now you are done. The definitions are illustrated in Figure 42 on page 80.

1. If the time of day is within the time frame defined by the service period *SERV1*, SA OS/390 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 OS/390 will start *TOR/APL/SYS3* because its parent is available.

Scenarios on How to Use the Customization Dialog

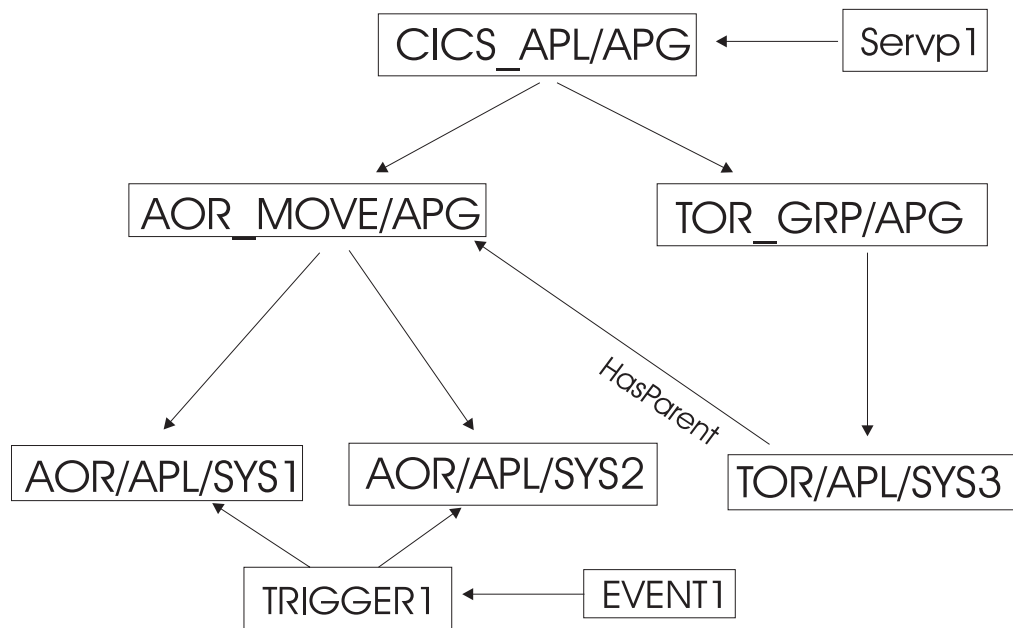


Figure 42. Scenario with Triggers, Events and Service Period

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 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 43 on page 81 is displayed.

Scenarios on How to Use the Customization Dialog

```

COMMANDS ACTIONS VIEW HELP
-----
A0FGUET          UET Entry-Type Selection          Row 20 from 40
Command ==>>>          SCROLL==>> PAGE

Entry Type : User E-T Pairs          PolicyDB Name : SAOS390
Entry Name : JOBCARDS                Enterprise Name : SCENARIO

Action  Entry Name          Type
E_____ JOBCARD          DEFAULTS
_____ JOBCARD          ICKDSF
_____ JOBCARD          IDCAMS
_____ JOBCARD          IMSFUNC
_____ JOBCARD          SENDMAIL

```

Figure 43. UET Entry-Type Selection Panel

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 OS/390 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 44.

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.

```

COMMANDS ACTIONS HELP
-----
A0FGUKD          UET Keyword-Data Specification          Row 48 from 130
Command ==>>>          SCROLL==>> PAGE

Entry Type : User E-T Pairs          PolicyDB Name : SAOS390
Entry Name : JOBCARDS                Enterprise Name : SCENARIO
Entry-Type : JOBCARD                DEFAULTS

Action  Keyword/Data(partial)
        JOBNAME
        HAIAC3*
        CLASS
        A
        MSGCLASS
        N
        MSGLEVEL
        (1,1)
        REGION
        4096K

```

Figure 44. 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.',

```

Scenarios on How to Use the Customization Dialog

```
' | 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  
JOB CARD DEFAULTS  
JOBNAME=HAIAOC3*  
CLASS=A  
MSGCLASS=N  
MSGLEVEL=(1,1)  
REGION=4096K
```

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

Example:

The HSM application issues the ARC0909E 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 OS/390 to automatically issue two commands to HSM, that will perform the recovery:

HOLD ALL

to stop all further activities for HSM

BACKVOL CDS

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 ARC0909E as shown in Figure 45 on page 83.

Scenarios on How to Use the Customization Dialog

```

COMMANDS  ACTIONS  HELP
-----
AOFPISSM          Message Processing
Command ==>>>                                SCROLL==>>> PAGE

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

Subsystem : HSM

Enter messages issued by this resource that will result in automated actions.
Actions: CMD = Command  REP = Reply  CODE = CODE  USER = User defined values

Action  Message ID                                Cmd  Rep  Code  User
-----  -
          ARC0909E_____
          HSM journal data set utilization_____
          _____
          _____
  
```

Figure 45. Message Processing Panel

To specify your user-defined data, now type **USER** into the **Action** column from the panel shown in Figure 45. This will lead you to the **User Defined Data** panel where you specify your input as shown in Figure 46.

```

COMMANDS  HELP
-----
AOFPISSU          User Defined Data
Command ==>>>                                SCROLL==>>> PAGE

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

Subsystem : HSM
Message ID : ARC0909E_____

To change keyword-data pair, specify the following:

Keyword
Data
LIMIT _____
97% _____
_____
_____
_____
_____
  
```

Figure 46. User Defined Data Panel

In your REXX exec that you will write in “Step 4: How to Write the REXX Exec” on page 84, you will compare the percentage value returned from the ARC0909E message with your value specified here, and, if it exceeds your specified value, you will cause SA OS/390 to issue the commands specified in “Step 3: How to Update the NetView Message Automation Table” on page 84. If one day you will decide to start your recovery already if 95% are exceeded, you need not edit your REXX exec, but only your customization dialog input and rebuild your system operations control files.

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

Scenarios on How to Use the Customization Dialog

Step 2: How to Specify the Commands

In the *Message Processing* panel (Figure 45 on page 83) now type CMD into the *Action*. This will lead you to the *CMD Processing* panel where you specify your input as shown in Figure 47.

```
COMMANDS  HELP
-----
AOFPISSC          CMD Processing          SCROLL==> PAGE
Command ==>

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

Subsystem : HSM
Message ID : ARC0909E_____

Enter commands to be executed when resource issues the selected message.

Pass/Selection Automated Function/'*'
Command Text
MVS F &SUBSJOB, HOLD ALL_____

MVS_F &SUBSJOB, BACKVOL CDS_____
```

Figure 47. *CMD Processing Panel*

Step 3: How to Update the NetView Message Automation Table

You need to specify an entry in your NetView message automation table that will trigger your REXX exec called YOUR_CLIST that you will write in “Step 4: How to Write the REXX Exec”. This entry should look like follows:

```
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 46 on page 83) 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 47), the ACFCMD command is used (**2**).

```
/* 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
```

Scenarios on How to Use the Customization Dialog

```
/* 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" , 1
" | 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
End
/* Automation not allowed */
When lrc < 3 Then Do
...
End
/* Resource not known to SA */
When lrc = 3 Then Nop
/* Error occurred */
Otherwise Do
...
End
End
Exit lrc
```

Scenarios on How to Use the Customization Dialog

Chapter 5. Defining Automation Policy

This chapter describes how to define all SA OS/390 policy objects. They are described in the following subchapters in the order they appear on the *Entry Type Selection* panel shown in Figure 48.

```

MENU  COMMANDS  HELP
-----
AOFGETYP          Entry Type Selection
Option ==>>

1 ENT  Enterprise          30 TMR  Timers
2 GRP  Group              31 TMO  Timeout Settings
3 SBG  SubGroup          32 TPA  Tape Attendance
4 SYS  System            33 MVC  MVS Component
5 APG  ApplicationGroup  (*) 34 MDF  MVSCOMP Defaults
6 APL  Application      (*) 35 SDF  System Defaults
7 EVT  Events           36 ADF  Application Defaults
8 SVP  Service periods  37 AOP  Auto Operators
9 TRG  Triggers         38 AMC  Auto Msg Classes
10 PRO Processor        39 NTW  Network
11 CMP Communications Path 40 NNT  NNT Sessions
                                     41 RES  Resident CLISTs
                                     42 SCR  Status Details
                                     43 CMT  Communications Task

20 PRD  Product Automation

                                     98 ICL  Includes
                                     99 UET  User E-T Pairs

(*) Multi-User-Capable
F1=HELP  F2=SPLIT  F3=END  F4=RETURN  F5=RFIND  F6=RCHANGE
F7=UP    F8=DOWN  F9=SWAP F10=LEFT  F11=RIGHT F12=RETRIEVE

```

Figure 48. Entry Type Selection Panel

For a quick retrieval of the single policy objects in this chapter you can use the following table:

Table 4. Quick Retrieval for Entry Types

Entry Type/Policy Object	Page
Enterprise Policy Object	88
Group Policy Object	92
SubGroup Policy Object	98
System Policy Object	99
ApplicationGroup Policy Object	132
Application Policy Object	141
Events Policy Object	201
Service Periods Policy Object	203
Triggers Policy Object	205
Processor Policy Object	208
Communications Path Policy Object	218
Product Automation Policy Object	229
Timers Policy Object	246

Defining Automation Policy

Table 4. Quick Retrieval for Entry Types (continued)

Entry Type/Policy Object	Page
Timeout Settings Policy Object	248
Tape Attendance Policy Object	250
MVS Component Policy Object	252
MVSCOMP Defaults Policy Object	260
System Defaults Policy Object	261
Application Defaults Policy Object	262
Auto Operators Policy Object	263
Auto Msg Classes Policy Object	269
Network Policy Object	270
NNT Sessions Policy Object	273
Resident CLISTs Policy Object	275
Status Details Policy Object	276
Communications Task Policy Object	279
Includes Policy Object	286
User E-T Pairs Policy Object	286

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

ACTIONS		HELP

AOFGEPOL	Policy Selection	
Command ==>>	SCROLL==>> PAGE	
Entry Type : Enterprise	PolicyDB Name	: DATABASE_NAME
Entry Name : YOUR_ENTERPRISE	Enterprise Name	: YOUR_ENTERPRISE
Action	Policy Name	Policy Description
	DESCRIPTION	Enter description
	SEND COMMAND OPERS	Define Operator Profile for sending commands
	INGSEND PARMS	Define INGSEND Command Parms
	PROCESSOR OPS INFO	Define processor operations focal point info

Figure 49. 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 refer to “More About Policy Item SEND COMMAND OPERS”.

INGSEND PARMS

This policy item is provided to allow you to migrate your INGSSEND usage away from the Gateways (OST/NNT) and toward the NetView RMTCMD command, which is the recommended command routing mechanism (CRM).

This policy item also allows you to specify the CRM to be used by INGSSEND when communicating with target domains. A wildcard character, *, can be used for NETID and LU. 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 OS/390 release and have changed your INGSSEND usage to be based entirely on RMTCMD.

RMTCMD-based INGSSEND 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 OS/390 default of ten seconds. CORRWAIT only applies if your command routing mechanism is RMTCMD.

PROCESSOR OPS INFO

This policy item allows you to designate primary and backup processor operations focal point names and the number of message monitor tasks and target control tasks for those focal point systems. For more information, refer to “More About Policy Item PROCESSOR OPS INFO” on page 91.

More About Policy Item SEND COMMAND OPERS

Figure 50 shows the panel that is displayed when the SEND COMMAND OPERS policy item is selected.

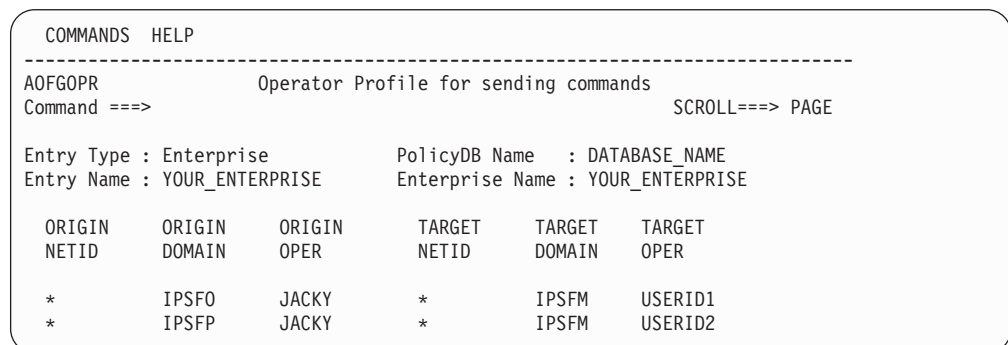


Figure 50. Operator Profile for Sending Commands

Enterprise Policy Object

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 OS/390 User's Guide* and *System Automation for OS/390 Operator's Commands*. For more information on RMTCMD, refer to *Tivoli NetView for OS/390 Automated Operations Network User's Guide*.

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 50 on page 89), 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.

DOMAIN

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 PROCESSOR OPS INFO

A panel as shown in Figure 51 is displayed if you select policy item PROCESSOR OPS INFO from the *Policy Selection* panel for enterprises. The panel contains the following fields:

```

COMMANDS  HELP
-----
ISQDPD02          Processor Operations Information
Command ==>>>

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

Enter the configuration information:

Primary Processor Operations focal point name KEY3
Backup Processor Operations focal point name .
Number of message monitor tasks . . . . . 1          0-999
Number of target control tasks . . . . . 1          1-999

Enter the fully qualified data set names for the Processor Control file
and Control file log:

Control file. . . . 'CONTROL.CTRF'
Control file log. . . 'CONTROL.CTRF.LOG'

```

Figure 51. Processor Operations Information Panel

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 OS/390 configuration. Names can contain 5 characters, starting with an alphabetic character, #, \$, or @. This field is required if you are using SA OS/390 processor operations functions. The customization dialog does not verify your entry.

Number of message monitor tasks

Enter the number of message monitor tasks to be defined for that focal point system. The valid range is from 0 to 999. For configurations with no processor operations PCs, specify a value of 0. For configurations with processor operations PCs, the minimum value is 1. The default value is also 1. There is no advantage in making the number of message monitor tasks greater than the number of communications tasks.

Number of target control tasks

Enter the number of target control tasks to be defined for that focal point system. The valid range is from 1 to 999. The number of target control tasks should be equal to the maximum number of active target hardware systems. The default value is 1.

Control file

Identifies the data set containing configuration information by processor operations under NetView. Enter the fully qualified name of the control file for SA OS/390 processor operations. The control file name cannot be DEBUG.

Control file log

Identifies the data set containing information about errors in the configuration definition. Enter the fully qualified name of the message log for the control file generation. 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.

Group Policy Object

A group is the largest subdivision of an SA OS/390 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 OS/390 processor operations functions. An SA OS/390 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 OS/390 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 25.

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 OS/390 sysplex. 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 52 on page 93 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 OS/390 Planning and Installation*).

```

COMMANDS  HELP
-----
AOFGGN00          Define New Entry
Command ==>>>

To define a new entry, specify the following information:
Type. . . . . Group
Name. . . . . Sysplex_Group

Group Type . . . . . SYSPLEX   STANDARD   SYSPLEX

ProcOps Commands . . . . NO           Group is valid for Processor
                                           Operations commands (YES|NO)

Short Description . . . This is an example sysplex group
Extended Description. .
. .
. .
. .
. .

```

Figure 52. Define New Entry Panel for a Sysplex Group

The unique fields are:

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.

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 OS/390 Operator's Commands* for information on 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 53 on page 94 shows the policy items that are displayed for a standard group.

Group Policy Object

```
ACTIONS  HELP
-----
AOFGEPOL          Policy Selection
Command ==>>>          SCROLL==>>> PAGE

Entry Type : Group          PolicyDB Name  : DATABASE_NAME
Entry Name  : STANDARD_GROUP Enterprise Name : YOUR_ENTERPRISE

Action  Policy Name      Policy Description
        DESCRIPTION    Enter description
        GROUP INFO     Display Group information
        SUBGROUPS      Select SubGroups for Group
        SYSTEMS        Select Systems for Group
-----
        COPY           Copy data from an existing entry
```

Figure 53. Policy Selection Panel for Groups

The unique policy items for groups are:

GROUP INFO

Group Info 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 indication.

SUBGROUPS

When you select this policy item, a list of all subgroups in the SA OS/390 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.

SYSTEMS

When you select this policy item, a list of all systems in the SA OS/390 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.

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 54 on page 95 is an example of the *Policy Selection* panel for Sysplex groups.

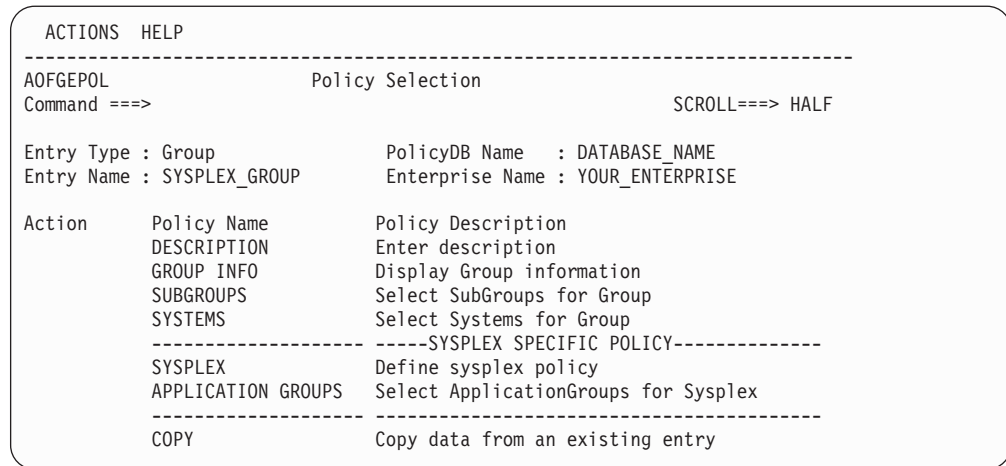


Figure 54. Policy Selection Panel for Sysplex Groups

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

SYSPLEX

Figure 55 on page 96 is an example of the *Sysplex Policy Definition* panel for sysplex groups.

This panel allows you to describe certain resources associated with the sysplex.

APPLICATION GROUPS

This policy item displays a list of all Sysplex Application Groups, indicating which application groups are selected for this sysplex. You can add or remove Sysplex Application Groups to or from this sysplex. A Sysplex Application Group can only be linked to one sysplex.

More About Policy Item SYSPLEX

A panel as shown in Figure 55 on page 96 is displayed if you select policy item SYSPLEX from the *Policy Selection* panel for sysplexes.

Group Policy Object

```

COMMANDS  HELP
-----
AOFGGPB0          Sysplex Policy Definition
Command ==>>>

Entry Type : Group          PolicyDB Name : DATABASE_NAME
Entry Name : SYSPLEX_GROUP_01 Enterprise Name : YOUR_ENTERPRISE

More:

Sysplex Timer Monitoring. . . . . YES  YES  NO
Number Monitored Sysplex Timers . . . 2   1  2

CDS High level Qualifier. . . . .

CDS type   Alternate volumes          Desired monitoring
          _____
Sysplex    _____                (PRIMARY ALTERNATE NONE)
          _____                PRIMARY
ARM        _____                PRIMARY
CFRM       _____                PRIMARY
LOGR       _____                PRIMARY
SFM        _____                PRIMARY
WLM        _____                PRIMARY

Logical SYSPLEX Group ID. . . _ Unique ID for this logical sysplex
                               (SA OS/390 Version 1.3 only)

```

Figure 55. Sysplex Policy Definition Panel for Sysplex Groups

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

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.

CDS High Level Qualifier

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 OS/390 data set naming rules. When allocating an alternate CDS, this qualifier is appended with the type (for example, SYSPLEX) and CDS0n, 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 OS/390 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 OS/390 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 (coloured 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

Note: The logical SYSPLEX Group ID is for SA OS/390 1.3 only.

A logical sysplex group ID may be specified to indicate that a subset of the members of an actual OS/390 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.

Group Policy Object

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*).

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 25.

Figure 56 is an example of the *Define New Entry* panel for subgroups.

```
COMMANDS  HELP
-----
AOFGBN00          Define New Entry
Command ==>>>

To define a new entry, specify the following information:
Type. . . . . SubGroup
Name. . . . . Subg1

ProcOps Commands. . . .NO  Subgroup is valid for Processor
                           Operations commands (YES|NO)

Short Description . . . This is an example subgroup
Extended Description. .
.
.
.
.
```

Figure 56. *Define New Entry Panel for a Subgroup*

The unique field 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 57 shows the policy items that are displayed when a subgroup is selected from the *Entry Name Selection* panel.

ACTIONS		HELP
AOFGEPOL		Policy Selection
Command ==>		SCROLL==> PAGE
Entry Type :	SubGroup	PolicyDB Name : DATABASE_NAME
Entry Name :	TARGET_SUBGROUP	Enterprise Name : YOUR_ENTERPRISE
Action	Policy Name	Policy Description
	DESCRIPTION	Enter description
	SUBGROUP INFO	Display Subgroup information
	SYSTEMS	Select Systems for SubGroup
	WHERE USED	List Groups linked to this entry
	COPY	Copy data from an existing entry

Figure 57. Policy Selection Panel for Subgroups

The unique policy items for subgroups are:

SUBGROUP INFO

When you select this policy item, information about subgroups is displayed for verification and updating.

SYSTEMS

When you select this policy item, a list of all systems in the SA OS/390 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 OS/390 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, TPF, 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 OS/390-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 25.

Figure 58 on page 100 shows the panel that is displayed when you create a new system from the *Entry Name Selection* panel for systems.

System Policy Object

If you want to monitor your newly created system as a stand-alone system on the NMC workstation, you need to specify an entry for this sysplex in the INGTOPOF file (see *System Automation for OS/390 Planning and Installation*).

```

COMMANDS  HELP
-----
AOFGSN00          Define New Entry
Command ==>>>

To define a new entry, specify the following information:
Type. . . . . System
Name. . . . . SYSTEM1

Operating system. . . MVS          MVS VM TPF VSE CF LINUX      More:  +

Specify information (MVS systems only):
MVS SYSNAME . . . . .          MVS system name

Clone Id. . . . .              &AOCCCLONE.
Clone Id 1. . . . .            &AOCCCLONE1.
Clone Id 2. . . . .            &AOCCCLONE2.
Clone Id 3. . . . .            &AOCCCLONE3.
Clone Id 4. . . . .            &AOCCCLONE4.
Clone Id 5. . . . .            &AOCCCLONE5.
Clone Id 6. . . . .            &AOCCCLONE6.
Clone Id 7. . . . .            &AOCCCLONE7.
Clone Id 8. . . . .            &AOCCCLONE8.
Clone Id 9. . . . .            &AOCCCLONE9.

Specify information for NMC Focal Point Communication (MVS systems only):
Heartbeat Interval. . . 5        1 - 60 (minutes)
Missing Heartbeat Delay 30      1 - 3600 (seconds)

Specify target system information for Processor Operations functions:
ProcOps name. . . . .          Processor Operations name

The following fields are for reference:
Short Description . . . _____
Extended Description. . _____
. . . _____
. . . _____
. . . _____
. . . _____

```

Figure 58. Define New Entry Panel for a System

The fields are:

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, processors, or communications paths defined in the current policy database. It must also be unique among ProcOps names, with the exception that this system's ProcOps name may match this system's entry name (see also the definition of entry field *ProcOps name* on page 102).

Operating system

A required identifier of the operating system to be run on the system. The allowed values are MVS, VM, TPF, VSE, LINUX, or CF. The supported default is MVS.

CF indicates the coupling facility control code. LINUX identifies the Linux for zSeries and S/390 operating system.

MVS SYSNAME

Required for OS/390 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 \$. The first three characters of MVS SYSNAME must not be "AOF" because these are reserved for internal SA OS/390 processing.

Note: The value of the MVS SYSNAME field for each OS/390 system under SA OS/390 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. Exception: for alternate configurations the specification of a 'plus' sign (+) as the last character is allowed. For more information on alternate configurations, refer to "Alternate Configuration Support" on page 361.
- May have numeric or national characters in any position.

Clone Id

You can enter up to ten clone IDs that can be used to substitute MVS job names, MVS automatic restart management element names, WLM resource names, and user-specified commands. For more information, refer to "Assigning Clone IDs" on page 102.

Heartbeat Interval

This specifies the time interval at which the SA OS/390 system operations functions sends a generic alert from this system to the NMC focal point system. The SA OS/390 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 OS/390 Heartbeat Function" on page 103. The heartbeat interval is a required field in the range 1 to 60 minutes that defaults to 5 minutes. Heartbeat information is relevant for OS/390 systems only.

The smaller the heartbeat interval, the faster SA OS/390 can detect a status forwarding path failure. For example, if you define an interval of two minutes then, on average, SA OS/390 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 OS/390 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 OS/390 Heartbeat Function" on page 103. 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 OS/390 systems only.

The smaller the missing heartbeat delay, the more sensitive SA OS/390 is to temporary status forwarding path delays. The longer the delay, the slower SA OS/390 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

System Policy Object

monitored, the larger the delay should be. This is to ensure that at resync time with peak NPDA alerts returning to the focal point, any NPDA alert processing delays do not cause unnecessary missing heartbeat processing.

ProcOps name

This specifies the name of the system as known to the processor operations functions. It is required. Like the system's Entry name, the ProcOps name must be unique among the names of all groups, subgroups, systems, processors, communications paths, and ProcOps names defined in the current policy database. However, the ProcOps name of this system may be the same as this system's entry name. The ProcOps name is 1 to 8 characters long and may consist of alphameric characters or @, #, or \$.

Assigning Clone IDs

If you are using SA OS/390 to automate an application that runs on more than one MVS system in the sysplex, you can assign up to ten clone IDs (&AOCCLONE. to &AOCCLONE9.) to each system. The value of the first clone ID (&AOCCLONE.) is automatically appended to the job name of a replicated application, unless you have specified that it or another clone value should appear elsewhere within the job name. In this way, each system can run with a unique job name while the application or subsystem shares a single automation policy across systems. Optionally, the substitution can occur anywhere within the job name and you can use any of the ten available clone IDs. You may use more than one clone ID if you want to.

Clone IDs can be used for job names, WLM resource names, and also to replicate applications that are being controlled by Automatic Restart Manager. You can use one clone ID for the job name and the WLM resource name, and then another clone ID for the Automatic Restart Manager element name. Or you can use various other combinations as required by your installation.

Note: You should 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.

The &AOCCLONE. function is not limited to use in sysplexes. In addition to qualifying the job name, you can use clone IDs to substitute anywhere within a defined literal, such as a shutdown command. The value of the clone IDs is also available to your automation through the NetView AOFAOCCLONEn (n = 1 to 9) common global variables. The AOFAOCCLONE and each AOFAOCCLONEn variable contains the value of the corresponding AOCCLONE ID.

SA OS/390 also supports the specification of a procname in addition to a job name. The normal startup command is:

```
MVS S jobname
```

but if a procname is specified, SA OS/390 issues:

```
MVS S procname,JOBNAME=jobname
```

Note: The job name is the result of the substitution of any applicable &AOCCLONE. value and will be the name used for all automation of the application.

To replicate an application, follow these steps:

1. Define your system, optionally assigning values to one or more of the clone IDs.
2. Define your application specifying YES for *Clone Job Name*. By default, this will append a tilde character, representing the first of the &AOCCLONE. values, to the end of the Job Name. Optionally, you can put the tilde in ANY position within the restrictions of IBM alphanumeric naming conventions, or you can substitute any other symbol using the &AOCCLONE n notation, where n is a number from 1 to 9.
3. Link the application through an application group to the system.
4. Build the automation policy for the system. The clone ID values are written out into the automation control file as a part of the system definition.
5. Build the application policy. If clone IDs are used in the job name, they are replaced with the appropriate &AOCCLONE. values.
6. Load the automation control file. The clone IDs are replaced with the appropriate symbol values.

The SA OS/390 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 OS/390 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 OS/390 topology manager tries to connect to another system in the sysplex. Only if all systems are not responding, SA OS/390 invokes the missing heartbeat processing.

SA OS/390 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 OS/390 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 OS/390 invokes its missing heartbeat processing for the remote system or the sysplex and changes the status of all associated resources to Unknown because SA OS/390 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. This is to ensure that at resync time with peak NPDA alerts returning to the Focal Point, any NPDA alert processing delays do not cause unnecessary missing heartbeat processing.

System Policy Object

When SA OS/390 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 59 shows the OS/390 policy items that are displayed when an MVS system is selected from the *Entry Name Selection* panel.

Note: Not all policies are displayed on one screen. Scroll forward to see the second part of this panel.

```

ACTIONS  HELP
-----
AOFGEPOL          Policy Selection
Command ==>>>                                SCROLL==>> PAGE

Entry Type : System          PolicyDB Name  : DATABASE_NAME
Entry Name  : SYSTEM1       Enterprise Name : YOUR_ENTERPRISE

Action      Policy Name      Policy Description
-----
DESCRIPTION  Enter description
SYSTEM INFO  Enter and display system information
APPLICATION GROUPS  Select ApplicationGroups for System
NETVIEW      Enter NetView-related information
AUTOMATION   Enter automation-related installed products
AUTOMATION CONSOLE  Enter MVS route codes for notifications
AUTOMATION SETUP  Define system environment for automation
AUTOMATION TIMERS  Select Timers for System
NNT SESSIONS  Select NNT Sessions for System
USER E-T PAIRS  Select User Entry-Type pairs for System
AUTOMATION TIMEOUT  Select Timeout Settings for System
RESIDENT CLISTS  Select Resident Clists for System
AUTO MSG CLASS  Select Auto Message Classes for System
TAPE ATTENDANCE  Select Tape Attendance for System
APPLICATION DEFAULTS  Select Application Defaults for System
SYSTEM DEFAULTS  Select System Defaults for System
MVSCOMP DEFAULTS  Select MVS Component Defaults for System
MVS COMPONENT   Select MVS Components for System
NETWORK        Select Network for System
AUTO OPERATORS  Select Automation Operators for System
SCREENS         Select screen status definitions
NOTIFY OPERATORS  Define forward messages to operators
DB2 SYSTEM DEFAULTS  Select DB2 System defaults for System
IMS XRF RESOURCES  Select IMS XRF resource names for System
IMS XRF STATUS FILES  Select IMS XRF Status files for System
OPC SYSTEM DETAILS  Select OPC System details for System
CONTROLLER DETAILS  Select OPC controller details for system
WORKSTATION DOMAINS  Select OPC Worstation domains for system
INCLUDES        Select Include members for System
-----POLICY FOR PROCESSOR OPERATIONS-----
PROCESSOR       Select Processor for System
PROCESSOR OPERATIONS  Enter Processor Operations information
-----
WHERE USED      List Groups/SubGroups linked to this entry
COPY            Copy data from existing entry

```

Figure 59. 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.

It also displays the current &AOCCLONE. values defined for the system. These values can be modified. Values you may find useful to define as &AOCCLONE. values are SYSCLONE, SYSNAME, and SUBAREA numbers.

Also, you can use this option to change the heartbeat interval and the missing heartbeat delay times. Refer to "System Policy Object" on page 99 for more information.

If you want to utilize sysplex message automation the processor operations target system name should be identical to the *MVS SYSNAME*

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.

NETVIEW

This policy item lets you enter the following details required for interacting 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 108.

AUTOMATION

This policy item allows you to document which automation products (RMF, OPC/ESA, SDSF) are operating on the system.

AUTOMATION CONSOLE

The automation console definitions are used when SA OS/390 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 109.

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 111.

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 246 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 273 for more information.

USER E-T PAIRS

This policy item displays a list of defined User E-T Pairs sets and allows

System Policy Object

you to choose which sets are to be associated with this system object. See “User E-T Pairs Policy Object” on page 286 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 248 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 275 for more information.

AUTO MSG CLASS

This policy item displays a list of Auto Msg Classes sets defined with entry type *Auto Msg Classes* and allows you to link such a set to the current system.

Classes of the specified set can be associated with a notify operator also defined for this system with policy item NOTIFY OPERATORS (see below). See “Auto Msg Classes Policy Object” on page 269 for more information on defining such message classes.

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 250 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 262 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 261 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 260 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 252 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 270 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 263 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 276 for more information.

NOTIFY OPERATORS

Notification operators are human NetView operators who receive SA OS/390 notification messages, which are messages about events SA OS/390 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. See the CLRHELD command in *System Automation for OS/390 Operator's Commands* for more information.

For more information refer to “More About Policy Item NOTIFY OPERATORS” on page 113.

DB2 SYSTEM DEFAULTS

If you select this policy item, SA OS/390 will display a list of DB2 system defaults which previously have been defined as described in “DB2 System Defaults” on page 230. 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 OS/390 only loading the last one processed.

IMS XRF RESOURCES

If you select this policy item, SA OS/390 will display a list of Extended Recovery Facility (XRF) resources which previously have been defined as described in “IMS XRF Resources” on page 238. 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 OS/390 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 237. 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 OS/390 will display a list of OPC system details which have been previously defined as described in “OPC System Details” on page 240. 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 OS/390 only loading the last one processed.

CONTROLLER DETAILS

If you select this policy item, SA OS/390 will display a list of OPC controller details which have been previously defined as described in “Controller Details” on page 242. 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

System Policy Object

be linked to a system, otherwise multiple entries will be built with SA OS/390 only loading the last one processed.

WORKSTATION DOMAINS

If you select this policy item, SA OS/390 will display a list of workstation domain IDs which have been previously defined as described in “Workstation Domains” on page 245. 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 OS/390 only loading the last one processed.

INCLUDES

If you select this policy item, SA OS/390 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 down-level SA OS/390 systems only and will no longer be supported.

PROCESSOR

Selection of this policy item displays a panel that lists all processors defined in the enterprise on which the system, as characterized by its operating system type, can run. You can select one processor to associate with the system.

PROCESSOR OPERATIONS

This policy item displays a series of panels that allow you to enter information needed by SA OS/390 processor operations. The first panel displayed is the *Processor Operations Target Definition* panel. The contents of the panels are determined by the processor associated with this system using the PROCESSOR policy item. This series of panels is described in section “More About Policy Item PROCESSOR OPERATIONS” on page 117.

More About Policy Item NETVIEW

The panel shown in Figure 60 on page 109 is displayed when you select policy item NETVIEW from the *Policy Selection* panel for Systems (Figure 59 on page 104).

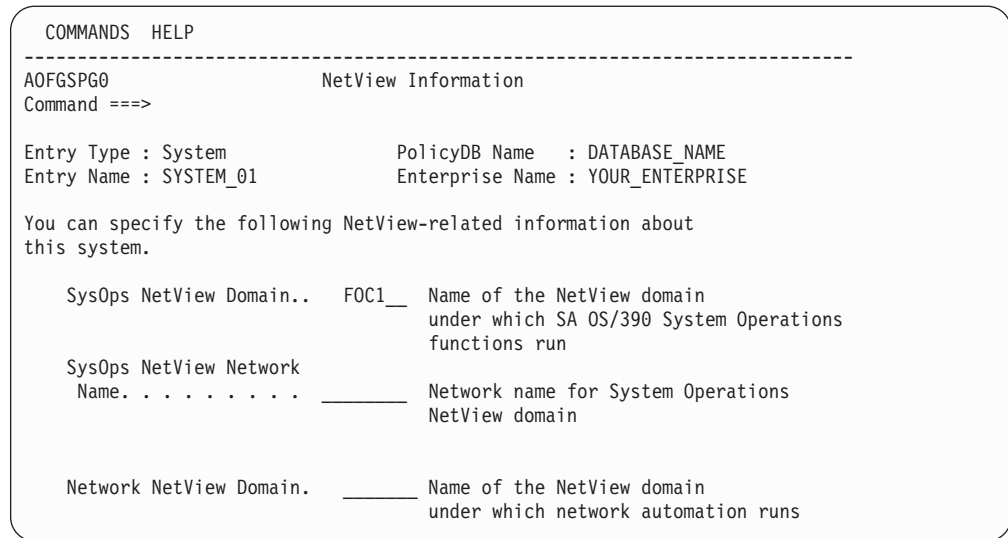


Figure 60. NetView Information Panel

The following information can be entered:

SysOps NetView Domain

This field is optional. It allows you to enter the NetView domain name of the NetView under which the SA OS/390 system operations functions runs on this system. This information is applicable only if you are using a dual NetView setup. The SA OS/390 NetView name can be up to five characters long.

SysOps 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 61 on page 110 is displayed when you select policy item AUTOMATION CONSOLE from the *Policy Selection* panel for Systems (Figure 59 on page 104).

System Policy Object

```
COMMANDS  HELP
-----
AOFPIEC0          Environment Definition: CONSOLE (Route Codes)
Command ==>>>

Entry Type : System          PolicyDB Name  : DATABASE_NAME
Entry Name  : SAMPLE_SYSTEM_01 Enterprise Name : YOUR_ENTERPRISE

Enter information and select S the route codes for SA OS/390 notifications.
Console ID ==> 01 or Console Name ==> _____

A Code    Definition
-----
All Display ALL messages
* 1 Master console action
  2 Master console information
  3 Tape pool
  4 Direct access pool
  5 Tape library
  6 Disk library
  7 Unit record pool
  8 Teleprocessing control
  9 System security
 10 System error, maintenance
 11 Programmer information
 12 Emulators
 13
 14
 15
 16 User

More:

Press ENTER to define MCSFLAGS
```

Figure 61. Environment Definition: CONSOLE (Route Codes) Panel

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 CONSOLE nn member of the OS/390 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 OS/390 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 OS/390 Customization: Using REXX and the NetView Command List Language*

More About Policy Item AUTOMATION SETUP

The panel shown in Figure 62 is displayed when you select policy item AUTOMATION SETUP from the *Policy Selection* panel for Systems (Figure 59 on page 104).

```

COMMANDS  HELP
-----
AOFPIES0          Environment Setup
Command ==>>>

Entry Type : System          PolicyDB Name  : DATABASE_NAME
Entry Name  : SAMPLE_SYSTEM_01 Enterprise Name : YOUR_ENTERPRISE

Enter the following information:

Primary JES          ==> JES2          Primary JES2/JES3 subsystem name
System Monitor Time ==> 00:30         Time between SYSTEM monitoring
                                     cycles (hh:mm or NONE)
Gateway Monitor Time ==> NONE         Time between GATEWAY monitoring
                                     cycles (hh:mm or NONE)
Monitor Option       ==> AOFAJMON     Default routine used to monitor
                                     application status
Message Table        ==> AOFMSG01     Netview message automation table members
MVS SYSNAME          ==> FOC1         The MVS SYSID in SYS1.PARMLIB
SDF Root Name        ==> FOC1         Root of this system's SDF tree
Exit name(s)         ==> SYSEXIT1,SYSEXIT2,SYSEXIT3,SYSEXIT4
                                     Environment setup user exit names
                                     (leave blank for no setup exit)

```

Figure 62. Environment Setup Panel

You can enter the following data:

Primary JES

The name of the primary JES. The named subsystem is the default scheduling subsystem.

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

System Policy Object

application monitor routine if Monitor Option is left blank is AOFAJMON. 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 150), then this routine overrides what you specify here. The following application monitor routines are valid:

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.

AOFAJMON

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 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.

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.

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.

Note: AOFAPMON should only be used as a programming facility because its only output is a return code.

AOFCPSM

This routine is a dedicated routine used to monitor the status of the SA OS/390 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 OS/390 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.

Note: AOFUXMON should only be used as a programming facility because its only output is a return code.

ISQMTSYS

With this routine, a processor operations target system resource represented by its proxy can be monitored. You find examples on how to use a proxy definition in *System Automation for OS/390 Customizing and Programming*. Active operator console connections are mandatory and will be used for sending an OS/390 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

Message Table

You may specify multiple message tables that will be loaded and used for message automation. The message tables defined here will replace the message table that was loaded during initialization. The default is AOFMSG01.

MVS Sysname

The name used by SA OS/390 for the system name for purposes of automation.

The MVS SYSNAME for each system under SA OS/390 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 OS/390 to map to the master include file built for the system. When SA OS/390 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 OS/390 Planning and Installation* for more information on how SA OS/390 determines the default automation control file.

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 OS/390 Customizing and Programming* for more information.

More About Policy Item NOTIFY OPERATORS

The panel shown in Figure 63 on page 114 is displayed when you select policy item NOTIFY OPERATORS from the *Policy Selection* panel for Systems (Figure 59 on page 104).

System Policy Object

```

COMMANDS  ACTIONS  HELP
-----
AOFPIN00          Operator Notification
Command ==>>>          SCROLL==>> PAGE

Entry Type : System          PolicyDB Name : DATABASE_NAME
Entry Name : SAMPLE_SYSTEM_01 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

A Operator      Operator      Notify  ----- Messages -----
  ID            Description          Information
                                     Eventual Action
                                     Immediate Decision
                                     System Wait
                                     Immediate Action
-  XXAOC05      SHIFT LEADER          YES     S   S   S   S   S
-  XXAOC06      SENIOR OPERATOR       NO      -   -   -   -   -
-  -----

```

Figure 63. Operator Notification Panel

Here 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.

Selecting a defined notification operator brings you to the panel shown in Figure 64 on page 115 where you can select the message classes defined by SA OS/390 (see Table 5 on page 116) or if you scroll down on this panel, you reach the input line

Assign User message classes ==> Yes Yes or No

where you can specify **Yes** to have the operator notified about user-defined message classes. In such a case, SA OS/390 displays existing message classes that have been defined with entry type *Auto Msg Classes* from which you can select the classes you want.

How to define messages for a message class and where to store them, is described in *System Automation for OS/390 Customizing and Programming*.

```

COMMANDS  HELP
-----
AOFPIN01      Operator Notification Message Classes
Command ==>>

Entry Type : System           PolicyDB Name  : DATABASE_NAME
Entry Name  : SYSTEM_AOCA     Enterprise Name : YOUR_ENTERPRISE

Operator ID : XXAOC05
Description : SHIFT LEADER

Select S the message classes that are appropriate for this operator.
More:      +

A  Automation agent related message classes

    0 - Errors
    1 - Debug
    40 - ALL agent automation messages
    41 - reserved
    42 - reserved
    43 - Information
    44 - Action
    ...
    ...
    ...
    89 - Satisfactory, all resources
    90 - SA Infrastructure/environment

Assign User message classes ==> YES  Yes or No

F1=HELP      F2=SPLIT      F3=END      F4=RETURN      F5=RFIND      F6=RCHANGE
F7=UP        F8=DOWN        F9=SWAP     F10=LEFT       F11=RIGHT     F12=RETRIEVE

```

Figure 64. Operator Notification Message Classes Panel

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 "Auto Msg Classes Policy Object" on page 269 for information about defining your own message classes.) Each SA OS/390 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 5 on page 116 shows the message classes defined by SA OS/390. 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 OS/390 Messages and Codes*.

System Policy Object

Table 5. SA OS/390 Message Classes

Message Class Description	Message Class Number
<i>General Message Classes</i>	
Errors	0
Debug	1
<i>Automation Agent Related Message Classes</i>	
All automation agent messages	40
Information	43
Action	44
Status	45
User intervention	46
Critical threshold setting exceeded	50
Frequent threshold setting exceeded	51
Infrequent threshold exceeded	52
Assist mode alert	55
IMS Automation Messages	60
<i>Automation Manager Related Message Classes</i>	
All automation manager messages	80
Intervention required	81
Automation denied	82
Automation Impacted	83
Awaiting Automation	84
In Automation	85
Degraded - Target resources	86
Degraded - All resources	87
Satisfactory - Target resources	88
Satisfactory - All resources	89
SA OS/390 Infrastructure/Environment	90

For example, any operator defined with a notification operator class of all SA OS/390 messages (class number 40 and 80) receives all SA OS/390 messages. Operators responsible for system operations should receive all SA OS/390 messages or a subset of those messages. As another example, you may want to have your lead operator receive all debug messages.

SA OS/390 uses the general and console automation message classes only. The SA OS/390 customization dialog supports the message classes used by SA OS/390 only.

SA OS/390 uses notification messages to update status display facility (SDF) status displays.

More About Policy Item PROCESSOR OPERATIONS

This section explains the series of panels that are displayed by SA OS/390 if you select policy item PROCESSOR OPERATIONS from the *Policy Selection* panel for *Systems*.

The first panel displayed is the one shown in Figure 65. Use it to modify one or all of the components of the selected target system.

The panel contains the following fields:

```

ISQDPT11                      Processor Operations Target Definition
Command ==>>>

Target hardware: P9021ESA      Target: SAMPLE_SYSTEM_01

Select one target system definition component or ALL.

 1 ALL          Define all components
 2 BASICS      Basic characteristics
 3 IPL          IPL information
 4 CONNECT     Connections
 5 TIME        Time zones
 6 OPLIST      Interested operator lists
 7 CP          CP states (Display only)
 8 CHPID      Channel path ID states (Display only)
The target is a non-( system. You only can display (not change) CP and
CHPID states of the associated target hardware

```

Figure 65. Processor Operations Target Definition Panel for Screen-Oriented Processors

Target hardware

Gives the name of the target hardware (processor) definition associated with the target system. This field cannot be modified from this panel.

Target Gives the name of the target system.

Use one of the following options:

- 1 Display the panels associated with menu options 2 through 8.
- 2 Display the *Target System Basic Characteristics* panel, beginning with Figure 66 on page 118.
- 3 Display the *Target System IPL Information* panels, beginning with Figure 71 on page 123.
- 4 Display the *Target System Connections* panel, Figure 74 on page 126.
- 5 Display the *Processor Operations Target System Time Zone* panel, Figure 75 on page 127.
- 6 Display the *Processor Operations Interested Operator List* panel, Figure 76 on page 128.
- 7 Display the *Processor Operations CP States* panels, beginning with Figure 77 on page 129.
- 8 Display the *CHPID Assignments* panel, Figure 136 on page 217.

The follow-on panels depend on the type of target hardware that you have specified in the PROCESSOR policy item for the system. Some examples are presented in the following sections.

System Policy Object

Target System Basic Characteristics for a Screen-Oriented Processor

For target systems with 9021, 9121, 3090, and 308x processors, use the panel shown in Figure 66. You reach this panel selecting option 2 from the *Processor Operations Target Definition* panel.

The panel contains the following fields:

```
ISQDPT5A                               Target System Basic Characteristics
Command ==>>>

Target hardware: TEST2                   Target: MVSTARG

Enter information for the target system.

Initialized target system   => NO_      YES, NO
IOCDS name                  => IOCDS2__ IO configuration data set
```

Figure 66. Target System Basic Characteristics Panel for 9021, 9121, 3090, and 308x Processors

Target hardware

Indicates the name of the target hardware definition associated with the target system.

Target Indicates the name of the target system.

Initialized target system

Enter YES or Y to indicate that the target system is automatically initialized when SA OS/390 processor operations starts. Otherwise, type NO or N.

Note: If you choose to initialize the target system automatically, be aware that when you start SA OS/390, the ISQSTART command may complete before the target system is initialized.

IOCDS name

Enter the name of the I/O configuration data set (IOCDS) that must be active for the target system. The name you choose must not contain blanks. The customization dialog does not verify your entry.

Note:

The IOCDS name 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 the processor operations function is stopped or restarted.

Target System Basic Characteristics for OCF-Based Processors in LPAR Mode

For target systems with OCF-based processors in (mode, use the panel shown in Figure 67 on page 119 to describe the target system for processor operations.


```

ISQDPT5B                                Target System Basic Characteristics
Command ==>>

Target hardware: CPC2                    Target: LPAR

Enter information for the target system.

Initialized target system    => NO_          YES, NO
Reset Profile Name          => TWOLPAR_    Activate Reset Profile
Image Profile Name          => P2_____    Activate Image Profile
Load Profile Name           => _____    Activate Load Profile
Secondary OCF Name         => P2_____    Secondary OCF Name
LPAR name                   => P2_____
Target mode                 => ESA          370, ESA or CF

```

Figure 67. Target System Basic Characteristics Panel for OCF-Based Processors in (Mode

The panel contains the following fields:

Target hardware

Indicates the name of the target hardware definition associated with the target system.

Target Gives the name of the target system.

Initialized target system

Enter YES or Y to indicate that the target system is automatically initialized when SA OS/390 processor operations starts. Otherwise, type NO or N.

Note: If you choose to initialize the target system automatically, be aware that when you start SA OS/390, the ISQSTART command may complete before the target system is initialized.

Reset Profile Name

Enter the Reset profile name defined at the support element. The name you choose must agree with the Reset 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.

Image Profile Name

Enter the Image profile name defined at the support element. The name you choose must agree with the Image profile defined at the support element. Leave this field blank to take the default defined at the support element.

Note: This profile name should contain only the characters A to Z and 0 to 9. The special characters, \$, #, and @, although allowed, are not recommended.

Load Profile Name

Enter the Load profile name defined at the support element. The name you choose must agree 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.

System Policy Object

Secondary OCF Name

Enter the Secondary operations command facility (OCF) name defined at the support element. This name must be the same as the (name and the Image profile name.

Note: This profile name should contain only the characters A to Z and 0 to 9. The special characters, \$, #, and @, although allowed, are not recommended.

LPAR Name

Enter the name of the logical partition for this target system as defined in the specified IOCDs. Names can contain from 1 to 8 alphanumeric characters, starting with an alphabetic character, #, \$, or @.

Target Mode

370, ESA, or CF

Target System Basic Characteristics for OCF-Based Processors in Basic Mode

For target systems running on a zSeries or 390-CMOS processor operating in basic mode, use the panel shown in Figure 68 to describe the target system for processor operations.

```
ISQDPT5C                               Target System Basic Characteristics
Command ==>>>

Target hardware: CPC1      Target: BASICM

Enter information for the target system.

Initialized target system  => NO_          YES, NO
Reset Profile Name        => 390RESET     Activate Reset Profile
Load Profile Name         => LOADMVS_     Activate Load Profile
```

Figure 68. Target System Basic Characteristics Panel for an OCF-Based Processor in Basic Mode

The panel contains the following fields:

Target hardware

Indicates the name of the target hardware definition associated with the target system.

Target Indicates the name of the target system.

Initialized target system

Enter YES or Y to indicate that the target system is automatically initialized when SA OS/390 processor operations starts. Otherwise, type NO or N.

Note: If you choose to initialize the target system automatically, be aware that when you start SA OS/390, the ISQSTART command may complete before the target system is initialized.

Reset Profile Name

Enter the Reset profile name defined at the support element. The name you choose must agree with the Reset 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.

Load Profile Name

Enter the Load profile name defined at the support element. The name you choose must agree 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 System Basic Characteristics for 9674

For target systems on 9674 processors, use the panel shown in Figure 69 to describe the target system for processor operations.

ISQDPT5D Target System Basic Characteristics
 Command ==>>

Target hardware: CPC2 Target: CFLPAR

Enter information for the target system.

Initialized target system	=> NO_	YES, NO
Reset Profile Name	=> CFLPAR__	Activate Reset Profile
Image Profile Name	=> P3_____	Activate Image Profile
Load Profile Name	=> _____	Activate Load Profile
Secondary OCF Name	=> P3_____	Secondary OCF Name
LPAR name	=> P3_____	
Target mode	=> CF	

Figure 69. Target System Basic Characteristics Panel for 9674

The panel contains the following fields:

Target hardware

Indicates the name of the target hardware definition associated with the target system.

Target Gives the name of the target system.

Initialized target system

Enter YES or Y to indicate that the target system is automatically initialized when SA OS/390 processor operations starts. Otherwise, type NO or N.

Reset Profile Name

Enter the Reset profile name defined at the support element. The name you choose must agree with the Reset 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.

Image Profile Name

Enter the Image profile name defined at the support element. The name you choose must agree with the Image profile defined at the support element. Leave this field blank to take the default defined at the support element.

Note: This profile name should contain only the characters A to Z and 0 to 9. The special characters, \$, #, and @, although allowed, are not recommended.

System Policy Object

Load Profile Name

Enter the Load profile name defined at the support element. The name you choose must agree 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.

Secondary OCF Name

Enter the Secondary operations command facility (OCF) name defined at the support element. This name must agree with the (name and the Image profile name.

Note: This profile name should contain only the characters A to Z and 0 to 9. The special characters, \$, #, and @, although allowed, are not recommended.

LPAR Name

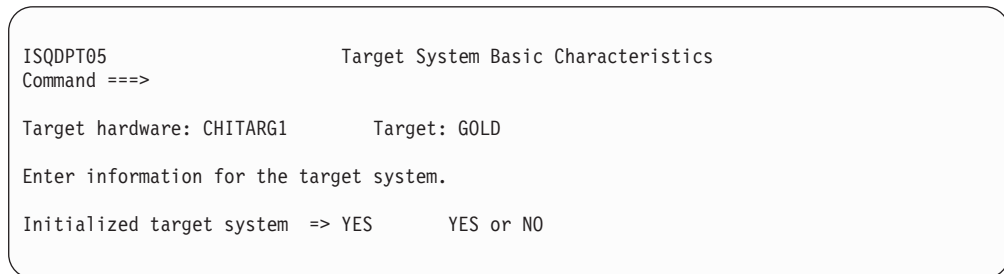
Enter the name of the logical partition for this target system as defined in the specified IOCDs. Names can contain from 1 to 8 alphameric characters, starting with an alphabetic character.

Target Mode

Indicates a target mode of CF. This field cannot be modified.

Target System Basic Characteristics for 4381 Processor

For target systems with 4381 processors, use the panel shown in Figure 70.



```
ISQDPT05                                Target System Basic Characteristics
Command ==>>>

Target hardware: CHITARG1                Target: GOLD

Enter information for the target system.

Initialized target system => YES          YES or NO
```

Figure 70. Target System Basic Characteristics Panel for 4381

The panel contains the following fields:

Initialized target system

Enter YES or Y to indicate that the target system is automatically initialized when SA OS/390 processor operations starts. Otherwise, type NO or N.

Note: If you choose to initialize the target system automatically, be aware that when you start SA OS/390, the ISQSTART command may complete before the target system is initialized.

Defining Screen-Oriented Target System IPL Information

Use the following panels to define target system IPL information.

ISQDPT06	Target System IPL Information	
Command ==>		
Target hardware: CHITARG1	Target: GOLD	information for the target system.
IPL address	=> 01A4	Hex
IPL parameter	=> 111	
Target CP for IPL	=> 2	0-9
Automatic IPL	=> Y	YES, NO, or Blank

Figure 71. Target System IPL Information Panel for Screen-Oriented Processors

This panel displays the target system and target hardware associated with the target system. The panel is not required and is bypassed for coupling facility target systems.

The panel contains the following fields:

Target hardware

Indicates the name of the target hardware definition associated with the target system.

Target Indicates the name of the target system.

IPL address

Enter the 4-digit direct access storage device (DASD) unit address from which the system is loaded, in hexadecimal.

IPL parameter

Enter the 1- to 8-digit load parameters used in the processor LOAD command, in hexadecimal.

Note: Similar information for OCF-based processors is specified in the appropriate profile definitions.

Target CP for IPL

Enter the number (0 to 9) of the CP to be used in the initial program load (IPL) process to activate the target system. The CP value is model-dependent.

Automatic IPL (logically partitioned systems only)

If the target system IPL procedure should be performed automatically during logical partition activation, enter YES or Y. Otherwise, enter NO or N. For target systems that do not run on logical partitions, leave this field blank.

Note: The IPL address, IPL parameters, Target CP, and Automatic IPL fields can be changed without generating and loading a new SA OS/390 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 OS/390 is stopped or restarted.

MVS and OS/390 Target System IPL Information

To define responses used during the MVS or OS/390 target IPL process, use the panel shown in Figure 72 on page 124. During the start process, the target system requests information from the operator. By providing the responses at this panel, the operator enables SA OS/390 to respond to these messages. Target systems

System Policy Object

using TPF, VSE, LINUX, or CF do not request information from the operator during startup.

The panel contains the following fields:

```
ISQDPT6A                      MVS Target System IPL Information
Command ==>>>

Target hardware: P9021ESA      Target: SYSTEM1

Enter responses to the following messages for the target system.

IEA101A specify system parameters
=> CLPA,SYSP=AS

IEA347A specify master catalog parameter
=> _____

IEA213A or IEA214A DUPLICATE VOLUME
=> AAA  999 1110 EAB  A9A 9A9  BBB0 CCC  DDD  EEE0 FFF  AFF  _
_____

Provide default response when neither device is specified. => YES
```

Figure 72. MVS Target System IPL Information Panel

Target hardware

Gives the name of the target hardware definition associated with the target system.

Target Gives the name of the target system.

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; SA OS/390 automation provides them. 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 an OS/390 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 are to 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 being required devices. For IEA214A messages, the current IPL (SYSRES) devices 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.

System Policy Object

- 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 OS/390 processor control file by the use of the host ISQXOPT command before IPLing the target system. However, these changes are not saved if SA OS/390 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 OS/390 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 73. During the start process, the target system requests information from the operator. By providing the responses at this panel, the operator enables SA OS/390 to respond to these messages.

```
ISQDPT6B          VM Target System IPL Information
Command ==>>>

Target hardware: CHITARG1 Target: G      er message for the  target system.

VM IPL START message response
=> _____
```

Figure 73. VM Target System IPL Information Panel

Target hardware

Gives the name of the target hardware definition associated with the target system.

Target Gives the name of the target system.

VM IPL START message response

Enter the desired response to the VM IPL START message. The customization dialog does not verify your entry.

System Policy Object

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 OS/390 is stopped or restarted.

Target System Connections

To assign a particular PC and connection card as the active and backup operator console (OC) for the current target system, use the panel shown in Figure 74. If you have an OCF-based processor, use of this panel is optional.

The panel contains the following fields:

```
ISQDPT08                      Target System Connections
Command ==>>>                                SCROLL ==>>> PAGE

Target hardware: CHITARG1  Target: GOLD

Assign operator consoles using the Action column.
Actions: A =   For active operator console  B =   For backup operator console

Action  PS/2 name  Card type  Card number  Port letter
A       CALVIN   3270      2            B
B       WOODROW  3270      2            B
-       STEVE   3270      1            A
-       TEDDY   3270      1            A
-       COMPLEX5 3270      1            A
-       CHICAGO2 3270      1            A
-       CHICAGO3 3270      1            A
-       CHICAGO4 3270      1            A
```

Figure 74. Target System Connections Panel

Target hardware

Gives the name of the target hardware definition associated with the target system.

Target Gives the name of the target system.

Action

Presents the action you can select. Enter one of the following:

- A** Designates this card as the PC port for the active operator console. Required for systems that use a PS/2 connection for their operator console. Select only one.
- B** Designates this card as the port for the backup operator console. Optional. Select only one.

The remaining fields display the defined operator console connections of the target hardware definition associated with this target system.

PC Name

Gives the name of the PC.

Card type

Gives the card type.

Card number

Gives the connection card number.

Port letter

Gives the port letter of the connection card.

Target System Time Zones

To describe the target time zone in relation to the focal point system, use the panel shown in Figure 75.

The panel contains the following fields:

```

ISQDPT09          Processor Operations Target System Time Zones
Command ==>>

Target hardware: CHITARG1 Target:

Enter information as required.

Transmitted time signal in use   => NO    YES or NO

If transmitted time signal is not in use, fill in one field.

Target time same as host   =>    YES or blank
Target time earlier       => 1    1 to 23, or blank
Target time later         =>    1 to 23, or blank

```

Figure 75. Processor Operations Target System Time Zone Panel

Target hardware

Gives the name of the target hardware definition associated with the target system.

Target Gives the name of the target system.

Transmitted time signal in use

If you have a WWV radio receiver installed for this target system, type YES or Y in this field and leave the remaining fields on this panel blank. If not, type NO or N and complete the next field.

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.

If the WWV radio receiver provides the time signal, leave this field blank.

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.)

If the WWV radio receiver provides the time signal, leave this field blank.

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.)

If the WWV radio receiver provides the time signal, leave this field blank.

Note to OS/390 Users: SA OS/390 supports local time only in reply to a time-of-day (TOD) request.

Interested Operator Lists

To designate up to ten operators to receive messages from the target system active consoles, use the panel shown in Figure 76.

System Policy Object

```
ISQDPT12          Processor Operations Interested Operator Lists
Command ==>>>

Target hardware: CHITARG1 Target: GOLD

Enter operators who are to receive messages from console.

System Console
Operator 1 => STEVENP_      Operator 2 => CHARLIE_
Operator 3 => JOHN_____  Operator 4 => SOPHIA__
Operator 5 => _____    Operator 6 => _____
Operator 7 => _____    Operator 8 => _____
Operator 9 => _____    Operator 10 => _____

Operator Console
Operator 1 => ALAN_____  Operator 2 => ALICE___
Operator 3 => LLOYD_____ Operator 4 => ZOOT____
Operator 5 => _____    Operator 6 => _____
Operator 7 => _____    Operator 8 => _____
Operator 9 => _____    Operator 10 => _____
```

Figure 76. Processor Operations Interested Operator List Panel

Note: For OCF-based systems using console integration, the system messages on the system path go to the system list.

This panel contains the following fields:

Target hardware

Gives the name of the target hardware definition associated with the target system.

Target Gives the name of the target system.

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 go to the system console interested operator list.
2. For screen-oriented systems, operators must be on both the system console list and the operator console list to receive system console and operator console messages.
3. Operators not on these lists do not see messages from these consoles unless their user IDs are added using the ISQXMON command when SA OS/390 is running.

CP States for 9021, 9121, and 3090 Processors

For target systems on target hardware with logically partitioned processors, use the panel shown in Figure 77 on page 129 to change the states of the central processors and the vector processors and to allocate the storage size of the current target system.

For non-LPAR target systems, CP states can be displayed but not modified.

```

ISQDPT07          Processor Operations CP States
Command ==>>>

Target hardware: CHITARG1 Target: G

Enter information for the target system.

Central storage size  => 36_      Megabytes
Expanded storage size => 112_     Megabytes

CP      Online      Vector Processor
      (ON or OFF)  (YES or NO)
0 => ON_          YES
1 => ON_          YES
2 => ON_          NO_
3 => ON_          NO_
4 => ON_          YES
5 => ON_          NO_
6 => ON_          NO_
7 => ON_          NO_
8 => ON_          NO_
9 => ON_          NO_
  
```

Figure 77. Processor Operations CP States Panel for 9021, 9121, and 3090 Processors

The panel contains the following fields:

Target hardware

Gives the name of the target hardware definition associated with the target system.

Target Gives the name of the target system.

Central storage size

Specify the amount of primary storage (in megabytes) assigned to this logical partition. This field can contain up to four digits.

Expanded storage size

Specify the amount of expanded storage (in megabytes) assigned to this logical partition. This field can contain up to four digits.

CP Gives the target hardware central processor number.

Online

Enter ON next to the CPs required to activate the system. Enter OFF next to the CPs not required.

Vector processor

Enter YES or Y next to the number of the vector processors required to activate the system. Enter NO or N if the vector processor is not installed.

Note: The “Central Storage size,” “Expanded Storage size,” can be changed without generating and loading a new processor operations control file. You can do this by issuing the host ISQXOPT command after you IPL the target system. However, these changes are not saved if processor operations is stopped or restarted.

The following panel is displayed when the END key is pressed on panel ISQDPT07.

System Policy Object

Use the panel shown in Figure 78 to define logical partitions for the current target system defined for the target hardware definition, which is in (mode.

```
ISQDPT7A                                CP LPAR States
Command ==>>>

Target hardware: CHITARG1 Target: G0

Enter information for the target system.

Favored LPAR          => YES          YES or NO
LPAR name             => YEAREND
Initial LPAR state    => ACTIVE      ACTIVE or INACTIVE
LPAR processing weight => 100
LPAR capping          => YES          YES or NO
Target mode           => ESA
```

Figure 78. CP LPAR States Panel

The panel contains the following fields:

Target hardware

Gives the name of the target hardware definition associated with the target system.

Target Gives the name of the target system.

Favored LPAR

If this target system is a favored logical partition, type YES or Y. Otherwise type NO or N.

If there is competition for available system hardware resources, a favored logical partition has priority. More than one logical partition can be favored.

LPAR name

Enter the name of the logical partition for this target system as defined in the specified IOCDS. Names can contain from 1 to 8 alphanumeric characters, starting with an alphabetic character, #, \$, or @.

Initial LPAR state

If this logical partition should automatically be activated at the startup of the target hardware, type ACTIVE. If not, type INACTIVE.

LPAR processing weight

Enter the LPAR processing weight.

Processing weight determines the amount of processor resources provided to this particular logical partition. Valid values are 1 through 999, or D for a dedicated LPAR. The default is a blank.

LPAR capping

To use LPAR capping, type YES or Y. Otherwise type NO or N.

By typing YES in this field, you restrict this target system to using only the share of processor resources determined by the LPAR processor weight value. By typing NO, you enable this target system to use more than its share of resources, when those resources are available.

Target mode

Enter the target mode (ESA, 370, or CF).

Note: The Favored LPAR, LPAR processing weight, and LPAR capping fields 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 OS/390 is stopped or restarted.

CP States for a 308x Processor

For target systems with 308x processors, use the panel shown in Figure 79 to display the states of the central processors.

```
ISQDPS7B                      Processor Operations CP States
Command ==>>>

Target Hardware: CHITARG1

Enter information for the target hardware.

Central storage size   => 24__      Megabytes

CP      Online
        (ON or OFF)
0      => OFF
1      => OFF
2      => ON_
3      => ON_
```

Figure 79. Processor Operations CP States Panel for a 308x Processor

The panel contains the following fields:

Target hardware

Indicates the name of the target hardware definition.

Central storage size

Indicates the amount of primary storage (in megabytes) associated with the target hardware.

CP Indicates the target hardware CP number.

Online

Enter ON next to the CPs that are available on the target hardware. Enter OFF next to the CPs that are not available.

CP States for a 4381 Processor

For target systems with 4381 processors, use the panel shown in Figure 80 on page 132 to display the states of the central processors and the storage size for a target hardware definition.

System Policy Object

```
ISQDPS7C                               Processor Operations CP States
Command ==>>

Target Hardware: CHITARG1

Enter information for the target hardware.

Central storage size => 102_           Megabytes
Expanded storage size => 134_           Megabytes

CP      Online
        (ON or OFF)
0  =>  ON_
1  =>  OFF
```

Figure 80. Processor Operations CP States Panel for a 4381 Processor

The panel contains the following fields:

Target hardware

Indicates the name of the target hardware definition.

Central storage size

Indicates the amount of primary storage (in megabytes) associated with the target hardware.

Expanded storage size

Indicates the amount of expanded storage (in megabytes) associated with the target hardware.

CP Indicates the target hardware CP number.

Online

Enter ON next to the CPs that are available on the target hardware. Enter OFF next to the CPs that are not available.

CHPID Assignments

To specify mandatory and optional channel path identifiers (CHPIDs) for a logically partitioned target system, see Figure 136 on page 217. For non-LPAR target systems CHPIDs can only be displayed on Figure 136 on page 217.

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.

ApplicationGroup Policy Object

In contrast to System Application Groups which may be associated with any number of groups, Sysplex Application Groups may be associated with only one sysplex group. See “Policy Items for Sysplex Groups” on page 94 for more information on sysplex groups.

You define an application group as either SYSTEM or SYSPLEX when you create it using the *Define New Entry* panel. A Sysplex Application Group can be associated with only one sysplex group.

The SA OS/390 customization dialog allow concurrent user access capability for entry type *ApplicationGroup*. See also “Appendix A. Concurrent Customization by Multiple Users” on page 349 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 25.

Figure 81 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.

```
COMMANDS  HELP
-----
A0FGAN00          Define New Entry
Command ==>>>

To define a new entry, specify the following information:
Type . . . . . ApplicationGroup
Name . . . . . CICS_APG

Application Group Type . SYSPLEX      SYSTEM  SYSPLEX
Nature . . . . . BASIC              BASIC MOVE SERVER
Default Preference . . . *DEF        0 to 1000, *DEF
-----

Automation Name . . . . .           Name
Automatically link Application-Resources
into APG . . . . . YES              YES      NO

Behaviour. . . . . ACTIVE           ACTIVE PASSIVE

Short Description . . .
Extended Description . .
. .
. .
. .
. .
-----
```

Figure 81. Define New Entry Panel for a Sysplex Application Group

The unique fields are:

Name is the name of the application group. It must be unique within the policy database. Valid names may contain up to twenty alphanumeric characters. The name cannot be changed after the application group is created.

Application Group Type

The type of application group you are creating; either SYSPLEX or SYSTEM. This value cannot be changed after the application group is

ApplicationGroup Policy Object

created. SYSPLEX is the default. This attribute determines the boundaries from where you can select the members for the Application Group:

SYSPLEX

This type lets you select applications and resources that may reside on multiple systems within the specified sysplex group (see below).

SYSTEM

This type lets you select applications and resources that reside on single systems.

Nature

In this field you can define the availability behaviour 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 MOVE group, because only one member should be available. The second MOVE group now starts the shared member *RES1*. 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 51. 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

on page 139 and to page 72. Or, if you want to read how an operator can override preference values, refer to *System Automation for OS/390 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 OS/390 generates the following resources:

```
SYST_APG/APG/SYSTEM1  
SYST_APG/APG/SYSTEM2
```

Automatically link Application-Resources into APG

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).

Behaviour

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.

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 ApplicationGroups

As described in "Creating a New ApplicationGroup" on page 133, there are two types of application groups:

- System Application Groups
- Sysplex Application Groups

Figure 82 on page 136 shows the policy items that are displayed when a System Application Group is selected from the *Entry Name Selection* panel.

ApplicationGroup Policy Object

ACTIONS		HELP	
AOFGEPOL		Policy Selection	
Command ==>		SCROLL==> HALF	
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	Display ApplicationGroup information	
	APPLICATIONS	Select Applications for ApplicationGroup	
	TRIGGER	Select Trigger	
	SERVICE PERIOD	Select Service Period	
	RELATIONSHIPS	Define Relationships	
-----RESOURCES-----			
	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 Systems linked to this entry	
	COPY	Copy data from existing entry	

Figure 82. 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 upcoming 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 137 for more information.

APPLICATIONS

When you select this policy item, a list of all applications in the SA OS/390 enterprise is displayed, indicating which, if any, are associated with this 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 138 for more information on this policy item.

TRIGGER

Each application group may be linked to one trigger. With this policy item, in the upcoming panel *Trigger for ApplicationGroup*, you can 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, in the upcoming panel *Service Period for ApplicationGroup*, you can link one service period to the application group so that its resources are automatically started up or shut down in the time frame specified with the service period.

RESOURCES

If you have defined the current application group to become a resource (by specifying an *Automation Name* during generation), then, if you select this policy item, SA OS/390 proceeds as follows depending on the type of the current application group:

ApplicationGroup Policy Object

- for type SYSTEM, SA OS/390 displays panel *Select APG-Resource for Grouping* showing a list of all application group resources that have been generated for the current application group. Here you can select an APG-resource, for which you want to modify resource members. You will see panel *Select Resources* with the resources that are defined for the selected APG-resource. From this panel, you can add or remove resources into your application group.
- for type SYSPLEX, SA OS/390 directly displays panel *Select Resources* with a list of selectable resources defined for the specified sysplex. From this panel, you can add or remove resources into your application group.

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 51 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 140 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 for ApplicationGroups” on page 141 for more details.

RELATIONSHIPS

This policy item has the same functionality for entry types *Application* and *ApplicationGroup*. For each application or application group you may specify several relationships. They will be used to automate the startup or stopping of those applications and application groups. If you select this policy item, in the upcoming *Relationship Selection List* panel, you can view existing relationships or define new relationships for the current application or application group.

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 157 for more information on seeing existing or defining new dependency relationships.

More About Policy Item APPLGROUP INFO

A panel like Figure 83 on page 138 will be displayed if you select policy item APPLGROUP INFO from the Policy Selection Panel for Application Groups.

ApplicationGroup Policy Object

```

COMMANDS  HELP
-----
AOFGAPA0          Application Group Information
Command ==>>>

Entry Type : ApplicationGroup      PolicyDB Name : DATABASE_NAME
Entry Name : CICS_APG             Enterprise Name : YOUR_ENTERPRISE

The following field was specified when the Application Group was
defined and cannot be modified:
  Application Group Type : SYSTEM
-----

Update the following information:

Nature . . . . . BASIC          BASIC MOVE SERVER
Default Preference . . . *DEF      0 to 1000, *DEF
Automation Name . . . . CICS_APG  Name

Behaviour. . . . . ACTIVE        ACTIVE PASSIVE

Automatically link Application-Resources
into APG . . . . . YES          YES      NO
-----

```

Figure 83. Application Group Information Panel

The fields contained in this panel are explained in “Creating a New ApplicationGroup” on page 133. You can update the values shown in the lower part of this panel.

More About Policy Item APPLICATIONS

If you select an application group’s policy item *APPLICATIONS*, you reach the *Applications for ApplicationGroup* panel. Figure 84 shows an example.

```

COMMANDS  ACTIONS  VIEW  HELP
-----
AOFGX4E          Applications for ApplicationGroup
Command ==>>>                                     SCROLL==>> PAGE

Entry Type : ApplicationGroup      PolicyDB Name : SCENARIO
Entry Name : NET                   Enterprise Name : SCENARIO

Action      Status      Application
          SELECTED
          CICSAOR1
          CICSAOR2
          CICSAOR3
          CICSFOR
          CICSMAIN
          CICSTOR
          DB2IRLM
          DB2MAIN
          DWH
          SELECTED TCPIP
          SELECTED VTAM
          WBSRV
-----

```

Figure 84. Applications for ApplicationGroup Panel for System Application Groups

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 OS/390 will generate a resource for this application. If you set the option *Automatically link Application-Resources* to YES when creating the application group, SA OS/390 will also add the new resource into this application group. Otherwise, you just have an "independent" application resource that you can start or stop or automate as required.

More About Policy Item RESOURCES

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 column *Preference* in the following screen).

```

COMMANDS  ACTIONS  HELP
-----
A0FGARS2          Select Resources
Command ==>>>                                SCROLL==>> PAGE

Entry Type : ApplicationGroup      PolicyDB Name : DATABASE_NAME
Entry Name : DWHAPG                Enterprise Name : YOUR_ENTERPRISE

Number of selected Resources . . . : 0
Behaviour. . . . . : ACTIVE

Update the following information:
  Availability Target. . . . . 1          1 (for Move-APGs)

  Show valid APG-Resources . . . . YES    YES NO
  Show selected Resources only . . NO    YES NO
  Show only Resources with String.

Action   Preference  Resource Name
-----
         _____ CICS/APG
         _____ CICSAOR/APG
         _____ DB2/APG/SYSTEM2
         _____ DB2/APG/SYSTEM3
         _____ 700_ DWHAPG2/APG
         _____ 500_ DWHAPG3/APG
         _____ NET/APG
  
```

See the following list for available preference values. Preferences other than those listed are also accepted if you wish to define your own preferences. You can find more information on the meaning of preferences in *System Automation for OS/390 User's Guide*.

- 1000** The resource is always started, other resources from the same group with preference 700 are stopped in favour of this member. Use this value if this APL- or APG-Resource must run.
- 900** The resource is selected for startup after an IPL and after recovery situations. Members with preference 500 will be stopped in favour of this group.
- 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 (1000) alternative becomes available. This value is the default. Use it in a MOVE group for a resource that normally runs.
- 600** This value is used as a threshold. A resource with preference 600 and lower will not be started at group start up.
- 500** This resource is not selected for startup after IPL but it is started as a backup after failure and continues running until 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.

ApplicationGroup Policy Object

- 300 Like 200, but a move to an alternate backup (Preference 500) would only occur at a (scheduled) outage.
- 200 This resource will be started as backup and stopped as soon as a better alternative (preference 450 and greater) becomes available. Use this value where you can have a backup, but you do not really want one.
- 1 A resource with this preference will never be selected by SA OS/390 automation. If it is available, it will be used, but automation will never ask for it to be started if it is unavailable. Operators may make such resources available interactively by using the INGGROUP command and give it a higher preference. Use this value to provide completely manually controlled application availability.
- 0 This value indicates that the member is passive and that no requests should be sent to it. If a member is made passive, then any requests that have been previously sent will be withdrawn.

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 OS/390 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.

```
COMMANDS  HELP
-----
AOFGXRSR          Generated Resources
Command ==>>>          SCROLL==>> PAGE

Entry Type : ApplicationGroup      PolicyDB Name  : DATABASE_NAME
Entry Name  : NET                  Enterprise Name : YOUR_ENTERPRISE

Resource Name      Generated by link to
- Resources for THIS entry:
NET/APG/SYSTEM1   SYSTEM1
NET/APG/SYSTEM2   SYSTEM2
- Resources for APPLICATIONS (generated by links to this entry):
TCP/IP/APL/SYSTEM1 NET
TCP/IP/APL/SYSTEM2 NET
VTAM/APL/SYSTEM1  NET
VTAM/APL/SYSTEM2  NET
```

Figure 85. 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 for ApplicationGroups

The *Member of* panel appears if you select policy item *MEMBER OF* from the *Policy Selection* panel for application groups.

COMMANDS		HELP	

AOFGXMOF	Member of		
Command ==>			SCROLL==> PAGE
Entry Type : ApplicationGroup	PolicyDB Name :	SCENARIO	
Entry Name : NET	Enterprise Name :	SCENARIO	
Own Resource...	...is a member of	Preference	Nature
NET/APG/SYSTEM1	WBSRVAPG/APG	700	MOVE/SERVER
NET/APG/SYSTEM2	WBSRVAPG/APG	700	MOVE/SERVER

Figure 86. Member of Panel for ApplicationGroups

This panel lists all resources, of which the current application group is a member. This panel is for information only. No actions are allowed.

If the owning ("is a member of") APG resource is of nature SERVER or MOVE, then column *Preference* shows the preference value that was set for the current resource in the RESOURCES policy item of the owning resource. If the owning resource's nature is BASIC, it shows "SELECTED". For more details about available preference values see "Creating a New ApplicationGroup" on page 133.

In our example from Figure 86, the resource *NET/APG/SYSTEM1* in turn is a member of resource *WBSRVAPG/APG*.

Application Policy Object

An *application* is an OS/390 subsystem, started task, application, batch job, or non-MVS resource that runs on a system in the SA OS/390 enterprise. SA OS/390 monitors applications through the messages they issue to the operator. Application is also used to refer to a subsystem or process on an SNMP-capable machine. Note that SA OS/390 does not provide automation for applications on non-MVS systems.

An application cannot be linked to a system directly. It must be defined as part of an application group, and it may belong to more than one application group. If a system is assigned multiple application groups containing the same application, SA OS/390 ensures that the application is defined only once in the automation control file.

The SA OS/390 customization dialog allow concurrent user access capability for entry type *Application*. See also "Appendix A. Concurrent Customization by Multiple Users" on page 349 for further information.

Creating a New Application

How to create new policy objects in general, is described in "How to Create New Policy Objects" on page 25.

Application Policy Object

Figure 87 shows the panel that is displayed when you create a new application from the *Entry Name Selection* panel for applications.

```
COMMANDS  HELP
-----
AOFGLN00          Define New Entry
Command ==>>>

To define a new entry, specify the following information:
Type. . . . . Application
Application Name. . . . APPL1_____
Subsystem Name. . . . . APPL1_____
-----

Object Type . . . . . INSTANCE  CLASS  INSTANCE
Application Type . . . . STANDARD  STANDARD JES2 JES3 CICS IMS DB2 OPC USS
                                   (Only the value STANDARD can be changed)
                                   (once, all others cannot be changed)
                                   (after the application has been created)
Subtype . . . . . _____ (Only for type CICS, IMS, OPC or DB2)
Clone Job Name. . . . . NO      YES      NO
Job Name. . . . . _____
Scheduling Subsystem. . . . _____ MSTR, JES Subsystem or blank
JCL Procedure Name. . . . _____ (Proc used with JOBNAME=)
MVS Automatic Restart
Management Element
Name. . . . . _____
                                   (Only if the application uses )
                                   (MVS Automatic Restart Management)
WLM Resource Name . . . . _____
Short Description . . . . _____
Extended Description. . . . _____
. . . . . _____
. . . . . _____
. . . . . _____
. . . . . _____
-----
```

Figure 87. Define New Entry Panel for Applications

When you define a new application, you provide the following information:

Application 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 OS/390 automation. You may have multiple applications with the same subsystem name, but they cannot be linked to the same 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 alphabetic or numeric character or any of @, #, \$, ? or _ (underscore), with the first character being alphabetic, numeric 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.

Object Type

CLASS

If the application represents a common policy to be inherited by one or more other application entries

INSTANCE

If the application is to be managed by SA OS/390. You can omit policy items from an application instance and allow them to be inherited from an application 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 OS/390 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 OS/390 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 OS/390 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. Other type values are JES2, JES3, CICS, DB2, IMS, OPC or USS. The value STANDARD can be changed once, all other values cannot be changed after the application is created. If you specify one of these, additional fields are displayed in the *Policy Selection* panel for the application, allowing you to enter policy specific to the subsystems. These application specific policy items are described in the following subsections:

- “Policy Items for JES2 Subsystems” on page 181
- “Policy Items for JES3 Subsystems” on page 184

Application Policy Object

- “Policy Items for CICS Subsystems” on page 185
- “Policy Items for DB2 Subsystems” on page 190
- “Policy Items for IMS Subsystems” on page 191
- “Policy Items for OPC Subsystems” on page 199

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.

Clone Job Name

This field is optional. Use it to indicate whether the job name can be cloned.

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 OS/390 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.

You can substitute a defined clone value anywhere in the job name using the &AOCCLONE n . notation, where n may be omitted or is a number from 1 to 9. You may also use the tilde character (~) to get the &AOCCLONE. clone value.

You can also specify standard OS/390 and NetView cloning variables within the job name.

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.

Scheduling Subsystem

The Scheduling Subsystem is optional. If the application is to be scheduled by a subsystem other than the primary JES subsystem, the name of the scheduling subsystem must be provided. The subsystem so identified must be defined as being of type JES2 or JES3. A key value of MSTR is also permitted to identify those subsystems scheduled directly by MVS/ESA master scheduler.

JCL Procedure Name

The JCL Procedure Name is optional. This specifies the name of the JCL procedure under which the application job is to run.

You can substitute a defined clone value anywhere in the procedure name using the &AOCCLONE n . notation. Additionally, you can specify standard OS/390 and NetView cloning variables.

MVS Automatic Restart Manager 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.

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.

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 for which you want Automatic Restart Manager recovery. Automatic restart management requires that USERPROC have a different element name on each system. Defining USERPROC to SA OS/390 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 OS/390 application with a different element name on each system.

You can substitute a defined clone value anywhere in the element name using the &AOCCLONEn. notation, where *n* is a number from 1 to 9. You can also specify standard OS/390 and NetView cloning variables.

WLM Resource Name

This field is optional. You may specify the name of the WLM resource that is associated with the application.

You can substitute a defined clone value anywhere in the WLM resource name using the &AOCCLONEn. notation, where *n* is a number from 1 to 9. You can also specify standard OS/390 and NetView cloning variables. If cloning is used with job names, you can keep the WLM resource naming convention consistent across all systems.

Whenever the status of an application which has a WLM resource name specified changes, system operations issues an MVS MODIFY command to update the status of the WLM resource. If the automation status of the application is UP or ENDED, system operations sets the status of the WLM

Application Policy Object

resource to ON. In any other case it is set to OFF. This is independent of the setting of the application's automation flag.

How to Use Clone Variables

SA OS/390 provides support for automating applications that can run on more than one system in a sysplex. You do not need to define automation policy separately for every system on which the application runs. With SA OS/390 you can specify up to ten clone IDs to identify a system. These clone IDs are then used to qualify the application job name to ensure a unique job name on each system.

Clone IDs can be used for job names, WLM resource names and also to replicate applications that are controlled by Automatic Restart Manager. You can use one clone ID for the job name and the WLM resource name and then another clone ID for the Automatic Restart Manager element name, or various other combinations as required by your installation.

A unique system value can also be substituted anywhere within a literal, such as a shutdown command. The value is also available to your automation procedures through the AOFAOCCLONEn NetView common global variables. Refer to *System Automation for OS/390 Defining Automation Policy* for more information on restrictions to use of clone IDs.

The clone IDs are resolved when the automation control file is loaded during SA OS/390 initialization and before SA OS/390 sends any commands. Once the automation control file is loaded, the value cannot be changed without reloading the automation control file.

If you are using a single procedure for the application and using the JOBNAME= parameter on the MVS start command, you need to tell SA OS/390 about this. If you do not specify the PROCNAME on the application definition, SA OS/390 automatically builds an appropriate start command. The normal start up command is:

```
MVS S jobname
```

But if a PROCNAME is specified, SA OS/390 issues:

```
MVS S procname,JOBNAME=jobname
```

Note that the job name is the result of the symbolic substitution described above and is the name used for all automation of the application.

To replicate application automation, you need to do the following things during SA OS/390 customization:

- When you define each system to SA OS/390, specify one or more unique &AOCCLONE. values for that system.
- If your job name varies depending on the system, specify that it should be replicated and link it to each system on which it runs. By default, SA OS/390 appends the first &AOCCLONE. value at the end of the application job name to derive the job name to be used on that system.
- Alternatively, you can use one or more of the &AOCCLONE. values in any position in the job name.
- When specifying shutdown commands for replicated applications, you may find it useful to use the &SUBSJOB substitution variable rather than reconstructing the cloned job name in the command.

SA OS/390 also supports standard OS/390 and NetView cloning variables within the command definitions. These are for example, &DOMAIN., &SYSNAME., &SYSPLEX. and &SYSCclone..

Note: Those variables cannot be used for cloning the jobname.

Policy Items for Applications

Figure 88 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, the start timeout can be set to a default value of one minute for all applications. If the definition of an application has no value specified for the start timeout, the value specified as the *Application Defaults* is used. If no value is specified in the *Application Defaults*, the value specified in *System Defaults* is used. If no value is specified for *System Defaults*, a hard-coded default is used. See also “Application Defaults Policy Object” on page 262 and “System Defaults Policy Object” on page 261.

ACTIONS		HELP
AOFGEPOL		Policy Selection
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	Enter and display 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
	SYSTEM ASSOCIATION	Define primary and secondary associations
-----RESOURCES-----		
	GENERATED RESOURCES	List resources generated for this entry
	MEMBER OF	List resources where this entry is a member
-----USS SPECIFIC POLICY-----		
	USS CONTROL	Define USS Control specifications

	WHERE USED	List ApplicationGroups linked to this entry
	COPY	Copy data from an existing entry

Figure 88. Applications Defined as Instance Objects and of Type STANDARD

The policy items for applications are:

LINK TO CLASS

See Figure 88. This field is displayed if the application was defined as an instance (refer to “Creating a New Application” on page 141). 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 field is displayed if the application was defined as a class (refer to “Creating a New Application” on page 141). 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, on which you may see 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.

AUTOMATION INFO

This policy item is used to define the application to SA OS/390 automation. Refer to “More About Policy Item AUTOMATION INFO” on page 150 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 “Automation Flags” on page 42.

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 behaviour 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 “Event and Trigger Support” on page 47.

SERVICE PERIOD

Service periods allow users to schedule the availability of applications or application groups. 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.

RELATIONSHIPS

Relationships are used to define dependencies between resources, which are used to perform start up or shutdown actions in a controlled way of dependent and supporting applications and application groups.

This policy item has the same functionality for entry types *Application* and *ApplicationGroup*. For each application or application group you may

Application Policy Object

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 group.

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 currently selected, will become the dependent resource, if you are defining a new relationship.

Refer to “More About Policy Item RELATIONSHIPS” on page 157 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 163.

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 169 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 OS/390. 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 173.

THRESHOLDS

Error threshold definitions for an application define how manyabend 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 OS/390 Uses Error Thresholds” on page 39.

MINOR RESOURCE FLAGS

This policy item allows you to define flags for minor resources for the 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

Application Policy Object

See the description of the policy item MINOR RESOURCE FLAGS in “MVS Component Policy Object” on page 252 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 47 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 OS/390 determine whether or not an application should be started by SA OS/390 when SA OS/390 is initialized on a system. These categories allow SA OS/390 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 177.

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 85 on page 140, 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 for ApplicationGroups” on page 141 for more details.

More About Policy Item AUTOMATION INFO

Panel *Application Automation Definition* is displayed, if you select the policy item *AUTOMATON INFO* from panel *Policy Selection* for the *Application* entry type.


```

COMMANDS  HELP
-----
AOFPISS1          Application Automation Definition
Command ==>>>

Entry Type : Application          PolicyDB Name : DATABASE_NAME
Entry Name : APPL1              Enterprise Name : YOUR_ENTERPRISE

Subsystem : APPL1
Description:
Job Name   : APPL1

More:

Job Type . . . . . _____ Job properties (MVS NONMVS TRANSIENT)
Transient Rerun . _____ Transient Jobtype can be rerun (YES NO)
Command Prefix . . _____ Console command character(s)
Message Prefix . . _____
Enter one or more prefixes (above)

Sysname . . . . . _____ JES Subsystem (name)

Start on IPL . . . _____ Start with Netview init (YES NO NONE blank)
Start on Recycle . _____ Start with Sys-Ops recycle (YES NO NONE blank)
Start Timeout . . _____ Time allowed to reach "UP" status (hh:mm:ss)

Monitor Routine. . _____ Routine used for monitoring (name NONE)
Periodic Interval. _____ Periodic monitoring interval (hh:mm NONE)

Restart Option . . _____ Restart Circumstances (ALWAYS ABENDONLY NEVER)
External Startup . _____ External Startup (INITIAL ALWAYS NEVER blank)
External Shutdown. _____ External Shutdown (FINAL ALWAYS NEVER blank)

Shut Delay . . . . . _____ Time between attempts to shutdown (hh:mm:ss)
Term Delay . . . . . _____ Time for termination cleanup (hh:mm:ss)

```

Figure 89. Application Automation Definition Panel

This panel contains the following fields:

Job Name

The job name used for the application.

JobType

This may be one of:

MVS This is the standard MVS job type that SA OS/390 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 OS/390 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 SHUTDOWN passes specified and a user START specified.

TRANSIENT

This is a type of job that may be started by SA OS/390 but terminates without the intervention of SA OS/390. It is a short

Application Policy Object

running job that will generally, but not necessarily, terminate before SA OS/390 does. It can be monitored using an ASCB check.

A TRANSIENT job, as far as monitoring is concerned, is treated as an OS/390 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 OS/390. These passes are optional.

Transient Rerun

This field specifies whether or not a transient job may be rerun within the lifetime of an SA OS/390. 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 OS/390.

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 OS/390 is recycled. If this is not desired, then the SA OS/390 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 OS/390 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 OS/390 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.

Command Prefix

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 an OS/390 console should be prefixed with MVS. NetView commands to control NetView automation do not require the MVS prefix.

You can substitute a defined clone value anywhere in the command prefix using the &AOCCLONEn. notation. Additionally, you can specify standard OS/390 and NetView cloning variables.

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 jobname. These message prefixes must be unique for this application.

Application Policy Object

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.

Start on IPL

The Start on IPL option specifies how SA OS/390 determines the initial status of the application on the first occasion that SA OS/390 is started after an OS/390 IPL or an automation status file reallocation.

Reallocating the automation status file has the same effect as an OS/390 IPL because SA OS/390 stores the last IPL date and time in the automation status file. When the automation status file is deleted, SA OS/390 assumes that the next startup is the first startup following an OS/390 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 OS/390 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 OS/390 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 OS/390 determines the initial status of the application when SA OS/390 is re-initialized on a system that has not had an MVS IPL since SA OS/390 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 OS/390 to restart it.

Application Policy Object

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 CTLDDOWN, 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 OS/390 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.

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 111). 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 OS/390 refer to "More About Policy Item AUTOMATION SETUP" on page 111. You can, however, specify a user-written monitor routine, or enter NONE if you do not want this application to be monitored.

If the user-written monitor routine is not found when SA OS/390 is running, then the routine AOFAJMON is used.

Periodic Interval

This is the amount of time between application subsystem monitoring cycles. Monitoring may be suspended by specifying NONE.

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 OS/390.

Abandonly

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 OS/390.

Application Policy Object

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

application will only initially be started externally.

ALWAYS

application will always be started externally.

NEVER

application will never be started externally.

blank indicates that the value can be inherited from an application CLASS entry.

External Shutdown

specifies whether the application is stopped externally or via a specified shutdown procedure.

FINAL

application will be stopped externally if the external supporting resource (parent) is also being stopped.

ALWAYS

application will always be stopped externally.

NEVER

application will never be stopped externally.

blank indicates that the value can be inherited from an application CLASS entry

Shut 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.

Term Delay

The delay between application monitoring checks performed by SA OS/390 after the final termination message (TERMMSG FINAL=YES) for an application is processed. The termination procedure is as follows:

1. When SA OS/390 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 OS/390 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 Term 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 Term Delay is 12 seconds.

More About Policy Item RELATIONSHIPS

The *Relationship Selection List* panel Figure 90 is displayed if you select policy item RELATIONSHIPS from the *Policy Selection* panel for application groups (see “Policy Items for ApplicationGroups” on page 135) or applications (which is the case in our example). 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 (Figure 91 on page 158).

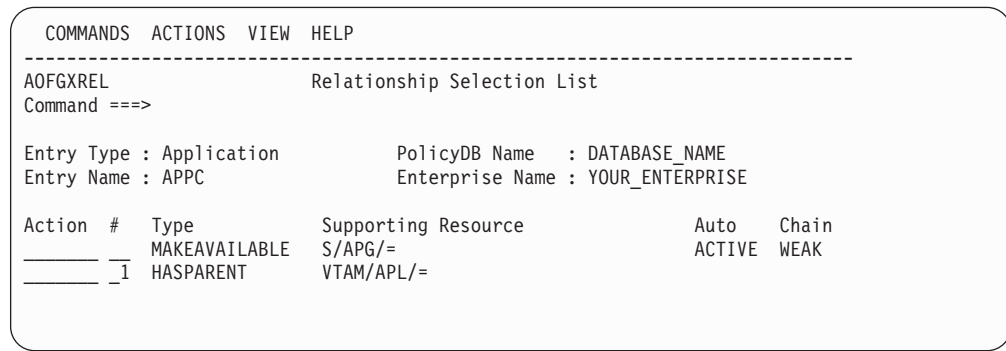


Figure 90. Relationship Selection List Panel

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 “Define Relationships”.

Entering the *NEW* command from this panel to define new relationships, will bring you the *Define Relationship* panel described in “Define Relationships”.

Define 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 91 on page 158) you will specify the name of the supporting resource (in field *Supporting Resource*).

You can define new relationships for the current resource by issuing the *NEW* command from the *Relationship Selection List* panel (Figure 90). SA OS/390 will display the *Define Relationship* panel (Figure 91 on page 158).

Application Policy Object

Note:

If invoked for applications, the *Define Relationship* panel will display the application's subsystem name in the field *Subsystem*

If invoked for application groups, the group's automation name appears in the field *Group ID* instead.

```
COMMANDS  HELP
-----
A0FGXREQ          Define Relationship
Command ==>>>

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

Subsystem :      APPC
Description. . . . . _____

Relationship Type. . _____  MAKEAVAILABLE MAKEUNAVAILABLE
                                PREPAVAILABLE  PREPUNAVAILABLE
                                HASPARENT  EXTERNALLY

Supporting Resource. _____
                                Resource Name
Sequence Number. . . . . ____  Sequence Number (1-99,blank)

Automation . . . . . _____  ACTIVE  PASSIVE
Chaining . . . . . _____  STRONG  WEAK
Condition . . . . . _____
                                Satisfy condition
                                (? for list of possible values)
```

Figure 91. Define Relationship Panel

This panel contains the following fields :

Group ID

This field is displayed for entry type *ApplicationGroup* only and contains the automation name of the application group for which you are currently defining the relationship. You cannot update this entry.

Subsystem

This field is displayed for entry type *Application* only and contains the application's subsystem name. You cannot update this entry.

Description

Here you can type a fourty-character text to describe the relationship.

Relationship Type

Here you define the relationship between the application group or application, 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 it. Dependencies are only checked if they are of importance for the existing desired goal.

Select one of the following values:

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 entry field *Supporting Resource* (see below). 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 entry field *Condition* below). 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.

Note: This relationship type implements the parent/child relationship from earlier SA OS/390 releases. Those are converted into HASPARENT.

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 OS/390 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.

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.

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:

Application Policy Object

APGAutomationName/APG/MVSSysname
APGAutomationName/APG/=
APGAutomationName/APG
SubSystemName/APL/MVSSysname
SubSystemName/APL/=

The SubSystemName may be cloned with one AOCCLONE variable.

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 PARENT=(.) field. The position within the list is determined by this number. Valid values are 1 - 99 and blank.

You only need to specify sequence number 1 if you plan to use any of the SUBP... variables of the common routine AOCQRY. For performance reasons, it is not recommended that you specify a sequence number if you do not use SUBP... variables.

Note: The number is forced to blank for relationships where the supporting resource is an APG resource.

Automation

defines the processing of SA OS/390 in order to satisfy a relationship:

ACTIVE

specifies that SA OS/390 will attempt to bring the supporting resource to the desired condition to satisfy the dependency. This is the default for MAKEAVAILABLE and MAKEUNAVAILABLE.

PASSIVE

specifies that SA OS/390 will wait until the supporting resource is in the desired condition to satisfy the dependency. That is, the request to bring the supporting resource into the desired condition needs to come from outside, such as an operator command.

This field is forced to blank for HASPARENT and EXTERNALLY relationships. This field is forced to PASSIVE for PREPAVAILABLE and PREPUNAVAILABLE relationships.

Chaining

defines whether the status of the supporting resource, or the status of the complete internal dependency graph is to be considered:

STRONG

specifies that SA OS/390 evaluates the status of the underlying dependency graph up to the next WEAK dependency, when more dependencies exist on higher levels.

WEAK

specifies that SA OS/390 evaluates just the status of the next (direct) dependency (this is the default for MAKEAVAILABLE and MAKEUNAVAILABLE).

Strong chaining causes SA OS/390 to check all relationships of the involved resources along the dependency graph. Weak chaining simply requires that the action is permitted on the dependent resource.

This field is forced to blank for HASPARENT and EXTERNALLY relationships.

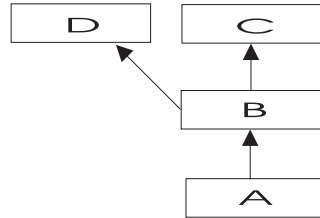


Figure 92. Example for Strong Chaining

In the example, as shown in Figure 92, for strong chaining, SA OS/390 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*, *B* and *D* 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. 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

Refer to "Relationship Conditions" for more information on available condition values and the context in which they may be used.

Relationship Conditions: The relationship conditions that are available, depend on the relationship.

The relationship conditions for HASPARENT 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 162. *Observed status* means the observed status for the supporting resource. *Desired status* means the desired status for the supporting resource. So read this table like in the following example: 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 also is *Available*. For the condition *WhenRunning*, the action specified in *Relationship Type*

Application Policy Object

may be taken when the supporting resource's observed status is *Available* or *Degraded* and its desired status is *Available*.

Table 6. Relationship Conditions

Condition	Observed Status of supporting resource	Desired Status of supporting resource	Passive Only
WhenAvailable	Available, WasAvailable	Available	
WhenRunning	Available, WasAvailable, Degraded	Available	
WhenSoftDown	SoftDown	Unavailable	
WhenHardDown	HardDown	Unavailable	Yes
WhenDown	SoftDown, HardDown, SysGone	Unavailable	
WhenAssumedDown	SoftDown, HardDown, SysGone, Unknown	Unavailable	
WhenAvailableOrStarting	Starting, Available, WasAvailable	Available	
WhenRunningOrStarting	Starting, Available, WasAvailable, Degraded	Available	
WhenDownOrStopping	Stopping, SoftDown, HardDown, SysGone	Unavailable	
WhenAssumedDownOrStopping	Stopping, SoftDown, HardDown, SysGone, Unknown	Unavailable	
WhenStartableOrAvailable	Available, WasAvailable OR SoftDown, Unknown and start dependency satisfied	Available	
WhenStartableOrRunning	Available, WasAvailable, Degraded OR SoftDown, Unknown and start dependency satisfied	Available	
WhenStoppableOrSoftDown	SoftDown OR Available, WasAvailable, Degraded, Starting, Stopping, problem and stop dependency satisfied	Unavailable	
WhenStoppableOrDown	SoftDown, HardDown, SysGone OR Available, WasAvailable, Degraded, Starting, Stopping, problem and stop dependency satisfied	Unavailable	
WhenStoppableOrAssumedDown	SoftDown, HardDown, SysGone, Unknown OR Available, WasAvailable, Degraded, Starting, Stopping, problem and stop dependency satisfied	Unavailable	
WhenObservedDown	HardDown, SoftDown, SysGone	n/a	

Table 6. Relationship Conditions (continued)

Condition	Observed Status of supporting resource	Desired Status of supporting resource	Passive Only
WhenObservedAssumedDown	HardDown, SoftDown, SysGone, Unknown	n/a	
WhenObservedSoftDown	SoftDown	n/a	
WhenObservedHardDown	HardDown	n/a	Yes
WhenObservedAvailable	Available	n/a	
WhenObservedRunning	Available, Degraded	n/a	
WhenGroupHasNotFailed	Group/Failed flag not set	n/a	Yes

Any of these conditions can be used with any of the action relationships (PrepareAvailable, MakeAvailable, PrepareUnavailable, MakeUnavailable).

More About Policy Item MESSAGES/USER DATA

If you select policy item MESSAGES/USER DATA from the *Policy Selection* panel for Applications (Figure 88 on page 147), you come to the *Message Processing* panel shown in Figure 93.

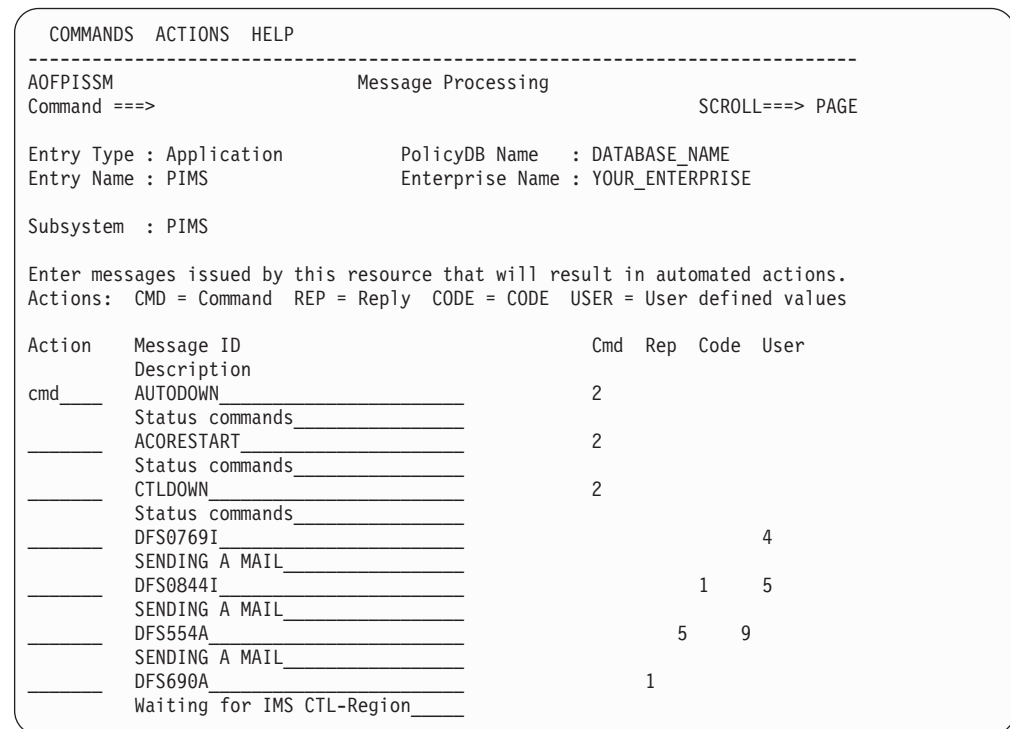


Figure 93. Message Processing Panel

On the *Message Processing* panel, you specify how SA OS/390 should react if the application issues the specified message. Therefore you can type one of the following types of message response into the *Action* column. Depending on what

Application Policy Object

you type here, SA OS/390 displays a follow on panel where you further specify SA OS/390s automated reaction to the message.

For each of the automated responses, you need to add an entry to the NetView message automation table specifying which routine SA OS/390 should use to process your specified automated response. There are special SA OS/390 generic routines or common routines that you can use for this purpose, or you can use a user-specified automation routine. Information on how to add an entry to the NetView message automation table is presented in *System Automation for OS/390 Customizing and Programming*.

- CMD** for issuing commands as response to a message, see “CMD Processing” on page 165. In the NetView message automation table use the ISSUECMD routine to process this kind of message processing.
- REP** for issuing replies, see “Reply Processing” on page 166. In the NetView message automation table use the ISSUEREP routine to process this kind of message processing.
- 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 167. In the NetView message automation table use the CDEMATCH routine to process this kind of message processing.
- USER** You can use the upcoming panel to let SA OS/390 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 *System Automation for OS/390 Programmer’s Reference*.

Reserved Message IDs

With the following message IDs and also with any of the SA OS/390 defined keywords you can invoke a special SA OS/390 automation:

- ACORESTART (which is associated with action *CMD*)
- WTORS
- MVSDUMPTAKEN
- MVSDUMPFULL
- SMFDUMP
- LOGREC
- SYSLOG
- or any of the system operations status

You can also specify a short description of the action the response performs.

Note: For product automation (for example CICS, IMS) there are further keywords defined. For details see the documentation of the automated products.

For product automation, the following reserved message IDs exist:

- for CICS:
 - ABCODETRAN (.xxxxx, where xxxxx is the optional transaction name)
Additional transaction keyword dropped
 - HEALTHCHK

- LISTSHUT
- RCVRAUTO
- RCVRSOS
- RCVRTRAN (.xxxxx, where xxxxx is the optional transaction name)
Additional transaction keyword dropped
- RCVRVIOL
- for IMS:
 - ABCODETRAN (.xxxxx, where xxxxx is the optional transaction name)
Additional transaction keyword dropped
 - ABCODEPROG (.xxxxx, where xxxxx is the program name) Additional
program keyword dropped

For more information on how to use these various methods of message processing, and for information on how to add the related entries into the NetView message automation table, refer to *System Automation for OS/390 Customizing and Programming*.

CMD Processing

SA OS/390 displays the *CMD Processing* panel if you entered the action code *CMD* into the *Action* column from panel *Message Processing*.

```

COMMANDS  HELP
-----
AOFPISSC          CMD Processing
Command ==>>>          SCROLL==>> PAGE

Entry Type : Application      PolicyDB Name  : DATABASE_NAME
Entry Name  : JES3           Enterprise Name : YOUR_ENTERPRISE

Subsystem   : JES3
Message ID  : JESABEND

Enter commands to be executed when resource issues the selected message.

Pass/Selection Automated Function/'*'
Command Text

FREQ _____
MSG SYSOP '==> JES3ABEND FREQ ALERT <==' _____

CRIT _____
MSG SYSOP '==> JES3ABEND CRIT ALERT <==' _____
  
```

Figure 94. CMD Processing Panel

Use this panel to enter commands to be executed when the application issues the previously specified message. If you specify one of the SA OS/390 provided automation commands (described in *System Automation for OS/390 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.

You can use the following variables within the command specification. They will be replaced with their current value before the command is issued:

- &SUBPJOB
- &SUBSAPPL

Application Policy Object

- &SUBSCMDPFX
- &SUBPCMDPFX
- &SUBSDESC
- &SUBSJOB
- &SUBSSCHEDSS
- &SUBSSHUTDLY
- &SUBSSPARM
- &SUBSPROC

The value entered into 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 to 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/**'

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 OS/390 automated functions are mapped to NetView automation operators. This is documented in Table 11 on page 264.

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.

A maximum of 99 commands can be entered.

Reply Processing

SA OS/390 displays the *Reply Processing* panel if you entered the action code *REP* into the *Action* column from panel *Message Processing*.


```

COMMANDS  HELP
-----
AOFPISSR          Reply Processing
Command ==>>>          SCROLL==>> PAGE

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

Subsystem   : HSM
Message ID  : ARC0055A

Enter the replies to be issued when this resource issues the selected message.

Pass/      Retry Reply Text
Selection  Count
-----
          --  GO _____
          --  _____
          --  _____
          --  _____

```

Figure 95. Reply Processing Panel

Use this panel to enter the replies to be issued when this resource issues the previously specified message.

The use of the *Pass* or *Selection* field is the same as for the *CMD Processing* panel described in the previous section. 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 OS/390 console (without the R and the reply number). To specify a null reply, code NULL in this field.

Code Processing

SA OS/390 displays the following panel if you entered the action code *CODE* in the *Action* column from panel *Message Processing*.

Application Policy Object

```

COMMANDS  HELP
-----
AOFPISSSE                                Code Processing
Command ==>>>                                SCROLL==>> PAGE

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

Subsystem : NETVIEW
Message ID : WTORS

Enter the value to be passed to the calling CLIST when this resource
issues the selected message and the following codes are contained in
the message.

      Code 1          Code 2          Code 3          Value Returned
DSI802A _____ * _____ _____ NORMAL _____
DSI803A _____ * _____ _____ NORMAL _____
* _____ * _____ _____ IMPORTANT SEC _____
_____
_____
_____
_____

```

Figure 96. Code Processing Panel

Use this panel to enter the value to be passed to the calling CLIST when the 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.

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 of the 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.

Example:

```

      Code 1          Code 2          Code 3          Value Returned
ABEND* _____ SA22 _____ _____ ABENDING _____

```

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 our example, ABEND* and SA22 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.

User Defined Data

SA OS/390 displays the *User Defined Data* panel if you entered the action code *USER* into the *Action* column from panel *Message Processing*.

```

COMMANDS  HELP
-----
AOFPISSU                                User Defined Data
Command ==>>                                SCROLL==>> PAGE

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

Subsystem : HSM
Message ID : ARC0909E

To change keyword-data pair, specify the following:

Keyword
Data
LIMIT _____
97% _____
_____
_____
_____
_____

```

Figure 97. User Defined Data Panel

You can use this panel to let SA OS/390 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 *System Automation for OS/390 Programmer's Reference*.

More About Policy Item STARTUP

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 98, 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 OS/390 issue a sequence of prestart commands to *A* while the supporting resource *B* is starting.

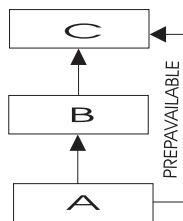


Figure 98. Using STARTUP Policy

Application Policy Object

If you select policy item STARTUP from the *Policy Selection* panel for Applications (Figure 88 on page 147), you come to the *Subsystem Startup Processing* panel shown in Figure 99.

```
COMMANDS  ACTIONS  HELP
-----
A0FGASTT          Subsystem Startup Processing
Command ==>>>

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

Subsystem   : APPC
Description : APPC/MVS automation policy

To specify automated commands when starting this subsystem,
select the appropriate startup phase.

Action      Phase      Description                                Cmd
-----
PRESTART    Executed before startup is initiated
STARTUP     Executed to initiate the startup
POSTSTART   Executed after startup has been initiated
```

Figure 99. 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 OS/390 perform certain actions for the preparation of an application. For example, before starting CICS, you can have the VTAM major node activated.

You can specify commands for the following startup phases:

PRESTART

Commands entered under this type are issued before the application is started. You can use these to perform some preparations, for example, let SA OS/390 issue VTAM commands to activate additional connections.

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 under this type are issued immediately after the START command has been issued.

The CMD field displays the number of commands that have been entered for a particular startup entry.

Application Policy Object

For all of these options, you come to panel *Startup Command Processing* (Figure 100) where you can enter the startup commands that are specific to the current application.

```
COMMANDS  HELP
-----
A0FGASTC          Startup Command Processing
Command ==>>>                                SCROLL==>> PAGE

Entry Type : Application          PolicyDB Name : DATABASE_NAME
Entry Name : IMS2                Enterprise Name : YOUR_ENTERPRISE

Subsystem          : IMS2
Startup Phase     : PRESTART

Scheduling Subsystem . . _____ MSTR, JES Subsystem or blank
JCL Procedure Name . . . _____ (Proc used with JOBNAME=)
Parameters . . . _____
                               Enter Subsystem Startup Parameters(above)

Type              Automated Function/'*'
Command text
WARM _____ NETOPER _____
VARY NET,ACT,ID=APL&SUBSJOB _____

-----
WARM _____ JESOPER _____
MVS $SI U0-U5 _____

-----
_____
-----
_____
-----
_____
-----
_____
-----
```

Figure 100. Startup Command Processing Panel

You can fill in the following fields:

Scheduling Subsystem

This specifies the name of the subsystem (application) that is to schedule the current application, as follows:

MSTR indicating that the application is scheduled directly by the OS/390 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 (see *Application Type* in the *Define New Entry* Panel for Applications Figure 87 on page 142).

This field is optional.

JCL Proc 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
```

Application Policy Object

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
```

Note: If you are running an MVS version earlier than 5.1 you should leave this field blank.

Parameters

Specifies the desired application startup parameters. These values are added onto the MVS START command for the application. For example, the desired start command for VTAM may be:

```
S VTAM,,,(LIST=00)
```

Therefore, if this command is to be submitted from SA OS/390, 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.

If at least one start command is available in the scrollable portion, no additional command is generated automatically, based on the entry fields in the panel's top section. However, the values from the fields can be used within the command text from any command via the variables &SUBSSCHEDSS, &SUBSPROC, &SUBSSPARM, giving a high degree of flexibility for the specification of the commands.

The scrollable portion of the panel defines the start commands:

Type Enter the condition (for example, ALWAYS) for which the command should be executed. If no value is entered, SA OS/390 takes ALWAYS as the default. Any value (for example, WARM, COLD, or RESTART) is allowed and can be used as the start type with the INGREQ command. The INGREQ command uses NORM as its default.

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 OS/390 automated functions are mapped to NetView automation operators. This is documented in Table 11 on page 264.

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.

A maximum of 99 commands can be entered.

More About Policy Item SHUTDOWN

If you select policy item SHUTDOWN from the *Policy Selection* panel for Applications (Figure 88 on page 147), you come to the *Subsystem Shutdown Processing* panel shown in Figure 101.

```

COMMANDS  ACTIONS  HELP
-----
AOFPISSH          Subsystem Shutdown Processing
Command ==>>>

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

Subsystem : APPC
Description: APPC/MVS automation policy

To specify automated commands or replies when shutting down this subsystem,
enter the appropriate action for the particular shutdown phase.

Actions:  CMD = Command  REP = Reply

Action      Phase  Description                                Cmd  Rep
-----
cmd_____  INIT   Executed when shutdown initiated
_____    NORM   Executed when normal shutdown invoked      1
_____    IMMED  Executed when immediate shutdown invoked
_____    FORCE   Executed when force shutdown invoked       2
_____    FINAL  Executed after final termination msg

```

Figure 101. Subsystem Shutdown Processing Panel

Similar to the startup processing of applications described in the previous section, you can specify certain options on how to process an application's shutdown here:

INIT Initial shutdown commands are issued when an application shutdown process is initiated. The specified commands are issued on every application shutdown attempt. Initial shutdown commands are issued only once per application and per shutdown request.

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

Normal shutdown commands or replies actually stop an application. These are the commands and replies that are used to perform a normal, orderly application shutdown.

IMMED

Immediate shutdown commands and replies are operator commands that force an application shutdown without delays. For example, normal shutdown commands may issue a warning message and wait for an interval of time before issuing the actual shutdown command. In contrast,

Application Policy Object

commands defined as immediate shutdown commands would not issue any warnings or delay issuing the shutdown command.

FORCE

Like immediate shutdown commands and replies, forced shutdown commands and replies force an application shutdown without delays. You may choose to define different commands and replies for immediate and forced shutdowns, even though these shutdown types are very similar.

FINAL

The commands that you will specify in the follow on panel will be executed after the shut down application has issued the final termination message. In the action field for this type of shutdown processing, you can only specify *CMD* (C).

SA OS/390 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 OS/390 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 OS/390 assumes that the application is down.

If SA OS/390 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, ... FINAL) 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 sections:

- “Command Shutdown Processing for INIT and FINAL” on page 175
- “Command Shutdown Processing for NORM, IMMED and FORCE” on page 175
- “Reply Shutdown Processing” on page 176

Command Shutdown Processing for INIT and FINAL

For INIT and FINAL shutdown commands, the *Shutdown Command Processing* panel is displayed. Figure 102 is an example. Specify the text of the command to be issued in the *Command Text* field. Also specify a value in the *Type* field: Enter the condition for which the command should be executed. NORM, IMMED and FORCE are the permissible values and correspond to the shutdown phases SHUTNORM, SHUTIMMED and SHUTFORCE.

Optionally, specify the automated function under which each command is to run in the *Automated Function* field. If no automated function is specified for a command, it runs under the AOFWRKxx automated function that is responsible for that subsystem. SA OS/390 automated functions are mapped to NetView automation operators. This is documented in Table 11 on page 264.

You cannot issue multiple passes of initial shutdown commands, neither should you invoke MDFYSHUT.

```

COMMANDS  HELP
-----
AOFPI3IC          Shutdown Command Processing
Command ==>>>          SCROLL==>>> PAGE

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

Subsystem:      VTAM
Shutdown Phase: INIT

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 102. Shutdown Command Processing Panel for INIT and FINAL

Command Shutdown Processing for NORM, IMMED and FORCE

For NORM, IMMED, and FORCE shutdown commands, the *Shutdown Command Processing* panel is displayed. Figure 103 on page 176 is an example. Specify the text of the command to be issued in the *Command Text* field. Optionally, specify the automation operator under which each command is to run in the *Automated Function* field. SA OS/390 automated functions are mapped to NetView automation operators. This is documented in Table 11 on page 264. If no automation operator is specified for a command, it runs under the AOFWRKxx automated function that is responsible for that subsystem.

If you use the multiple pass feature for your shutdown commands, specify the pass on which each shutdown command is to be issued. If you do not specify a pass for a command, it will be issued on every pass.

Application Policy Object

For example, Figure 103 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.

If you want to use MDFYSHUT to change the shutdown pass processing, you should specify it as a command on one of the passes.

```
COMMANDS  HELP
-----
AOFPI5HC          Shutdown Command Processing          Row 1 to 2 of 21
Command ==>>>                                     SCROLL==>> PAGE

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

Subsystem   : APPC
Shutdown Phase: NORM

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 103. Shutdown Command Processing Panel for NORM, IMMED, and FORCE

Reply Shutdown Processing

For NORM, IMMED, and FORCE shutdown replies, the *Shutdown Reply Processing* panel is displayed if you enter action code REP. Figure 104 on page 177 is an example. Specify the retry count in the *Reply Count* field. This value specifies the number of times, at 2-second intervals, that SA OS/390 checks for an outstanding reply number in the automation status file. This checking process is used to determine whether SA OS/390 should retry issuing the reply if the first attempt is not successful. The checking process continues until one of the following occurs:

- A reply number is found in the automation status file and the reply is issued.
- The number of retries reaches the value specified in the Retry Count field. In this case, the reply is not issued, and common routine ACFREP issues a nonzero return code. The shutdown process will issue an AOF750I message.

A pass number is required for each shutdown 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.

```

COMMANDS  HELP
-----
AOFPIHR           Shutdown Reply Processing           Row 1 to 5 of 20
Command ==>>>                                SCROLL==>> PAGE

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

Subsystem      : APPC
Shutdown Phase: NORM

Enter the replies to be issued when the selected shutdown phase is invoked
for this subsystem.

Pass Retry Reply Text
  Count
1_  9_  SHUT_____
3_  9_  I TA_____
4_  9_  I TER ACQ_____
5_  9_  END_____
--  --  _____
--  --  _____
--  --  _____
--  --  _____
--  --  _____

```

Figure 104. Shutdown Reply Processing Panel for NORM, IMMED and FORCE

More About Policy Item MEMBER OF for Applications

The *Member of* panel appears if you select policy item *MEMBER OF* from the *Policy Selection* panel for applications.

```

COMMANDS  HELP
-----
AOFGXMOF           Member of
Command ==>>>                                SCROLL==>> PAGE

Entry Type : Application           PolicyDB Name  : SCENARIO
Entry Name  : AOR1                 Enterprise Name : SCENARIO

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 105. Member of Panel for Applications

In our 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.

More About Policy Item SYSTEM ASSOCIATION

This policy item is available for user purposes only. It is not used for automation by SA OS/390.

Application Policy Object

If you select an application group's policy item SYSTEM ASSOCIATION, SA OS/390 displays the *System Association* panel.

COMMANDS ACTIONS VIEW HELP			
AOFGLPD0		System Association	
Command ==>		SCROLL==> PAGE	
Entry Type : Application		PolicyDB Name : DATABASE_NAME	
Entry Name : APPC		Enterprise Name : YOUR_ENTERPRISE	
Action	Association	System	Sysplex
_____	PRIMARY	SYSTEM_AOCA	SYSPLEX_AOCPLEX
_____	PRIMARY	SYSTEM_AOCB	SYSPLEX_AOCPLEX
_____	PRIMARY	SYSTEM_AOCC	SYSPLEX_AOCPLEX
_____	PRIMARY	SYSTEM_AOCD	SYSPLEX_AOCPLEX
_____	PRIMARY	SYSTEM_KEY6	SYSPLEX_KEY67
_____	PRIMARY	SYSTEM_KEY7	SYSPLEX_KEY67
_____	PRIMARY	SYSTEM_KEY1	SYSPLEX_SYSP

Figure 106. System Association Panel

SA OS/390 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 OS/390 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 OS/390 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 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 OS/390 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 OS/390 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. OS/390 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 **CTLDOWN** 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 42 for more information on automation statuses.

Using **MOVED** and **FALLBACK**

SA OS/390 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 OS/390 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**.

Defining JES Subsystems

When creating an application, you can define it either as a JES2 or JES3 subsystem using the entry field *Application Type* (see “Creating a New Application” on page 141). The *Policy Selection* panel will display additional policy items that relate specifically to the specified JES.

For JES2 the additional policy items are:

- JES2 SPOOLFULL
- JES2 SPOOLSHORT
- JES2 DRAIN

For JES3 the additional policy items are:

- JES3 SPOOLFULL
- JES3 SPOOLSHORT
- JES3 MONITOR
- JES3 ABENDS

The remainder of this section contains information required for both JES2 and JES3. “Policy Items for JES2 Subsystems” on page 181 and “Policy Items for JES3 Subsystems” on page 184 contain information specific to these JES subsystems.

There are two types of spool problems that are recovered by SA OS/390, both relating to excessive spool usage:

SPOOLSHORT

This recovery is initiated when a **HASP050** message is received. It will be prefixed by a character indicating the JES2 subsystem from which it came. Your NetView automation table needs to be updated to have an entry with the correct prefix coded for each JES2 whose spool you want SA OS/390

Application Policy Object

to recover. The message indicates that your current spool utilization has exceeded the current TGWARN value on that JES. TGWARN is defined in the SPOOLDEF statement in your JES initialization member and can be changed dynamically. SA OS/390 will base its recovery targets on the value of TGWARN when the recovery is initiated, so dynamically changing its value will not affect an in-progress recovery.

SPOOLFULL

This recovery is initiated when a HASP355 message is received. The message indicates that all the spool volumes associated with the JES are full. Recovery targets in this case are based on a figure of 100% spool utilization.

You should code TGWARN so that a SPOOLSHORT recovery will be initiated before a SPOOLFULL condition is reached. If this is not done, the recovery process may become unpredictable. When resetting after a SPOOLFULL condition, the problem is downgraded to a SPOOLSHORT. SA OS/390 expects the previously running SPOOLSHORT recovery to activate and try to downgrade the problem to an OK. Without the prior SPOOLSHORT recovery, the spool status will remain in SPOOLSHORT after a successful SPOOLFULL recovery.

JES2 Spool Recovery Process

When a recovery is initiated, a recovery target is worked out by finding either the current TGWARN value for SPOOLSHORT or by using 100% for SPOOLFULL and subtracting the appropriate buffer value. An EVERY timer is used to run the pass processor every RETRY interval.

The first thing the pass processor does is to issue a D SPOOL to the JES to obtain the current spool usage. If it is less than the recovery target, it stops the recovery and sets a timer to reset the pass count after the RESET interval. It then issues the \$L SYS to see who is responsible for the recovery. If this pass processor is responsible, it then checks the Recovery automation flag for the subsystem. If either of the above conditions is not true, the recovery is suspended and the pass count is not incremented. A SPOOLSHORT recovery is always suspended while a SPOOLFULL recovery is running for the same subsystem.

If the pass processor is responsible for the recovery, and the Recovery automation flag is enabled, the pass processor will then increment the pass count and call ACFCMD to issue the appropriate commands from the automation control file. By default, SA OS/390 re-executes the recovery sequence from pass 1 if all recovery actions have been taken and the Spool condition is not resolved. You can change this behavior by setting the appropriate advanced automation option from the AOFEXDEF exit.

You can use the AOFSPPOOLSHORTCMD variable (for SPOOLSHORT conditions) and the AOFSPPOOLFULLCMD variable (for SPOOLFULL conditions) to tell SA OS/390 to stop recovery attempts when all commands have been executed and to issue message AOF294I to inform the operator that manual intervention is required in order to resolve the Spool condition.

Displaying Spool Status with SDF

The spool status is posted to SDF under the SPOOL generic, with the name of the subsystem as its specific name. To get these displayed on an SDF panel, you need status fields for xxxx.SPOOL, elements 1 through n, where n is the number of different subsystems that use the spool.

SA OS/390 Spool Usage Predictions

SA OS/390 makes predictions about spool usage which are presented to the operator through the SDF status update. There are three different predictions made:

Long Linear

This looks at the first data point and the latest data point (data points occur if either the pass processor runs a spool recovery initialization or a message is received (HASP050 or HASP355)). It is a simple extrapolation to find out how long it is until a line drawn between the points would cut the 100% line.

Short Linear

This is like the long linear, but it is based on the last two data points.

Short 1st order

This takes the last three data points, works out the rate of change between each pair of consecutive points, then works out the variation between these two rates of change. It then extrapolates using this (quadratic) variation to determine when the curve would intersect the 100% lines.

The accuracy of these predictions will depend on the characteristics of your spool usage and recovery.

The values reported for the predictions are:

hh:mm:ss

This is the time interval until the prediction reached 100%.

NONE

Insufficient data points collected to make the prediction.

STABLE

There is no change in spool usage between the data points. Spool usage is neither increasing nor decreasing.

SAFE The projected spool usage falls below the lowest data point rather than reaching 100%.

DROPPING

The latest data point is lower than the others.

FULL The current data point shows 100% spool utilization.

Spool Recovery Limitations

Spool utilization is nonlinear. Activating the recovery procedure introduces a direct feedback loop. There is a chance that this will lead to your spool exhibiting chaotic behavior. This is most likely if your recovery commands are aggressive and your spool is growing rapidly compared to the interval between the recovery passes. It is probably safest to keep TGWARN high (70%+), use short pass intervals (5-15 minutes), and not to make the recovery commands too aggressive. For example, PS0000-9999 should not be used on SPOOLSHORT pass 1.

Policy Items for JES2 Subsystems

Figure 107 on page 182 is an example of a *Policy Selection* panel for a JES2 subsystem.

Application Policy Object

```

ACTIONS  HELP
-----
AOFGEPOL                                Policy Selection
Command ==>                                SCROLL==> PAGE

Entry Type : Application                  PolicyDB Name : DATABASE_NAME
Entry Name : JES2                         Enterprise Name : YOUR_ENTERPRISE

Action      Policy Name                    Policy Description
:
          -----JES2 SPECIFIC POLICY-----
JES2 SPOOLFULL      Define JES2 spool full procedures
JES2 SPOOLSHORT     Define JES2 spool short procedures
JES2 DRAIN          Define JES2 drain procedures
          -----
:

```

Figure 107. Policy Selection Panel for JES2 Subsystems

See “Policy Items for Applications” on page 147 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 108 on page 183 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 “JES2 Spool Recovery Process” on page 180.

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 OS/390 displays the *Environment Definition: JES2* panel.


```

COMMANDS  HELP
-----
AOFPIEJA          Environment Definition: JES2
Command ==>>>

Entry Type : Application      PolicyDB Name  : DATABASE_NAME
Entry Name  : JES2           Enterprise Name : YOUR_ENTERPRISE

Enter SPOOLFULL settings.
Retry Time . . . .          Spool recovery attempt interval (hh:mm:ss)
Buffer . . . . .           Recovery target below TGWARN (0->50)
Reset Time . . . .         Recovery reset interval (hh:mm:ss)

Priority of systems for spool recovery:

CPUID  1 CPU1  2 CPU2  3      4      5      6      7      8
      9      10     11     12     13     14     15     16
      17     18     19     20     21     22     23     24
      25     26     27     28     29     30     31     32

Edit Spoolfull Pass Commands . . NO

```

Figure 108. Environment Definition: JES2 Panel

The following parameters can be entered:

Retry Time

The time interval, in the format hh:mm:ss, between successive recovery passes.

The default is 5 minutes.

Buffer An integer that specifies the amount of spool that is to be recovered under the threshold that triggered the alert (TGWARN for SPOOLSHORT, 100% for SPOOLFULL). This integer must be positive and less than 75% of the threshold value. If an invalid integer is specified, a value of 5 is used.

The default is 5.

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 an L SYS command is issued and the first machine in the list that is active on the L SYS display 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 OS/390 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 Spoolful Pass Commands

Entering YES in this field allows you to edit the recovery commands.

Application Policy Object

Figure 109 shows the *SPOOLFULL Processing* panel on which you can enter commands to be issued during SPOOLFULL processing.

```
COMMANDS  HELP
-----
AOFPISPF          SPOOLFULL Processing
Command ==>>>          SCROLL==>> PAGE

Entry Type : Application      PolicyDB Name : DATABASE_NAME
Entry Name  : JES2            Enterprise Name : YOUR_ENTERPRISE

Subsystem   : JES2
Description : JES2 policy definition

Enter the appropriate commands to issue when SPOOL is full.

Pass Automated Function/'*'
Command Text
 1 _____
MSG SYSOP '=====> SPOOL ALERT <=====' ' _____
-----
 2 _____
MSG SYSOP ' JES3 SPOOL SPACE IS FULL      ' _____
-----
 3 _____
MSG SYSOP ' PERFORM INSTALLATION RECOVERY. ' _____
-----
 4 _____
MSG SYSOP '=====> SPOOL ALERT <=====' ' _____
-----
```

Figure 109. SPOOLFULL Processing Panel for JES2

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.

Policy Items for JES3 Subsystems

Figure 110 on page 185 shows a *Policy Selection* panel for a JES3 subsystem.

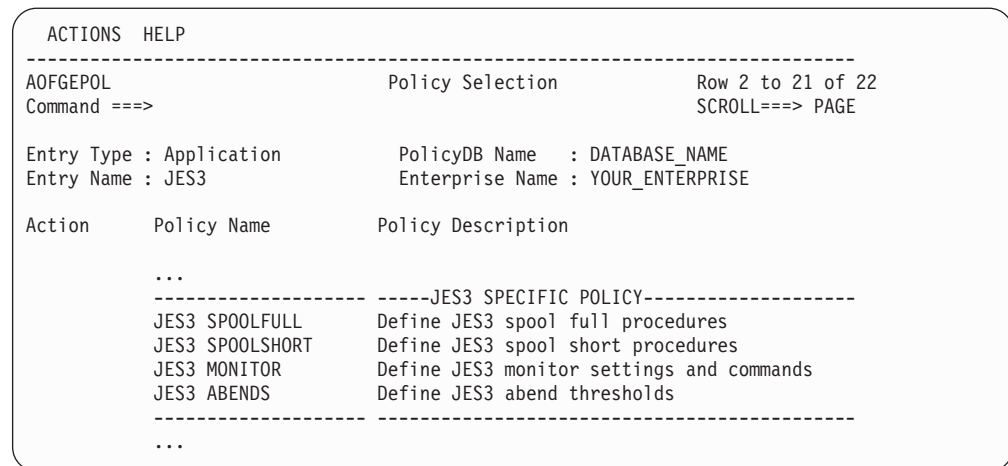


Figure 110. Policy Selection Panel for JES3 Subsystems

See “Policy Items for Applications” on page 147 for descriptions of most policy items on this panel. The additional policy items are:

JES3 SPOOLFULL

Provides for recovery from JES spool full condition. For more information refer to “Defining JES Subsystems” on page 179.

JES3 SPOOLSHORT

Provides for recovery from JES spool short condition. For more information refer to “Defining JES Subsystems” on page 179.

JES3 MONITOR

Allows you to define JES3 monitor settings and commands.

JES3 ABENDS

Set thresholds for JES3 abends. For more information on setting thresholds refer to “How SA OS/390 Uses Error Thresholds” on page 39.

Policy Items for CICS Subsystems

Figure 111 on page 186 shows the *Policy Selection* panel for applications that have been created with application type CICS (see entry field *Application Type* in Figure 87 on page 142). 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.

Application Policy Object

```

ACTIONS  HELP
-----
AOFGEPOL          Policy Selection          Row 14 to 25 of 25
Command ==>>>          SCROLL==>> PAGE

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

Action   Policy Name      Policy Description
...
-----CICS SPECIFIC POLICY-----
CICS CONTROL      Define CICS subsystem extensions
CICS CONNECTION   Select CICS link monitoring definitions
STATE ACTION TABLE Select CICS state action table
RESOURCE THRESHOLDS Define CICS Resource thresholds
...
***** Bottom of data *****

F1=HELP   F2=SPLIT   F3=END    F4=RETURN  F5=RFIND   F6=RCHANGE
F7=UP     F8=DOWN    F9=SWAP   F10=LEFT   F11=RIGHT  F12=RETRIEVE

```

Figure 111. 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 187 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 232. 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 OS/390 CICS Automation Programmer’s Reference and Operator’s Guide*. By selecting a link here, SA OS/390 creates an entry in the automation control file as follows:

```
CONNECTION subsys.connection_id
```

where connection `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 OS/390 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 231. 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 189 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.

```

COMMANDS  HELP
-----
AOFPCXL          CICS control specifications
Command ==>>>

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

Application name: CICS

Enter or update the following fields:

APPLid. . . .
Major node. .
Keypoint req.          Keypoint request (Yes/No)
SMFID . . . .

Auto operator          ( SA13 only )
Start timer            ( SA13 only )
Cold timer             ( SA13 only )
Warm timer             ( SA13 only )
Emerg. timer          ( SA13 only )
F1=HELP      F2=SPLIT  F3=END      F4=RETURN  F5=RFIND  F6=RCHANGE
F7=UP        F8=DOWN   F9=SWAP   F10=LEFT  F11=RIGHT F12=RETRIEVE

```

Figure 112. CICS control specifications

Fill in the following fields:

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, generic-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

Application Policy Object

shutdown. The status will also be changed to stopping if SA OS/390 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 OS/390 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.

AUTO-OPERATOR

(SA OS/390 1.3 only); Identifies the keyword used in the control file entry described in section *AUTOOPS--Automation Operators in System Automation for OS/390 CICS Automation Programmer's Reference and Operator's Guide*. That defines the automation operator used for this CICS subsystem.

START TIMER

(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

(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

(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

(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 minute.

The extended CICS subsystem automation that you define here is also described in section *CICSCNTL* of *System Automation for OS/390 CICS Automation Programmer's Reference and Operator's Guide*.

More About Policy Item RESOURCE THRESHOLDS

With this policy item you can define CICS resource thresholds.

```

COMMANDS  HELP
-----
AOFPIMF3          CICS Resource thresholds selection          Row 1 to 10 of 20
Command ==>>>                                SCROLL==>> PAGE

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

Application name: CICS

Select CICS resource thresholds to be altered.

- _____
- _____
- _____
- _____
- _____

F1=HELP   F2=SPLIT   F3=END     F4=RETURN   F5=RFIND   F6=RCHANGE
F7=UP     F8=DOWN     F9=SWAP    F10=LEFT    F11=RIGHT  F12=RETRIEVE

```

Figure 113. CICS Resource thresholds selection

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 19 on page 41 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.

Application Policy Object

Policy Items for DB2 Subsystems

Figure 114 shows the *Policy Selection* panel for applications that have been created with application type DB2 (see entry field *Application Type* in Figure 87 on page 142). 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.

```
ACTIONS  HELP
-----
AOFGEPOL          Policy Selection          Row 14 to 23 of 23
Command ==>>>          SCROLL==>> PAGE

Entry Type : Application          PolicyDB Name : DATABASE_NAME
Entry Name : DB2                  Enterprise Name : YOUR_ENTERPRISE

Action      Policy Name          Policy Description
...
-----DB2 SPECIFIC POLICY-----
DB2 CONTROL      Define DB2 Control entries
RESOURCE THRESHOLDS  Define DB2 resource Thresholds
...
***** Bottom of data *****

F1=HELP      F2=SPLIT      F3=END      F4=RETURN      F5=RFIND      F6=RCHANGE
F7=UP        F8=DOWN       F9=SWAP     F10=LEFT       F11=RIGHT     F12=RETRIEVE
```

Figure 114. Policy Selection Panel for DB2 Subsystems

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 equal to the *CICS Resource thresholds selection* panel shown in Figure 113 on page 189.

More About Policy Item DB2 CONTROL

With this policy item you can specify the following additional data for a DB2 subsystem:

- DB2 subsystem id
- Active log data set
- Log full threshold
- Shutdown if indoubt
- TSO logoff delay
- STOP dataspace delay
- Connection Mon.Cycle
- Thread termination delay

- Thread termination cycles
- Log full iterations
- Active log alert

Refer to the online help of that panel for further details.

```

COMMANDS  HELP
-----
AOFDPXL          DB2 control entries
Command ==>>>

Entry Type : Application          PolicyDB Name : DATABASE_NAME
Entry Name  : DB2                Enterprise Name : YOUR_ENTERPRISE

Application name: DB2

Enter or update the following fields:

DB2 subsystem id . . DSNP_____
Active log dataset . 'SYS1.PROD.DNLOGP'_____
Log full threshold . 95_    percent
Shutdown if indoubt. ___    Yes/No
TSO logoff delay . . _____ Seconds
STOP dataspace delay _____ Seconds
Connection Mon.Cycle _____ Seconds
Thread term. delay . _____ Seconds
Thread term. cycles. _____
Log full iterations. _____
Active log alert . . _____ Seconds

F1=HELP    F2=SPLIT    F3=END     F4=RETURN  F5=RFIND   F6=RCHANGE
F7=UP      F8=DOWN     F9=SWAP   F10=LEFT  F11=RIGHT  F12=RETRIEVE
    
```

Figure 115. DB2 control entries Panel

Policy Items for IMS Subsystems

Figure 116 on page 192 shows the *Policy Selection* panel for applications that have been created with application type IMS (see entry field *Application Type* in Figure 87 on page 142). 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 ENVIRON and RESOURCE THRESHOLDS will be available.

Application Policy Object

```

ACTIONS  HELP
-----
AOFGEPOL          Policy Selection          Row 14 to 25 of 25
Command ==>>>          SCROLL==>> PAGE

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

Action   Policy Name      Policy Description
...
-----
          IMS CONTROL      Define IMS Control region entries
          IMS ENVIRON      Define IMS Environ tailoring entries
          STATE ACTION TABLE  Select IMS state action table
          RESOURCE THRESHOLDS  Define IMS resource Thresholds
          ...
-----
***** Bottom of data *****

F1=HELP   F2=SPLIT   F3=END   F4=RETURN   F5=RFIND   F6=RCHANGE
F7=UP     F8=DOWN    F9=SWAP  F10=LEFT   F11=RIGHT  F12=RETRIEVE

```

Figure 116. 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 OS/390 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".

IMS ENVIRON

Use this policy item to define additional IMS information for all types of IMS subsystems. This is described in the *System Automation for OS/390 IMS Automation Programmer's Reference and Operator's Guide* in section ENVIRON. This policy item should be defined for every region that is part of an IMS system. For more information, refer to "More About Policy Item IMS ENVIRON" on page 194.

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 113 on page 189 where you can fill in the required information.

STATE ACTION TABLE

If you select this policy item, SA OS/390 will display a list of state/action tables which previously have been defined as described in "IMS State/Action Tables" on page 236. 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, SA OS/390 brings up the following panel where you describe the IMS/DC control region.

```

COMMANDS  HELP
-----
AOFPIXL          IMS control specifications
Command ==> _____

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

Subsystem: IMS

Enter or update the following fields:
APPLid. . . . _____
APPL1 connect _____
APPL2 connect _____
Major node1.. _____ ( SA13 only )
Major node2.. _____ ( SA13 only )
Other major1. _____ ( SA13 only )
Other major2. _____ ( SA13 only )
Default HSBID 1
Startup parm1 _____
Startup parm2 _____

F1=HELP      F2=SPLIT      F3=END      F4=RETURN      F5=RFIND      F6=RCHANGE
F7=UP        F8=DOWN       F9=SWAP     F10=LEFT      F11=RIGHT     F12=RETRIEVE

```

Figure 117. IMS control specifications Panel

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 OS/390 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.

Major node1/2

SA OS/390 1.3 users: If you want SA OS/390 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 2.1 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:

Application Policy Object

- Active: PIPE CORR CMD V NET,ACT,ID=xxx | CONS
- Inactive: PIPE CORR CMD V NET,INACT,ID=xxx | CONS

Other major1/2

SA OS/390 1.3 users: If you want SA OS/390 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 2.1 users: Specify the VTAM V NET,ACT,ID=xxx or V NET,INACT,ID=xxx commands for the required nodes in the 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 CORR CMD V NET,ACT,ID=xxx | CONS
- Inactive: PIPE CORR CMD V NET,INACT,ID=xxx | CONS

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.

More About Policy Item IMS ENVIRON

If you select policy item IMS ENVIRON from the *Policy Selection* panel for IMS subsystems, SA OS/390 brings up the following panel where you describe additional information for IMS subsystems.

```

COMMANDS  HELP
-----
AOFPIEN          IMS Environ specifications
Command ==>>>

Entry Type : Application          PolicyDB Name : DATABASE_NAME
Entry Name : IMS                  Enterprise Name : YOUR_ENTERPRISE
-----
More:
The following field can only be modified on the APPLICATION INFO definitions:
Subtype . . . . . CTL

Enter or update the following fields:
XRF . . . . . YES NO
Shutgo. . . . . YES NO
IMS subsystem ID. . . . . IMS Subsystem ID
Default start type. . . . . AUTO COLD BUILDQ WARMSDBL
Default force . . . . . DUMP NODUMP
Applid connect. . . . . YES NO
Broadcast delay . . . . . Minutes
Force transfer delay. . . . . Minutes
Immed transfer delay. . . . . Minutes
Region recovery delay . . . . . Minutes
Shutdown verify delay . . . . . Minutes
Shutdown exit . . . . . User exit name ( SA13 only )
Control region name . . . . . Subsystemname of owning control region
Start monitoring delay. . . . . Seconds
DB control. . . . . YES NO
XRF details if XRF = YES
Start type (shutdown) . . . . . AUTO COLD BUILDQ WARMSDBL
Start type (recovery) . . . . . AUTO COLD BUILDQ WARMSDBL
XRF Synchronization . . . . . Y S N P T
Default XRF start type. . . . . XRF ACTIVE BACKUP
XRF shut delay. . . . . Seconds
XRF Watch wait. . . . . Seconds
XRF unlock delay. . . . . Minutes
XRF alternate start . . . . . YES NO ( SA13 only )
XRF new alternate start . . . . . YES NO ( SA13 only )
Non automatic takeover. . . . . YES NO
Partner subsystem ID. . . . . IMS subsystem ID of partner IMS
Partner domainid. . . . . Netview domain ID of partner IMS
Partner subsystem . . . . . Subsystem name of backup control region
XRF Watch notify. . . . . Seconds
XRF Watch alert . . . . . Seconds
F1=HELP      F2=SPLIT      F3=END      F4=RETURN      F5=RFIND      F6=RCHANGE
F7=UP        F8=DOWN       F9=SWAP     F10=LEFT      F11=RIGHT     F12=RETRIEVE

```

Figure 118. IMS Environ specifications Panel

The ENVIRON entry serves as the main facility to tailor the operation of IMS Automation. The ENVIRON entry should be coded for each region which is a part of an IMS system. Many of the keyword parameters are not required for non-XRF systems. The syntax specification indicates those keyword parameters required for any IMS control region definition. When defining an XRF IMS subsystem, refer to the description of individual parameters to determine the requirements.

Fill in the following fields:

XRF This parameter is required for the control region only. It specifies whether the control region will be executing as an XRF complex.

Note: Specifying XRF=NO in conjunction with the Partner domain and Partner subsystem keyword parameters indicates an FDR enabled IMS.

Application Policy Object

Shutgo

This parameter is required for the control region only. 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

This parameter is required for the control region only. It 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 OS/390 determines that the subsystem should be running but is not. If the SA OS/390 SETSTATE 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 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.

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.

Control region name

is the SA OS/390 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 can substitute a defined clone value anywhere in the region name using the &AOCCCLONEn. notation. Additionally, you can specify standard OS/390 and NetView cloning variables.

Note that the control region name field is no longer recommended and will be deleted in a later release. Instead, a HASPARENT relationship with a sequence number of 1 should be defined in the APPLICATION policy item for the IMS dependent region pointing to the owning control region.

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.

DB control

This parameter indicates whether the control region is a DB control region. (If YES, then SUBTYPE=CTL and XRF=NO must also be specified.)

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.

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).

- 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.
- 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

Application Policy Object

not participate in any type of synchronization; however, it will be started when the alternate IMS is started.

- 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.
- 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.

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.

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 DFSHSBxx 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 OS/390 subsystem name of the alternate/backup control region. For the dependent regions this is the SA OS/390 subsystem name of the corresponding XRF partner dependent regions. For an FDR enabled system, this is the SA OS/390 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.

Policy Items for OPC Subsystems

Figure 119 on page 200 shows the *Policy Selection* panel for applications that have been created with application type OPC (see entry field *Application Type* in Figure 87 on page 142). 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.

Application Policy Object

```

ACTIONS  HELP
-----
AOFGEPOL                Policy Selection                Row 14 to 22 of 22
Command ==>>>                SCROLL==>> PAGE

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

Action      Policy Name      Policy Description
...
-----OPC SPECIFIC POLICY-----
OPC CONTROL      Define OPC Control entries
-----
...
***** Bottom of data *****

F1=HELP      F2=SPLIT      F3=END      F4=RETURN      F5=RFIND      F6=RCHANGE
F7=UP        F8=DOWN       F9=SWAP     F10=LEFT      F11=RIGHT     F12=RETRIEVE

```

Figure 119. 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 from the *Policy Selection* panel for OPC subsystems, SA OS/390 brings up the following panel where you describe additional information for OPC subsystems.

```

COMMANDS  HELP
-----
AOFPOXL                OPC control specifications
Command ==>>>

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

Subsystem name: OPC

The following field can only be modified on the APPLICATION INFO definitions:
Subtype. . . . . DATASERVER                OPC Controller type

Enter or update the following field:
Subsystem ID . . . _____

F1=HELP      F2=SPLIT      F3=END      F4=RETURN      F5=RFIND      F6=RCHANGE
F7=UP        F8=DOWN       F9=SWAP     F10=LEFT      F11=RIGHT     F12=RETRIEVE

```

Figure 120. OPC control specifications Panel

The *Subtype* parameter describes the type of OPC controller type that is being defined. It can be entered on the APPLICATION INFO policy item of the application or the application class into the *Subtype* field. So 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.

Into the *Subsystem ID* field, you enter the OPC subsystem ID.

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 OS/390 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, in the panel shown in Figure 8 on page 25 you enter 44 or EVT for events into the command line.

This will lead you to a panel as shown in Figure 121. This screen might show a list of already existing events. You type in the command **new** into the command line and press ENTER.

```

COMMANDS  ACTIONS  VIEW  HELP
-----
AOFGENAM          Entry Name Selection
Command ==> new          SCROLL==> PAGE

Entry Type : Event          PolicyDB Name : DATABASE_NAME
                          Enterprise Name : YOUR_ENTERPRISE

Action      Entry Name      C Short Description
EVENT1      EVENT1          Startup event, backup completed
EVENT2A     EVENT2A        Startup event, AOR1 is ready
EVENT2B     EVENT2B        Startup event, AOR2 is ready

```

Figure 121. Entry Name Selection Panel for Events

Now you will see a panel similar to the one shown in Figure 122. Here you specify the event name, the UNSET condition and a description.

```

COMMANDS  HELP
-----
AOFGEN00          Define New Entry
Command ==> _____

To define a new entry, specify the following information:
Type. . . . . : Event
Name. . . . . : EVENT1__

Unset Condition . . . . NONE_ DOWN NONE START UP

Short Description . . . Backup successfully completed_____
Extended Description. . _____
. . _____
. . _____
. . _____
. . _____

```

Figure 122. Define New Event Panel

The unset condition is preset to NONE. You can overtype this input field if desired. For more information on the unset condition, see “More About Policy Item CONDITION” on page 202.

Events Policy Object

Policy Items for Events

Figure 123 shows the policy items that are displayed when an event is selected from the *Entry Name Selection* panel for events.

```

  ACTIONS  HELP
-----
A0FGEPOL          Policy Selection
Command ==>>>                                SCROLL==>> PAGE

Entry Type : Event          PolicyDB Name  : DATABASE_NAME
Entry Name  : EVENT3       Enterprise Name : YOUR_ENTERPRISE

Action      Policy Name      Policy Description
-----
S_____   DESCRIPTION     Enter description
          CONDITION         Define the unset condition
-----
          COPY              Copy data from an existing entry

```

Figure 123. Policy Selection Panel for Events

The unique policy item for *Events* is:

CONDITION

With this policy item you can specify when SA OS/390 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 124 where you can define the *unset condition* of the current event.

```

  COMMANDS  HELP
-----
A0FGAPEV          Unset Condition
Command ==>>> _____

Entry Type : Event          PolicyDB Name  : DATABASE_NAME
Entry Name  : EVENT3       Enterprise Name : YOUR_ENTERPRISE

Verify the following condition

Unset condition . . . NONE_ DOWN NONE START UP

```

Figure 124. Unset Condition Panel for Events

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 125 shows the policy items that are displayed when an existing *Service Periods* entry is selected from the *Entry Name Selection* panel.

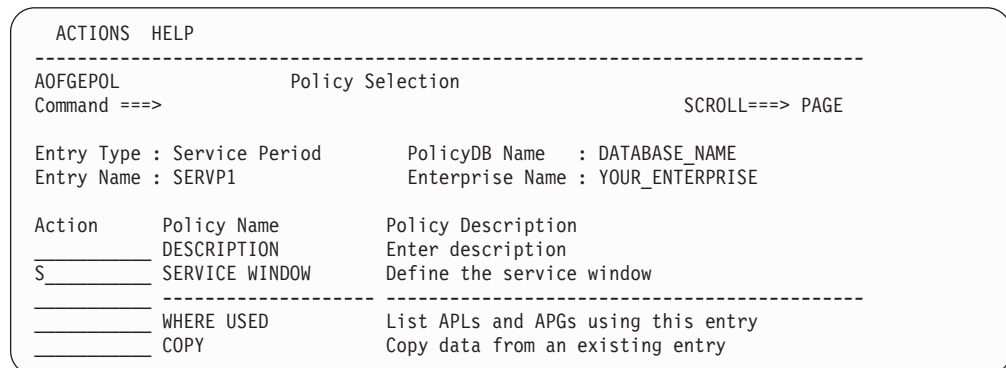


Figure 125. Policy Selection Panel for Service Periods

The unique policy item for *Service Periods* is:

SERVICE WINDOW

With this policy item you define the time frame(s) when an application or an application group should be available and/or unavailable. You need to link a service period with a defined service window to *Applications* or *ApplicationGroups*.

Selecting the SERVICE WINDOW policy item will bring up a panel as shown in Figure 126 on page 204 where you can define the time duration of the service period.

Service Periods Policy Object

```

COMMANDS  HELP
-----
A0FGAPSP          Service Windows
Command ==>>>                                SCROLL==>> PAGE

Entry Type : Service Period          PolicyDB Name : DATABASE_NAME
Entry Name  : SERVPI                 Enterprise Name : YOUR_ENTERPRISE

NOTE: Review help panel for complete description of data entry fields.
      Valid start/stop times are 00:00-24:00, valid priorities (P) are L and H

          P Start/Stop  Start/Stop  Start/Stop  Start/Stop  Start/Stop
Monday  U _ 00:00 24:00  _____  _____  _____  _____
        D _ _____  _____  _____  _____  _____
Tuesday U _ 00:00 24:00  _____  _____  _____  _____
        D _ _____  _____  _____  _____  _____
Wednesday U _ _____  _____  _____  _____  _____
         D _ _____  _____  _____  _____  _____
Thursday  U _ _____  _____  _____  _____  _____
         D _ _____  _____  _____  _____  _____
Friday    U _ _____  _____  _____  _____  _____
         D _ _____  _____  _____  _____  _____
Saturday  U _ 08:00 14:00 _____  _____  _____  _____
         D _ _____  _____  _____  _____  _____
Sunday    U _ _____  _____  _____  _____  _____
         D _ _____  _____  _____  _____  _____
Weekday   U _ 07:00 18:00 _____  _____  _____  _____
         D _ _____  _____  _____  _____  _____
Weekend   U _ _____  _____  _____  _____  _____
         D _ _____  _____  _____  _____  _____
Daily    U _ _____  _____  _____  _____  _____
         D _ _____  _____  _____  _____  _____

```

Figure 126. Service Windows Panel for Service Periods

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 Start/Stop times for the periods when the resource should be available (UP). A line preceded by letter **D** will contain Start/Stop 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:

```

          P Start/Stop  Start/Stop  Start/Stop  Start/Stop  Start/Stop
Monday  U _ 00:00 24:00  _____  _____  _____  _____
        D _ _____  _____  _____  _____  _____
Tuesday U _ 00:00 24:00  _____  _____  _____  _____
        D _ _____  _____  _____  _____  _____

```

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)

Service Periods Policy Object

- **weekend days:** the weekend schedule (Saturday and Sunday)
- **Daily:** all seven days; the daily schedule

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:

- L** 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.
- H** for high priority; H schedules override L schedules and low priority operator requests.

This priority is checked in case two goals get into conflict. The hierarchy of requests and priorities is implemented as follows:

Request from	wins over	request from
-----	-----	-----
Service period high	=====>	service period low, INGREQ low
INGREQ low	=====>	service period low
INGREQ high	=====>	service period high, service period low, INGREQ low

Note: There are other methods of generating requests within SA OS/390. For a complete description of these methods and the implemented hierarchy of their priorities refer to the documentation of the SA OS/390 concepts in *System Automation for OS/390 User's Guide*.

When defining a service period keep in mind that the startup time is the time at which SA OS/390 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 OS/390. 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 **START** and **STOP** times, see the online help of 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 OS/390 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).

Triggers Policy Object

Figure 127 shows the policy items that are displayed when a trigger is selected from the *Entry Name Selection* panel.

ACTIONS HELP		

A0FGEPOL	Policy Selection	SCROLL==> PAGE
Command ==>		
Entry Type : Trigger	PolicyDB Name : DATABASE_NAME	
Entry Name : TRIGGER1	Enterprise Name : YOUR_ENTERPRISE	
Action	Policy Name	Policy Description
	DESCRIPTION	Enter description
S	CONDITION	Define the Trigger condition

	WHERE USED	List APLs and APGs using this entry
	COPY	Copy data from an existing entry

Figure 127. Policy Selection Panel for Triggers

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 128 on page 207 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 128 on page 207.

Note:

This feature is available only for compatibility with SA OS/390 1.3. In SA OS/390 2.1, service periods are directly linked 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 207.
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 207.


```

COMMANDS  ACTIONS  VIEW  HELP
-----
A0FGAPT0                                Trigger Conditions
Command ==>>>                                SCROLL==>> PAGE

Entry Type : Trigger                        PolicyDB Name : DATABASE_NAME
Entry Name : TRIGGER1                       Enterprise Name : YOUR_ENTERPRISE

Note: Service Period definitions are for SA OS/390 Version 1.3 only.
      In this case define them prior to the STARTUP/SHUTDOWN condition.

Action      Condition      Names
S_____    SERVICE PERIOD  SERVP1  1
_____     STARTUP 1        SERVICE EVENT1  2
_____     STARTUP 2
_____     STARTUP 3
_____     STARTUP 4
_____     STARTUP 5
_____     STARTUP 6
_____     STARTUP 7
_____     STARTUP 8
_____     STARTUP 9
_____     STARTUP 10
_____     SHUTDOWN 1      EVENT3  3
_____     SHUTDOWN 2
_____     SHUTDOWN 3
_____     SHUTDOWN 4
_____     SHUTDOWN 5
_____     SHUTDOWN 6
_____     SHUTDOWN 7
_____     SHUTDOWN 8
_____     SHUTDOWN 9
_____     SHUTDOWN 10

```

Figure 128. Trigger Conditions Panel

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 in the panel from Figure 128 (**2**), SA OS/390 will display another panel like the one shown in Figure 129 on page 208.

Triggers Policy Object

```
COMMANDS  ACTIONS  VIEW  HELP
-----
A0FGAPT1          STARTUP Condition for Trigger
Command ==>>>                                SCROLL==>> PAGE

Entry Type : Trigger                          PolicyDB Name  : DATABASE_NAME
Entry Name  : TRIGGER1                        Enterprise Name : YOUR_ENTERPRISE

Description. . . .

Service Period . . YES  YES NO

Action      Status  Event      Short Description
-----
            SELECTED  EVENT1     Startup event, backup completed
            EVENT2A   Startup event, AOR1 is ready
            EVENT2B   Startup event, AOR2 is ready
            EVENT3     Shutdown event in case of emergency
```

Figure 129. *STARTUP Condition for Trigger Panel*

Figure 129 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** into the *Service Period* entry field connects the already selected service period to this condition. In our scenario, you can only connect service period **SERVPI** to any of the conditions for **TRIGGER1**. So you will enter **YES** into the *Service Period* entry field only for **STARTUP 1**.

Note:

This feature is available only for compatibility with SA OS/390 1.3. In SA OS/390 2.1, service periods are directly linked to application groups or applications.

If you type an S in front of a SHUTDOWN condition in the panel from Figure 128 on page 207 (**3**), SA OS/390 will display another panel similar to the one shown in Figure 129. 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

Throughout this publication the term *processor operations PC* (*ProcOps PC*) is used to refer to the personal computer (PC) that is connected to one or more target systems and which contains the processor operations distributed feature code. In the worksheets and SA OS/390 panels, the terms "PC", "PS/2" or "ProcOps PS/2" are used to refer to this workstation. The workstation may be a Micro Channel[®] Architecture (MCA) bus personal computer, a PCI bus personal computer, or an IBM-compatible ISA bus personal computer.

The *processor* resource object enables you to control your processors.

Creating a New Processor

How to create new policy objects in general, is described in “How to Create New Policy Objects” on page 25. Figure 130 shows a panel for defining a new *Processor* policy object.

```

COMMANDS  HELP
-----
AOFGCN00          Define New Entry

To define a new entry, specify the following information:
Type. . . . . Processor
Name. . . . .

Mode . . . . . LPAR   ESA   XA   370
Processor Type . . . . . eServer 9672 9674 2003 3000
                          9021 9121 3090 4381 308X

Model. . . . .
Physical Partition . . . SI   A   B
Short Description . . .
Extended Description. .
                    :
                    :
                    :

```

Figure 130. Defining a New SA OS/390 Processor Entry

Name The processor name needs to have 1 to 8 alphanumeric characters. Name must begin with an alphabetic character, #, \$, or @ and be different from the names of all groups, subgroups, systems (including the ProcOps names), processors, and communications paths defined in the current policy database.

Mode Is required. For zSeries, 390-CMOS, or 9021, 9121, 3090 processors in LPAR mode, specify LPAR. For a TPF system, specify ESA. For a Linux system specify LINUX if the processor hardware is configured in 'LINUX Only' mode, otherwise specify ESA.

Processor Type
Is required. Select from list.

Model Is an optional field, used for documentation only. It can be up to four alphanumeric characters.

Physical Partition
Is an optional field, used for documentation only. It indicates whether a system is physically partitioned. Enter A or B to indicate which side of the processor is the target hardware whose name is in the corresponding Name field. Enter SI (single image) to indicate processor is not physically partitioned.

Depending on the type of processor you entered in the previous panel, you next see a panel for an OCF-based or screen-oriented processor. An OCF-based processor uses an operations command facility for interacting with human operators or external programs to perform operations management functions. A screen-oriented processor is one that interacts with human operators through line-mode display consoles. For System/390 and System/370 processors, the display consoles use either an IBM 3270 data stream or an ASCII data stream.

Processor Policy Object

Policy Items for Processors

If your processor is OCF-based, you specify:

- Description
- Processor Info
- NVC path name
- Op Cons Connections (Optional)

If your processor is screen-oriented, you specify:

- Description
- Processor Info
- Op Cons Connections
- CP States
- Sys Cons Connections
- Op Cons Connections
- Channel Path IDs

The following figures show the two versions of the *Policy Selection* panel.

ACTIONS		HELP
AOFGEPOL		Policy Selection
Command ==>>		SCROLL==>> PAGE
Entry Type : Processor		PolicyDB Name : DATABASE_NAME
Entry Name : CHITARG3		Enterprise Name : YOUR_ENTERPRISE
Action	Policy Name	Policy Description
	DESCRIPTION	Enter description here
	PROCESSOR INFO	Enter and display Processor information
	NVC PATH NAME	Define name of NetView connection path
	-----OPTIONAL POLICY-----	
	OP CONS CONNECTIONS	Define connections for operator console

	WHERE USED	List Systems linked to this entry
	COPY	Copy data from an existing entry

Figure 131. Policy Selection Panel for Processors (OCF-based)

ACTIONS		HELP
AOFGEPOL		Policy Selection
Command ==>>		SCROLL==>> PAGE
Entry Type : Processor		PolicyDB Name : DATABASE_NAME
Entry Name : CHITARG3		Enterprise Name : YOUR_ENTERPRISE
Action	Policy Name	Policy Description
	DESCRIPTION	Enter description here
	PROCESSOR INFO	Enter and display Processor information
	CP STATES	Define target hardware CP states
	SYS CONS CONNECTIONS	Define connections for system console
	OP CONS CONNECTIONS	Define connections for operator console
	CHANNEL PATH IDS	Define mandatory and optional CHPIDs

	WHERE USED	List Systems linked to this entry
	COPY	Copy data from an existing entry

Figure 132. Policy Selection Panel for Processors (Screen-Oriented)

PROCESSOR INFO

You may update: - Processor mode - Processor type - Processor model - Processor physical partition information

NVC PATH NAME

This policy item is displayed for an OCF-based processor. This policy item allows you to enter or update the name of the NetView connection path between the focal point system and the target system. This name needs to be defined as a communications path object of type NVC. See Figure 137 on page 218 and Figure 144 on page 228 for more information.

CP STATES

This policy item is displayed for a screen-oriented processor. This policy item allows you to indicate the CPs (and vector processors, if appropriate) that are available on the processor. You can also indicate the amount of central storage (and expanded storage, if appropriate) associated with the processor. This information is for documentation purposes only.

SYS CONS CONNECTIONS

This policy item is displayed for a screen-oriented processor. This policy item allows you to define the primary and backup PCs and 3270 connection cards for the system console of the Processor. See “System Console Connections” on page 213 for more information.

OP CONS CONNECTIONS

This policy item is displayed for either an OCF-based or a screen-oriented processor. This policy item allows you to define the PC and connection cards for operator console connections for the processor. Up to 30 operator console connections may be defined. Use of this policy item is optional for OCF-based processors. See “OpCons Connections” on page 215 for more information.

CHANNEL PATH IDS

This policy item is displayed for a screen-oriented processor. This policy item allows you to specify which channel paths must be online to activate a non-(target system associated with this processor. See “Channel Path IDs” on page 216 for more information.

CP States for Screen-Oriented Processors

To define a screen-oriented processor, you need one of the following panels.

Your choice depends on your processor. If the processor is:

- 308X, see “CP States for a 308x Processor” on page 131
The Central storage size and CP fields can be changed without generating and loading a new processor control file. You can do this by issuing the host ISQVARS command. However, these changes are not saved if processor operations or SA OS/390 is stopped or restarted.
- 4381, see “CP States for a 4381 Processor” on page 131
The Central storage size and CP fields can be changed without generating and loading a new SA OS/390 processor operations control file. You can do this by issuing the host ISQVARS command. However, these changes are not saved if processor operations or SA OS/390 is stopped or restarted.
- 9021, 9121, 3090, see Figure 133 on page 212 to change the states of the central processors and allocate storage size of a target hardware definition.

Processor Policy Object

```
ISQDPS07                               Target Hardware CP States
Command ==>>>

Target hardware: CHITARG1

Enter information for the target hardware.

Central storage size => 102             Megabytes
Expanded storage size => 134           Megabytes

CP      Online      Vector Processor
      (ON or OFF)  (YES or NO)
0 => ON             YES
1 => ON             YES
2 => ON             NO
3 => ON             NO
4 => ON             YES
5 => ON             NO
6 => OFF            NO
7 => OFF            NO
8 => OFF            NO
9 => OFF            NO
```

Figure 133. Target Hardware CP States Panel

The panel contains the following fields:

Target hardware

Indicates the name of the target processor.

Central storage size

Indicates the amount of primary storage (in megabytes) associated with the target processor.

Expanded storage size

Enter the amount of expanded storage (in megabytes) associated with the target processor.

CP Indicates the target processor CP number.

Online

Enter ON next to the CPs that are available on the target processor. Enter OFF next to the CPs not available.

Note: 9121 processors support CP numbers 1, 2, 3, and 4 only. Do not specify ON for CP numbers 0, 5, 6, 7, 8, or 9 of a 9121 processor.

Vector processor

Enter YES in the Vector Processor column to indicate that a vector processor is available for a CP. Enter NO in the Vector Processor column to indicate that a Vector Processor is not available for a CP.

Note: 9121 processors support vector processor numbers 1, 2, 3, and 4 only. Do not specify YES for vector processor numbers 0, 5, 6, 7, 8, or 9 of a 9121 processor.

More about the CONNECTIONS Policy Items

This section provides more information on the kinds of connections for processors offered by the customization dialog:

- “System Console Connections”
- “OpCons Connections” on page 215

System Console Connections

Use the panel shown in Figure 134 to define the primary and backup PCs and 3270 connection cards for the system console of the selected target hardware definition. The panel contains the following fields:

ISQDPS08 Connections
 Command ==>>

Target hardware: CHITARG1

Enter information for the target hardware.

Active system console

PS/2 name =>	
card type =>	3270 or ASCII
card number =>	1-4 for 3270, 1-3 for ASCII

Backup system console

PS/2 name =>	
card type =>	3270 or ASCII
card number =>	1-4 for 3270, 1-3 for ASCII

Figure 134. Connections

Target hardware

Gives the name of the target hardware definition. The name cannot be modified on this panel.

Active system console

PS/2 name

Enter the name of the PC connected as the Active system console. Names can contain from 1 to 8 alphanumeric characters, starting with an alphabetic character, #, \$, or @. This name must be defined as a communications path of type PS/2. See Figure 137 on page 218

card type

Enter either 3270 or ASCII to indicate the card type that provides the connection with the active system console. The card type must match that specified when the PC named as the “Active system console PS/2 name” was defined as a communications path. See Figure 141 on page 221.

For a 9121 system console connection, specify ASCII. For all other system console connections, specify 3270.

card number

Enter the number assigned to the card, for example by the PC

Processor Policy Object

Reference Diskette, when the PC was configured. See also *System Automation for OS/390 Planning and Installation*.

When an ASCII card provides the system console connection, the valid values are 1, 2, or 3, which correspond to COM1, COM2, or COM3.

When a 3270 connection card provides the system console connection, the valid values are 1, 2, 3, or 4. These values correspond to the first, second, third, and fourth numbers assigned at configuration time. If the PC has been configured by a Reference Diskette, it designates the card number as *only choice*, specify the card number as 1.

The PC Reference Diskette may not display the Resource Used as first, second, third, or fourth, but may instead display an I/O address. The I/O addresses correspond to the "Resource Used", as follows:

Table 7. Mapping of Resources Used to I/O Address

Resource Used	Card Number	I/O Address
Only Choice		02D0h – 02DFh (3270 Connection Version A)
First	1	02D0h – 02DFh
Second	2	06D0h – 06DFh
Third	3	0AD0h – 0ADFh
Fourth	4	0ED0h – 0EDFh

The card number must match the card number specified for the PC when the communications path was defined.

Backup system console

PS/2 name

Enter the name that identifies which PC is the backup system console. This field is optional. Names can contain from 1 to 8 alphameric characters, starting with an alphabetic character, #, \$, or @.

This name must match the *PS/2* name specified for the PC when the communications path was defined.

The card number must match the card number specified for the PC when the communications path was defined.

card type

Enter either 3270 or ASCII to indicate the card type that provides the connection with the backup system console. This field is optional. The backup system console card type must be the same as the active system console card type.

For a 9121 system console connection, specify ASCII. For all other system console connections, specify 3270.

card number

Enter the number assigned to the card, for example by the PC Reference Diskette, when the PC was configured. See also *System Automation for OS/390 Planning and Installation*. This field is optional.

Processor Policy Object

When an ASCII card provides the system console connection, valid values are 1, 2, or 3 corresponding to COM1, COM2, or COM3.

When a 3270 connection card provides the system console connection, valid values are 1, 2, 3, or 4. These values correspond to the first, second, third, and fourth numbers assigned at configuration time. If the PC has been configured by a Reference Diskette, it designates the card number as *only choice*, specify the card number as 1.

The PC Reference Diskette may not display the "Resource Used" as first, second, third, or fourth, but may instead display an I/O address. The I/O addresses correspond to the "Resource Used", as listed in Table 7 on page 214.

OpCons Connections

Regardless of whether you are defining an OCF-based or a screen-oriented target processor, if your choice is OP CONS CONNECTIONS, Figure 135 is displayed. (If you have an OCF-based processor, use of this panel is optional. If no operator console connections are defined for OCF-based processors, console integration will be used to provide connectivity to the operator console. Due to performance considerations, it is highly recommended that you use this panel to define one or more connections to your operator console.)

To define the PC and connection cards for operator console connections for the selected target hardware definition, use the panel shown in Figure 135

The panel contains the following fields:

```
ISQDPS8A                      Operator Console Connections
Command ==>>>

Target hardware: TESTHW

Enter the operator console physical connections.
Card type: ASCII or 3270
Card number-ASCII: 1 to 3   Card number-3270: 1 to 4
```

PS/2 Name	Card Type	Card Number	PS/2 Name	Card Type	Card Number	PS/2 Name	Card Type	Card Number
TESTPS2_	3270_	2	_____	_____	-	_____	_____	-
_____	_____	-	_____	_____	-	_____	_____	-
_____	_____	-	_____	_____	-	_____	_____	-
_____	_____	-	_____	_____	-	_____	_____	-
_____	_____	-	_____	_____	-	_____	_____	-
_____	_____	-	_____	_____	-	_____	_____	-
_____	_____	-	_____	_____	-	_____	_____	-
_____	_____	-	_____	_____	-	_____	_____	-
_____	_____	-	_____	_____	-	_____	_____	-

Figure 135. Operator Console Connections

Target hardware

Gives the name of the target hardware definition. The name cannot be modified on this panel.

PS/2 Name

Enter the name of the PC connected as the operator console. Names can contain from 1 to 8 alphameric characters, starting with an alphabetic

Processor Policy Object

character, #, \$, or @. This name must match the *PS/2 name* specified for the PS/2 when the communications path was defined.

Card Type

Enter the card type of the connection card (ASCII or 3270).

Card Number

Enter the number assigned to the card, for example by the PC Reference Diskette, when the PC was configured. See also *System Automation for OS/390 Planning and Installation*.

The card number must match the card number specified for the PC when the communications path was defined.

When an ASCII card provides the operator console connection, valid values are 1, 2, or 3, corresponding to COM1, COM2, or COM3.

When a 3270 connection card provides the operator console connection, valid values are 1, 2, 3, or 4. These values correspond to the first, second, third, and fourth numbers assigned at configuration time. If the PC has been configured by a PC Reference Diskette, it designates the card number as *only choice*, specify the card number as 1.

The PC Reference Diskette may not display the "Resource Used" as first, second, third, or fourth, but may instead display an I/O address. The I/O addresses correspond to the "Resource Used", as listed in Table 7 on page 214. follows:

Entries on panel ISQDPS8A are positional. If you rearrange the locations of PS/2 names on panel ISQDPS8A, you likewise change the names of the active and backup target system consoles that have been chosen.

Channel Path IDs

To specify mandatory and optional channel path identifiers (CHPIDs), use the panel shown in Figure 136 on page 217.

```

ISQDPT10                      CHPID Assignments
Command ==>
CHPIDs have been changed or browsed.
Target Hardware: CHITARG1
CHPID states: . = N/A  M = Mandatory  O = Optional
  -0  -1  -2  -3  -4  -5  -6  -7  -8  -9  -A  -B  -C  -D  -E  -F
0-  M  .  .  M  .  .  M  .  .  M  .  .  M  .  .  M
1-  M  .  .  M  .  .  M  .  .  M  O  O  O  O  O  O
2-  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
3-  O  .  .  O  .  O  .  O  .  O  .  O  .  O  .  O
4-  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
5-  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
6-  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
7-  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
8-  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
9-  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
A-  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
B-  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
C-  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
D-  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
E-  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
F-  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .

Range => __ - __  State => _  Set sequence of CHPIDs to state specified above

```

Figure 136. CHPID Assignments

The panel contains the following fields:

Target Hardware

Gives the name of the target hardware definition. The name cannot be modified on this panel.

CHPID states

Move the cursor to the correct position and type the CHPID assignment. Valid values are:

- M** Must be online for the target system to be activated.
- O** Optional.
- .** Not checked by SA OS/390 during activation.

Range Designate a group of CHPIDs to be set to the specified state. Enter the row and column designator for the beginning and end of the range. For example, 00 - 7F specifies a range of 128 CHPIDs. You can specify ranges multiple times.

Note: Channels 80 to FF can be used only by ES/9000® processor models that support 256 channels.

State Enter the CHPID assignments (M, O, or .[period]) for the specified range. The dialog sets each CHPID in the range to this state.

Notes:

1. You can make multiple changes to this panel without exiting. When you have finished assigning CHPIDs, press PF3.
2. SA OS/390 provides channel resource checking for 9021, 9121, and 3090 processors only during the operation of the ISQXDLC, ISQXIPL, and ACTIVATE commands.

Communications Path Policy Object

The *Communications Path* policy object enables you to define and customize your communications paths. The two types of communications path used in SA OS/390 processor operations are the NetView connection (used with OCF-based processors) and the PC (used with both screen-oriented and OCF-based processors).

Creating a New Communications Path

How to create new policy objects in general, is described in “How to Create New Policy Objects” on page 25.

```
COMMANDS  HELP
-----
A0FGPN00          Define New Entry
Command ==>>>

To define a new entry, specify the following information:
Type. . . . . Communications Path
Name . . . . .

LU Name. . . . .
Path Type. . . . .          NVC PS2
Short Description. . . . .
Extended Description . . .
. . .
. . .
. . .
```

Figure 137. Defining a New SA OS/390 Communications Path Entry

The unique fields are:

Name Is required. This gives a unique 1 to 8 alphameric character name. Name must begin with an alphabetic character, #, \$, or @ and be different from the names of all groups, subgroups, systems (including ProcOps names), processors, and communications paths defined in the current policy database.

LU Name Is required. This gives a unique 1 to 8 alphameric character name to the logical unit (LU) of the processor operations PC or the NetView connection to the support element. The LU name must begin with an alphabetic character, #, \$, or @. This name is assigned by the VTAM coordinator.

Path Type Is required. Choose either NVC (NetView connection to a support element) or PS2 (processor operations PC)

Policy Items for Communications Path

There are two types of communications paths discussed in this section:

- Figure 138 on page 219, described in “Processor Operations PS/2 Communications Path” on page 219
- Figure 139 on page 219, described in “NetView Connection” on page 227

Communications Path Policy Object

```
ACTIONS  HELP
-----
AOFGEPOL          Policy Selection
Command ==>>>                                SCROLL==>> HALF

Entry Type : Communications Path  PolicyDB Name : DATABASE_NAME
Entry Name : CHARLIE1            Enterprise Name : YOUR_ENTERPRISE

Action   Policy Name      Policy Description
        DESCRIPTION      Enter description
        PS/2 CHARACTERISTICS Define PS/2 characteristics
        PORT CONFIGURATIONS Define 3270 and ASCII ports
-----
        COPY              Copy data from an existing entry
```

Figure 138. Policy Selection Panel for PS/2 Communications Path

```
ACTIONS  HELP
-----
AOFGEPOL          Policy Selection
Command ==>>>                                SCROLL==>> HALF

Entry Type : Communications Path  PolicyDB Name : DATABASE_NAME
Entry Name : CHARLIE2            Enterprise Name : YOUR_ENTERPRISE

Action   Policy Name      Policy Description
        DESCRIPTION      Enter description
        NVC DEFINITION    Define characteristics of a NVC path
-----
        COPY              Copy data from an existing entry
```

Figure 139. Policy Selection Panel for NetView Connection Communications Path

The fields and options are:

PS/2 CHARACTERISTICS

This displays Figure 140 on page 220.

PORT CONFIGURATIONS

This displays Figure 141 on page 221.

NVC DEFINITION

This displays Figure 144 on page 228.

Processor Operations PS/2 Communications Path

Use the panel shown in Figure 140 to display or change the processor operations focal point port letter, PS/2 control port letter, initial PS/2 status, and WWV installed specifications for the selected personal computer.

Communications Path Policy Object

```
ISQDPP05                      Processor Operations PS/2 Characteristics
Command ==>>>

PS/2: CALVIN
LU Name: LU3456
Communications Task: 0

Enter information for the PS/2.

Focal point port letter      => E
PS/2 control port letter    => R
Initial PS/2 status         => ACTIVE   ACTIVE or INACTIVE
WWV installed                => YES     YES or NO
```

Figure 140. PS/2 Characteristics

The panel contains the following fields:

PC Gives the name of the PC.

LU Name

Displayed for information only and cannot be modified from this screen.

Communications Task

Displayed for information only and cannot be modified from this screen. Gives number of the communications task associated with the PC. If the computer is not currently assigned to a communications task, this field is 0.

Focal point port letter

Enter the letter of the port (A through Z) the PC uses for the focal point system connection. Port letters must be unique within one PC definition.

PS/2 control port letter

Enter the letter (A through Z) of the PC control port. Some processor operations commands address the PC control port only. Port letters must be unique within one PC definition.

Initial PS/2 status

If communication with the PC should be automatically activated at the startup of SA OS/390 processor operations, enter ACTIVE. If not, type INACTIVE.

Note: If you choose to initialize the PC automatically, be aware that when you start SA OS/390 processor operations, the ISQSTART command may complete before the PC is initialized.

WWV installed

If the WWV time synchronization hardware is installed at the PC, enter YES or Y. If not, type NO or N.

Port Configurations

Use the panel shown in Figure 141 to describe the number of 3270 connection and serial cards installed in the current PC.

```

ISQDPP01          Processor Operations Port Configurations
Command ==>>>

PS/2: CALVIN

To select or delete, use Action column.
To add a card, enter information on the Add line.
Actions:   S = Select Card   D = Delete Card
Action    Type  Address/Number
Add =>
-         3270  FIRST
-         3270  SECOND
-         ASCII  COM1
    
```

Figure 141. Processor Operations Port Configurations Panel

The panel contains the following fields:

PC Gives the name of the PC.

Add Enter the card type and card address/number.

Action

Presents the action you can select. Enter one of the following:

S Choose the connection card to describe.

D Delete the selected connection card.

Type Indicates the card type (3270 or ASCII).

The card type specified on panel ISQDPP01 must be the same as the card type specified on either of panels ISQDPS08 (see “More about the CONNECTIONS Policy Items” on page 213) or ISQDPS8A (see “OpCons Connections” on page 215).

Address/Number

Enter the number assigned to the card, for example by the PC Reference Diskette, when the PC was configured. See also *System Automation for OS/390 Planning and Installation*.

When an ASCII card provides the system console connection or the operator console connection, the values can be COM1, COM2, or COM3.

When a 3270 connection card provides the system console or the operator console connection, the values can be FIRST, SECOND, THIRD, or FOURTH. These values correspond to the first, second, third, and fourth numbers assigned at configuration time. If the PC has been configured by a PC Reference Diskette, it designates the card number as *only choice*, specify the card number as 1.

The PC Reference Diskette may not display the “Resource Used” as first, second, third, or fourth, but may instead display an I/O address. The I/O addresses correspond to the “Resource Used”, as follows: The I/O addresses correspond to the “Resource Used”, as listed in Table 7 on page 214.

The card number specified on ISQDPP01 must correlate to the card number specified on either of panels ISQDPS08 or ISQDPS8A, as follows:

Communications Path Policy Object

Table 8. Correlation of Card Number Fields (for 3270 Card Type)

ISQDPS08 or ISQDPS8A	ISQDPP01
1	FIRST
2	SECOND
3	THIRD
4	FOURTH

Table 9. Correlation of Card Number Fields (for ASCII Card Type)

ISQDPS08 or ISQDPS8A	ISQDPP01
1	COM1
2	COM2
3	COM3

3270 Port Description

Use the panel shown in Figure 142 to define the 3270 connection card and its relation to the target system.

The panel contains the following fields:

```

ISQDPP06                               3270 Port Description
Command ==>>

PS/2:      KGNPS2
3270 port: 1

Enter information for the 3270 port.

Port letter           => A
Console type         => MVS___
Command interval     => 5
Inbound retry count  => 2
Screen handler poll timer => 10__
Response timer       => 5___
VM console interval  => ___
MVS/VSE console interval => 60__
Control unit country code => 01
Console model        => MOD2
Target hardware name  => MP9121__
Passthru Screen Delay => 03
  
```

Figure 142. 3270 Port Description Panel

PS/2 Gives the name of the PC.

3270 port

Gives the number of the 3270 card being added or selected.

Port letter

Enter the port letter (A through Z) of this 3270 connection card. The default is a blank.

Console type

Enter the console type designator. Valid values are:

- SYS9021 for system console connection to 9021 processor
- SYS3090 for system console connection to 3090 processor
- SYS4381 for system console connection to 4381 processor

Communications Path Policy Object

- SYS308x for system console connection to 308x processor
- MVS for operator console connection to an MVS system
- VM for operator console connection to a VM system
- VSE for operator console connection to a VSE system
- TPF for operator console connection to a TPF system

The default is a blank.

Command interval

Enter the time interval (in tenths of a second) that the PC is to wait between issuing successive commands to a target console. Valid values are 0 to 60, which indicates the number of tenths of a second (0 to 6 seconds). The default value is 5 (1/2 second).

Inbound retry count

Enter the number of times (0 through 9) the PC is to attempt to retransmit a message to the focal point system when certain communications problems are detected. The default value is 2.

Screen handler poll timer

Enter the time interval (0 through 2000 in seconds) the PC is to wait before polling the port to verify that it is still responding. The default value is 10 seconds.

Response timer

Enter the time interval (0 through 1999 in seconds) the PC is to wait for the target console to accept input. This value must be less than the Screen handler poll timer value. The default value is 5 seconds.

VM console interval

Enter the time interval (0 through 2000 in seconds) that a VM READ or CP READ can stay on a VM console before a message is generated instructing you to take appropriate action. This field is used only for a VM console type. If this field is set to 0, the message is disabled. The default value is 60 seconds.

MVS/VSE console interval

Enter the time interval (0 through 2000 in seconds) the PC is to wait after a screen change before it clears the screen again with CONTROL E command. The default value is 60 seconds.

Control unit country code

Enter the value that corresponds to the control unit country code specified in the 3x74 to which this port is connected. Valid values are:

- | | |
|----|-------------------|
| 01 | English (US) |
| 03 | Austrian/German |
| 04 | Belgian |
| 05 | Brazilian |
| 07 | Danish |
| 09 | Finnish |
| 14 | International |
| 15 | Italian |
| 16 | Japanese English |
| 17 | Japanese Katakana |

Communications Path Policy Object

- 21 Spanish-speaking
- 22 English (UK)
- 23 Norwegian
- 24 Swedish
- 25 EBCDIC WT
- 29 Canadian bilingual
- 30 French
- 47 Netherlands

The default value is 01 for the United States.

Console model

Enter the model type for the system or operator console. The model type can be MOD2, MOD3, MOD4, MOD5, or 3205. Consider the following rules in selecting the model type:

- For a system console that connects to a 9021, 3090, or a 308x processor, you must select MOD2.
- For a system or operator console that connects to a 4381 processor, you must satisfy the following requirements:
 - All system and operator consoles that connect to a 4381 processor must have the same model type, either 3205 or MOD2.
 - The model type you select depends on the type of terminal. For a 3205 terminal, select 3205. For a 3278-2A terminal, select MOD2.

The same rules apply if you are defining an operator console that connects to a 4381 integrated communications adapter.

- For an operator console that connects through a 3x74 control unit, you can select MOD2, MOD3, MOD4, or MOD5; the choice depends on the operating system, as Table 10 shows. The customization dialog does not verify that the model type is valid for the selected console type.

Table 10. Operator Console Types Supported

Operating System	Console Model Type			
	MOD2	MOD3	MOD4	MOD5
VM	X			
	X	X	X	
MVS	X	X	X	
VSE	X			
TPF	X	X	X	

Note: The pass-through function of the ISQTCC command is restricted to MOD2 and 3205 console types. This restriction applies to the focal point system consoles and target system consoles.

Because the MOD2 and 3205 console types are limited to 80 characters per line, a one-line message issued from a console with a longer line may display on a MOD2 or 3205 console as multiple lines.

Target hardware name

Enter the name of the processor object associated with this console. Names can contain from 1 to 8 alphanumeric characters, starting with an alphabetic character, #, \$, or @.

Passthru screen delay

This two-character field enables you to change the rate at which the screen is updated. This field is a pacing value for rapid-display consoles. It has the effect of reducing network traffic for high activity display. It can be a numeric value from 0 to 99 seconds. If no value is specified, the default is 0 seconds.

Note: You should set this to the lowest value that will avoid input problems.

ASCII Port Description

Use the panel shown in Figure 143 on page 226 to define the use of the PC ASCII or serial ports.

System console port descriptions for the 9121 processors are shown on this panel rather than on the *3270 Port Description* panel, because the RS-422 card is an ASCII card, not a 3270 card.

You do not have to specify the screen handler or console model type of a system console port for the processors mentioned above. SA OS/390 automatically uses the SYS9021 or SYS9121 screen handler and defines the console model type as MOD2 for these ports. (The operator will see these values in status displays.)

The values shown in the figure are examples for these fields with the RS-422 port. When filling in this panel for an RS-422 card, provide values for the following fields only:

- Port letter
- Baud rate
- Data length
- Parity
- Number of stop bits
- Inbound retry count
- Screen handler poll timer
- Response timer
- Target hardware name
- Passthru screen delay (optional)

Communications Path Policy Object

```
ISQDPP07                               ASCII Port Description
Command ==>
PS/2:      KGNPS2
ASCII port: 2

Enter information for the ASCII port.

Port letter           => S
Baud rate             => 19200
Data length           => 8
Parity                => ODD_
Number of stop bits   => 1
XON code              => 11
XOFF code              => 13
ATTN code             => 12
Timeout code          => 1E
Enter code            => 05
End of message code   => 0D0A0C____
Inbound retry count   => 2
Screen handler poll timer => 10__
Response timer        => 5__
Target hardware name  => MP9121__
Passthru screen delay => 00
```

Figure 143. ASCII Port Description Panel

The panel contains the following fields:

PS/2 Gives the name of the PC.

ASCII port

Gives the ASCII or serial port to be defined.

Port letter

Enter the letter (A through Z) of this port. The default is a blank. For proper operation, however, you must fill in the field.

Baud rate

Enter the baud rate for this connection. Valid values are 300, 1200, 2400, 4800, 9600, or 19200. The default is 19200. The default will appear on the panel when it first appears. For ES/9000 processors, this value should be either 9600 or 19200, depending on the EC level of the service processor. For TPF systems, the value you specify depends on the ASCII protocol converter used at the target system.

Data length

Enter the 1-digit word length (number of data bits in a byte): 7 or 8. For ES/9000 processors, do not change this value; use the default. For TPF systems, the value you specify depends on the ASCII protocol converter used at the target system.

Parity Enter the type of parity checking for this connection. Valid values are EVEN, ODD, NONE, MARK, or SPACE. ODD is the default. For ES/9000 processors, this field must be ODD. For TPF systems, the value you specify depends on the ASCII protocol converter used at the target system.

Number of stop bits

Enter 1 or 2 to indicate how many bits follow each character to identify this as the end of the character. For ES/9000 processors, do not change this value; use the default. For TPF systems, the value you specify depends on the ASCII protocol converter used at the target system.

XON code

Enter the 2-hexadecimal-digit code that indicates the PC buffer is ready to receive communications from the focal point system. The default value is 11.

XOFF code

Enter the 2-hexadecimal-digit code that indicates the PC buffer receiving communications from the target system is full. The default value is 13.

ATTN code

Enter the 2-hexadecimal-digit code that indicates to the target system that the Attention key has been pressed at the specified PC. The default value is 12.

Timeout code

Enter the 2-hexadecimal-digit code that indicates a timeout character sent from the PC at a set time interval so the ASCII connection hardware knows the PC is still operational. The default value is 1E.

Enter code

Enter the 2-hexadecimal-digit code that will notify the target system that the Enter key has been pressed at the specified PC. The default value is 05.

End of message code

Enter the 2-to-8-hexadecimal-digit code that will notify the target system of the end of the message being sent from the specified PC. The default value is 0D0A0C.

Inbound retry count

Enter the number of times (0 through 9) the PC is to attempt to retransmit a message to the focal point system when certain communications problems are detected. The default value is 2.

Screen handler poll timer

Enter the time interval (0 through 2000 in seconds) the PC is to wait between messages received from the target system console. The default value is 10 seconds.

Response timer

Enter the time interval (0 through 1999 in seconds) the PC is to wait for the target console to accept input. This value must be less than the screen handler poll timer value. The default value is 5 seconds.

Target hardware name

Enter the name of the target hardware (processor) connected to this port. Names can contain from 1 to 8 alphanumeric characters, starting with an alphabetic character, #, \$, or @.

Passthru screen delay

This 2-character field enables you to change the rate at which the screen is updated. This field is a pacing value for rapid-display consoles. It reduces network traffic for high-activity display. It can be a numeric value from 0 to 99 seconds. You should set this to the lowest value that will avoid input problems.

NetView Connection

If you choose NVC DEFINITION, you can use the panel shown in Figure 144 to supply information when you add or select an NVC type communications path.

Communications Path Policy Object

Note: From the Hardware Management Console, you must enable the OCF-based processor's support element to accept operations management commands from the SA OS/390 focal point system and specify the authorization token. If this is not done, you will receive message ISQ421I during runtime operation.

The panel contains the following fields:

```
ISQDPP09                               NetView Connection Path Definition
Command ===>

Path Name: CPCPATH
LU Name:   LU1234

Enter information for the NetView Connection path.

Target Hardware Name    =>  CPC1_____
Path NETID              =>  CPCNET1
Path Auth Token         =>  CPCATH1
Path Poll Frequency     =>  15                (1 to 99 minutes)
Path Poll Retries      =>  2                (0 to 99)
```

Figure 144. NetView Connection Path Definition Panel

Target Hardware Name

Enter the target hardware name (name of the processor object) associated with this path.

Path NETID

Enter the SNA network ID of the target hardware support element.

Path Auth Token

Enter the operations management authorization token for the support element. This is a 4- to 8-alphanumeric character string defined at the support element during installation. It is used to verify the authority of a user to issue an operations management command.

The support element provided default for the authorization token is *OKOK*.

Path Poll Frequency

Enter how often (in minutes) SA OS/390 processor operations should poll the support element for status. Valid values are 1 to 99 minutes.

Path Poll Retries

Enter the number of attempts to retry that should occur before SA OS/390 processor operations notifies the operator that the connection failed. Valid values are 0 to 99 retries.

Communications Path Commands

Three commands facilitate the task of assigning processor operations PCs to communications tasks. These commands are found in the Commands pull-down menu of both the Communications Path *Entry Name Selection* panel and the Communications Task *Entry Name Selection* panel.

- DISTRPS2 (T)
Distribute PCs
- DISPUNAP (P)
Display Unassigned PCs
- DISPASGN (A)
Display Task-PC assignments

For details, see “Communications Task Commands” on page 282.

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 145 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”
- CICS, see “Defining Automation for CICS Components” on page 230
- IMS, see “Defining Automation for IMS Components” on page 235
- OPC, see “Defining Automation for OPC Components” on page 240

```

MENU  OPTIONS  HELP
-----
AOFGEPOM      Entry type selections for Product Automation
Command====>

DB2 components          IMS components
10 DEN      DB2 System defaults    20 ISA      State/Action tables
21 ISF      IMS XRF status files
22 IRN      IMS XRF Resources

OPC components          CICS components
30 OEN      OPC System details      40 CSA      State/Action tables
31 OCS      Controller details      41 CCN      Link definitions
32 OSR      Special resources        42 CVP      Link Monitor periods
33 ODM      Workstation domains

F1=HELP      F2=SPLIT      F3=END      F4=RETURN      F5=RFIND      F6=RCHANGE
F7=UP        F8=DOWN       F9=SWAP     F10=LEFT       F11=RIGHT     F12=RETRIEVE
    
```

Figure 145. Entry type selections for Product Automation Panel

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 OS/390 but not required for automation of other products or applications.

This entry type is described in more detail in the subsequent section.

Product Automation Policy Object

DB2 System Defaults

Figure 146 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.

```
ACTIONS  HELP
-----
AOFGEPOL          Policy Selection
Command ===>                                SCROLL==> PAGE

Entry Type : DB2 System Defaults  PolicyDB Name : DATABASE_NAME
Entry Name : DB2_DEFAULT1        Enterprise Name : YOUR_ENTERPRISE

Action   Policy Name      Policy Description
-----
         DESCRIPTION   Enter description
         DB2 SYSTEM DEFAULTS Define DB2 System defaults
         RESOURCE THRESHOLDS Define DB2 Resource thresholds
-----
         WHERE USED     List Systems linked to this entry
         COPY           Copy data from an existing entry
***** Bottom of data *****

F1=HELP   F2=SPLIT   F3=END     F4=RETURN   F5=RFIND   F6=RCHANGE
F7=UP     F8=DOWN    F9=SWAP    F10=LEFT    F11=RIGHT  F12=RETRIEVE
```

Figure 146. Policy Selection Panel for DB2 System Defaults

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 115 on page 191 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 113 on page 189 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” on page 231.

(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 OS/390.

(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 147 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.

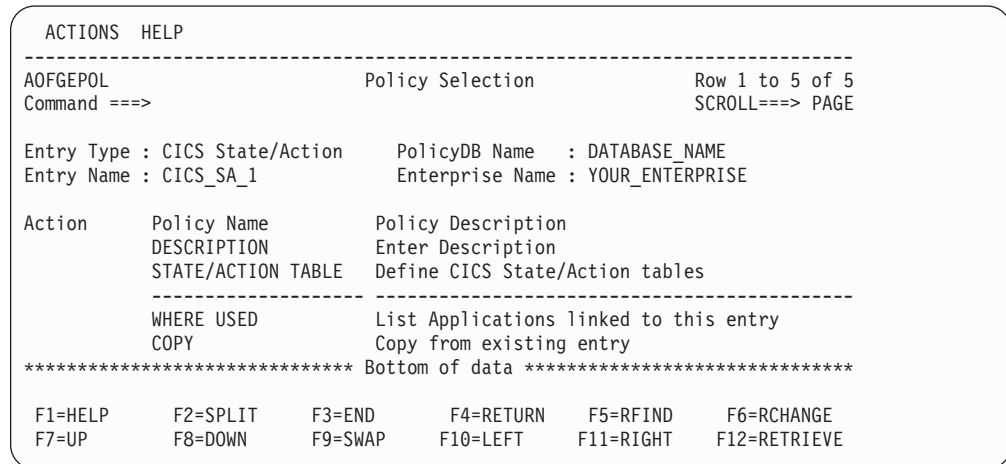


Figure 147. Policy Selection Panel for CICS State/Action Tables

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 147. 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 185.

Product Automation Policy Object

```

COMMANDS  HELP
-----
AOFPCSA          CICS State/Action tables
Command ==>>>

Entry Type : CICS State/Action      PolicyDB Name : DATABASE_NAME
Entry Name : CICS_SA_1              Enterprise Name : YOUR_ENTERPRISE

State action name: CICS_SA_1

Enter or update the following fields:

Short on storage . . _____
Storage violation. . _____
Autoinstall problem. _____
VTAM ACB problem . . _____

F1=HELP   F2=SPLIT   F3=END     F4=RETURN   F5=RFIND   F6=RCHANGE
F7=UP     F8=DOWN     F9=SWAP    F10=LEFT    F11=RIGHT  F12=RETRIEVE
  
```

Figure 148. CICS State/Action tables Panel

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 149 shows the policy items that are displayed when you create or modify a link definition for CICS from the related *Entry Name Selection* panel.

```

ACTIONS  HELP
-----
AOFGEPOL          Policy Selection          CICS link created
Command ==>>>          SCROLL==>> PAGE

Entry Type : CICS link      PolicyDB Name : DATABASE_NAME
Entry Name : LINK_DEF1     Enterprise Name : YOUR_ENTERPRISE

Action   Policy Name      Policy Description
DESCRIPTION
Enter description
CICS CONNECTION      CICS Link monitoring definitions
MONITORING PERIOD    Select Link monitoring period for connection
-----
WHERE USED          List APLs using this entry
COPY               Copy from existing entry
***** Bottom of data *****

F1=HELP   F2=SPLIT   F3=END     F4=RETURN   F5=RFIND   F6=RCHANGE
F7=UP     F8=DOWN     F9=SWAP    F10=LEFT    F11=RIGHT  F12=RETRIEVE
  
```

Figure 149. Policy Selection Panel for CICS Link Definitions

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 OS/390 CICS Automation. For more information on how to do this, see "More About Policy Item CICS CONNECTION". Also refer to *System Automation for OS/390 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 OS/390 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 OS/390 brings up the following panel where you describe the inter-region and inter-system communication (IRC and ISC) links so that they can be monitored by SA OS/390 CICS Automation.

```

COMMANDS  HELP
-----
AOFPCCN          CICS connection links
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 _____ (hh:mm)
Max repair tries . . . . 3 _____ (0-9)
Release delay . . . . . 05 _____ (4-99)
Acquire delay . . . . . 05 _____ (4-99)
Critical link (SDF) . . . NO YES NO
-----
F1=HELP    F2=SPLIT    F3=END     F4=RETURN  F5=RFIND   F6=RCHANGE
F7=UP      F8=DOWN     F9=SWAP   F10=LEFT   F11=RIGHT  F12=RETRIEVE
    
```

Figure 150. CICS connection links Panel

Fill in the following fields:

Connection id

The 4 character symbolic name by which the 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:

Product Automation Policy Object

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.

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 00:05.

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 151 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.

```

ACTIONS  HELP
-----
AOFGEPOL          Policy Selection          Row 1 to 5 of 5
Command ==>>>          SCROLL==>>> PAGE

Entry Type : Monitoring period      PolicyDB Name : DATABASE_NAME
Entry Name : MON_PERIOD_1          Enterprise Name : YOUR_ENTERPRISE

Action      Policy Name      Policy Description
           DESCRIPTION  Enter description
           MONITORING PERIOD  Define the monitoring period
-----
           WHERE USED      List APLs using this entry
           COPY            Copy from existing entry
***** Bottom of data *****

F1=HELP      F2=SPLIT      F3=END      F4=RETURN      F5=RFIND      F6=RCHANGE
F7=UP        F8=DOWN       F9=SWAP     F10=LEFT      F11=RIGHT     F12=RETRIEVE
  
```

Figure 151. 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 OS/390 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.

```

COMMANDS  HELP
-----
A0FGAPCP          Link monitoring service period
Command ==>>

Entry Type : Monitoring period      PolicyDB Name : DATABASE_NAME
Entry Name  : MON_PERIOD_1         Enterprise Name : YOUR_ENTERPRISE

Note: Review help panel for complete description of data entry fields.
      Valid start and stop times are 00:00-23:59, *****, and DOWN.

          Start/Stop  Start/Stop  Start/Stop  Start/Stop  Start/Stop
Monday  _____  _____  _____  _____  _____
Tuesday _____  _____  _____  _____  _____
Wednesday _____  _____  _____  _____  _____
Thursday _____  _____  _____  _____  _____
Friday   _____  _____  _____  _____  _____
Saturday _____  _____  _____  _____  _____
Sunday   _____  _____  _____  _____  _____
Weekday 08:00 17:00  _____  _____  _____
Weekend 09:00 14:00 _____  _____  _____
Daily   _____  _____  _____  _____  _____

F1=HELP   F2=SPLIT  F3=END    F4=RETURN  F5=RFIND  F6=RCHANGE
F7=UP     F8=DOWN   F9=SWAP  F10=LEFT  F11=RIGHT F12=RETRIEVE
  
```

Figure 152. Link monitoring service period Panel

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” on page 236.

(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 RSENAME to the IMS subsystems with which it is associated. This is described in more detail in “IMS XRF Resources” on page 238.

Product Automation Policy Object

These are described in more detail in the subsequent sections.

IMS State/Action Tables

Figure 153 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.

```
ACTIONS  HELP
-----
AOFGEPOL          Policy Selection
Command ==>>>                                SCROLL==>> PAGE

Entry Type : IMS State/Action      PolicyDB Name  : DATABASE_NAME
Entry Name  : IMS_SA_1             Enterprise Name : YOUR_ENTERPRISE

Action      Policy Name      Policy Description
DESCRIPTION Enter description
IMS STATE/ACTION Define IMS State/Action tables
-----
WHERE USED  List Applications linked to this entry
COPY       Copy data from existing entry
***** Bottom of data *****

F1=HELP    F2=SPLIT  F3=END    F4=RETURN  F5=RFIND  F6=RCHANGE
F7=UP      F8=DOWN   F9=SWAP   F10=LEFT   F11=RIGHT  F12=RETRIEVE
```

Figure 153. Policy Selection Panel for IMS State/Action Table

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 153. 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 OS/390 IMS Automation Programmer’s Reference and Operator’s Guide*.

```

COMMANDS  HELP
-----
AOFPIISA          IMS state action tables
Command ==>>>

Entry Type : IMS State/Action      PolicyDB Name : DATABASE_NAME
Entry Name : IMS_SA_1              Enterprise Name : YOUR_ENTERPRISE

State action name: IMS_SA_1

Enter or update the following table names:

MSC link recovery. . _____
OLDS error recovery. _____
MPP/BMP recovery . . _____

F1=HELP    F2=SPLIT    F3=END     F4=RETURN   F5=RFIND    F6=RCHANGE
F7=UP      F8=DOWN     F9=SWAP    F10=LEFT    F11=RIGHT   F12=RETRIEVE
    
```

Figure 154. IMS state action tables Panel

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 155 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.

```

ACTIONS  HELP
-----
AOFGEPOL          Policy Selection          Row 1 to 5 of 5
Command ==>>>          SCROLL==>> PAGE

Entry Type : IMS Status file      PolicyDB Name : DATABASE_NAME
Entry Name : IMS_XRF_SF1          Enterprise Name : YOUR_ENTERPRISE

Action    Policy Name      Policy Description
          DESCRIPTION    Enter description
          IMS STATUS FILE Define IMS status file names
-----
          WHERE USED     List Systems linked to this entry
          COPY           Copy data from an existing entry
***** Bottom of data *****

F1=HELP    F2=SPLIT    F3=END     F4=RETURN   F5=RFIND    F6=RCHANGE
F7=UP      F8=DOWN     F9=SWAP    F10=LEFT    F11=RIGHT   F12=RETRIEVE
    
```

Figure 155. Policy Selection Panel for IMS XRF Status File

The unique policy item for IMS XRF status files is:

Product Automation Policy Object

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 155 on page 237.

```
COMMANDS  HELP
-----
AOFPI SF          IMS Status file specification
Command ==>>>

Entry Type : IMS Status file      PolicyDB Name  : DATABASE_NAME
Entry Name  : IMS_XRF_SF1         Enterprise Name : YOUR_ENTERPRISE

IMS Status file entry name: IMS_XRF_SF1

Enter or update the following fields:

Partner domain name. _____
Partner status file.  _____

F1=HELP   F2=SPLIT   F3=END     F4=RETURN   F5=RFIND   F6=RCHANGE
F7=UP     F8=DOWN    F9=SWAP    F10=LEFT    F11=RIGHT  F12=RETRIEVE
```

Figure 156. IMS Status file specification Panel

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.

Fill in the following fields:

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 OS/390 initialization.

For more information about the status file specification refer to the description of the STSFILE keyword in *System Automation for OS/390 IMS Automation Programmer’s Reference and Operator’s Guide*.

IMS XRF Resources

Figure 157 on page 239 shows the policy items that are displayed when you select an IMS XRF resource from the related *Entry Name Selection* panel.


```

ACTIONS  HELP
-----
AOFGEPOL                                Policy Selection                                Row 1 to 5 of 5
Command ==>>>                                SCROLL==>> PAGE

Entry Type : IMS resource name      PolicyDB Name : DATABASE_NAME
Entry Name : IMS_XRF_1              Enterprise Name : YOUR_ENTERPRISE

Action      Policy Name      Policy Description
            DESCRIPTION    Enter description
            IMS RESOURCE NAMES Define IMS resource names
-----
            WHERE USED      List Systems linked to this entry
            COPY            Copy data from an existing entry
***** Bottom of data *****

F1=HELP    F2=SPLIT    F3=END      F4=RETURN    F5=RFIND    F6=RCHANGE
F7=UP      F8=DOWN     F9=SWAP     F10=LEFT     F11=RIGHT   F12=RETRIEVE

```

Figure 157. 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 157. 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 OS/390 IMS Automation Programmer’s Reference and Operator’s Guide* for the description of the IMSRESNM keyword.

```

COMMANDS  HELP
-----
AOFPIRN                                IMS Resource names
Command ==>>>

Entry Type : IMS resource name      PolicyDB Name : DATABASE_NAME
Entry Name : IMS_XRF_1              Enterprise Name : YOUR_ENTERPRISE

IMS resource name: IMS_XRF_1

Enter or update the following table fields:

XRF Resource ID. . . . _____
XRF partner name1. . . _____
XRF partner name2. . . _____

F1=HELP    F2=SPLIT    F3=END      F4=RETURN    F5=RFIND    F6=RCHANGE
F7=UP      F8=DOWN     F9=SWAP     F10=LEFT     F11=RIGHT   F12=RETRIEVE

```

Figure 158. IMS Resource names Panel

Product Automation Policy Object

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 OS/390 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.

OPC System Details

Figure 159 on page 241 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.

```

ACTIONS  HELP
-----
AOFGEPOL                Policy Selection                Row 1 to 5 of 5
Command ==>>>                SCROLL==>> PAGE

Entry Type : OPC System details      PolicyDB Name : DATABASE_NAME
Entry Name : OPC_SYS_DET1            Enterprise Name : YOUR_ENTERPRISE

Action      Policy Name      Policy Description
           DESCRIPTION  Enter description
           OPCA0 ENVIRON Define OPCA0 environ entries
-----
           WHERE USED      List Systems linked to this entry
           COPY            Copy data from an existing entry
***** Bottom of data *****

F1=HELP      F2=SPLIT      F3=END      F4=RETURN      F5=RFIND      F6=RCHANGE
F7=UP        F8=DOWN       F9=SWAP     F10=LEFT       F11=RIGHT     F12=RETRIEVE

```

Figure 159. Policy Selection Panel for OPC System Detail

The unique policy item for OPC system details is:

OPCA0 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 OS/390 displays the *OPC System details* panel which is described in more detail in “More About Policy Item OPCA0 ENVIRON”.

More About Policy Item OPCA0 ENVIRON

The OPCA0 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 OS/390 displays the panel shown in Figure 160.

```

COMMANDS  HELP
-----
AOFPOEN                OPC System details
Command ==>>>

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)

F1=HELP      F2=SPLIT      F3=END      F4=RETURN      F5=RFIND      F6=RCHANGE
F7=UP        F8=DOWN       F9=SWAP     F10=LEFT       F11=RIGHT     F12=RETRIEVE

```

Figure 160. 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

Product Automation Policy Object

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 OS/390 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 OS/390 OPC Automation Programmer's Reference and Operator's Guide*.

Controller Details

Figure 161 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.

```

ACTIONS  HELP
-----
AOFGEPOL                Policy Selection
Command ==>>>                SCROLL==>> PAGE

Entry Type : Controller details   PolicyDB Name : DATABASE_NAME
Entry Name : OPC_OCS_1           Enterprise Name : YOUR_ENTERPRISE

Action   Policy Name      Policy Description
-----
         DESCRIPTION
         OPCA PCS        Define OPCA PCS entries
         SPECIAL RESOURCES Select Special Resources for this controller
         -----
         WHERE USED      List Systems linked to this entry
         COPY            Copy data from an existing entry
         ***** Bottom of data *****

F1=HELP   F2=SPLIT   F3=END    F4=RETURN  F5=RFIND   F6=RCHANGE
F7=UP     F8=DOWN    F9=SWAP   F10=LEFT   F11=RIGHT  F12=RETRIEVE

```

Figure 161. Policy Selection Panel for OPC Controller Details

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 243.

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 162 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.

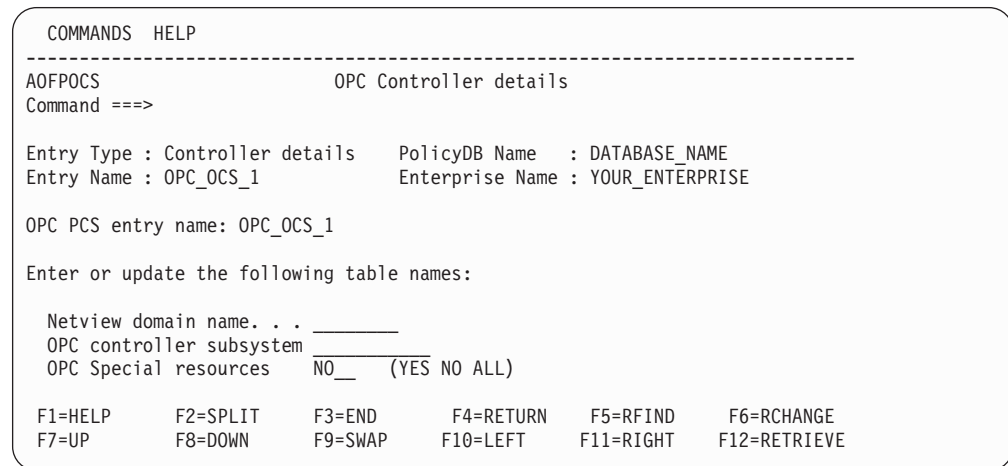


Figure 162. OPC Controller details Panel

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 OS/390 subsystem name of the OPC Controller as defined in the IEFSSNxx member. Note: When defining the OPC Controller to SA OS/390, 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 OS/390 monitored subsystem specified in the *OPC special resource definitions* panel (see Figure 164 on page 245). 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

Product Automation Policy Object

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 OS/390 Monitored subsystems. No Special Resource entries are required.

OPC Special Resources

Figure 163 shows the policy items that are displayed when you select an OPC special resource from the related *Entry Name Selection* panel.

```

ACTIONS  HELP
-----
OFGEPOL          Policy Selection
ommand ==>>>                                SCROLL==>>> PAGE

ntry Type : Special resources      PolicyDB Name : DATABASE_NAME
ntry Name  : OPC_RES1              Enterprise Name : YOUR_ENTERPRISE

ction         Policy Name          Policy Description
DESCRIPTION  Enter description
SPECIAL RESOURCE Define OPC special resources
-----
WHERE USED   List Controllers linked to this entry
COPY        Copy data from an existing entry
***** Bottom of data *****

F1=HELP      F2=SPLIT      F3=END        F4=RETURN     F5=RFIND      F6=RCHANGE
F7=UP        F8=DOWN        F9=SWAP       F10=LEFT      F11=RIGHT     F12=RETRIEVE
```

Figure 163. Policy Selection Panel for OPC Special Resources

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 OS/390 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 OS/390 is able to globally control the creation and setting of OPC Special resources based on the status of SA OS/390 monitored subsystems. This allows OPC to resolve job scheduling dependencies based on the status of SA OS/390 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 240).

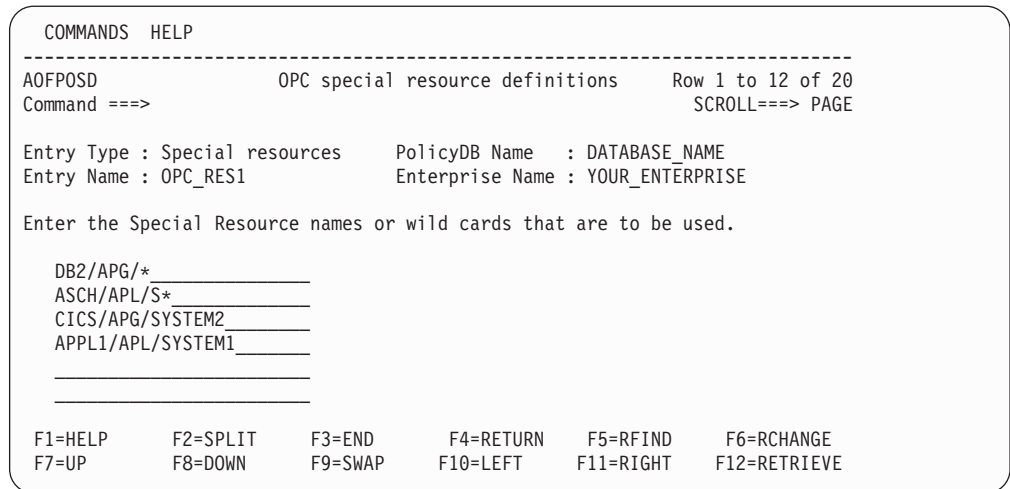


Figure 164. OPC special resource definitions Panel

Fill in your special resource names with up to 24 characters. Wildcards and patterns may be entered.

Workstation Domains

Figure 165 shows the policy items that are displayed when you select an OPC workstation domain from the related *Entry Name Selection* panel.

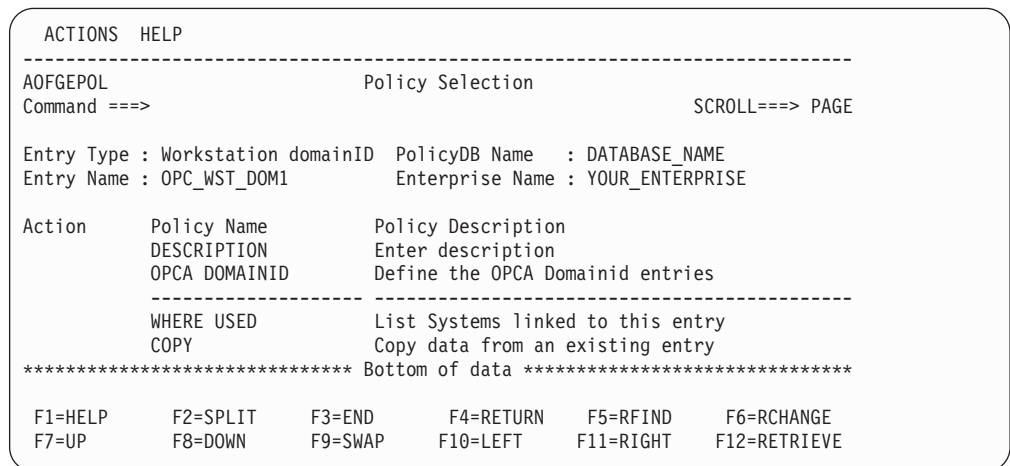


Figure 165. Policy Selection Panel for OPC Workstation Domain

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 OS/390 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 93 on page 163. The *Message Processing* panel for OPC shows minor differences in the top of the panel compared to the one shown in Figure 93 on page 163. These are shown here:

Product Automation Policy Object

Entry Type : Workstation domainID PolicyDB Name : DATABASE_NAME
Entry Name : OPC_WST_DOM1 Enterprise Name : YOUR_ENTÉRPRISE

Message type: OPCA_DOMAINID

For more information refer to *System Automation for OS/390 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 166 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.

ACTIONS		HELP

AOFGEPOL	Policy Selection	
Command ==>	SCROLL==> HALF	
Entry Type :	Timers	PolicyDB Name : DATABASE_NAME
Entry Name :	TIMER_NAME	Enterprise Name : YOUR_ENTÉRPRISE
Action	Policy Name	Policy Description
	DESCRIPTION	Enter description
	TIMERS	Define timers for automation

	WHERE USED	List Systems linked to this entry
	COPY	Copy data from an existing entry

Figure 166. Policy Selection Panel for Timers

The unique policy item for Timers is:

TIMERS

You define a timer in the *Timer specification* panel shown in Figure 167 on page 247.

SA OS/390 displays the following panel if you select the TIMERS policy item from the **Policy Selection** panel for *Timers*.


```

COMMANDS  ACTIONS  HELP
-----
A0FPIT10          Timer specification
Command ==>>>                                SCROLL==>> PAGE

Entry Type : Timers          PolicyDB Name : DATABASE_NAME
Entry Name : TIMER_NAME     Enterprise Name : YOUR_ENTERPRISE

Enter information for commands to be executed at scheduled times.

Action  Timer-id      Execution Time      Issuer  Save  Catchup
Command text
-----
MVS W   WRITELOG      Every day at 23:59:00  RECOPER  YES  NO
-----
TSTMSG  CATCHUP9    At 09:00:00          AUTST1   YES  YES
CATCHUP TIMER EXPIRED AT 09:00:00
-----

```

Figure 167. Timer specification Panel

A timer definition consists of:

Timer ID

When a user timer is defined with the SAVE-option outside the control of SA OS/390, 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 OS/390 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:

- A NetView operator ID, in which case SA OS/390 issues the timer-driven command only if that operator is logged on.
- The primary programmed operator interface task (PPT), in which case SA OS/390 issues the timer-driven command even when an operator is not logged on.
- An automation operator name.

Save Whether SA OS/390 should save the timed event in the NetView SAVE/RESTORE database. You can enter one of the following values:

- YES
- NO
- GMT
- LOCAL

Catchup

Whether a saved timer should be caught up after a system outage.

Command

The actual timer-driven command or automation procedure SA OS/390 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 OS/390 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 168 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.

ACTIONS		HELP

AOFGEPOL	Policy Selection	
Command ==>		SCROLL==> HALF
Entry Type :	Timeout Settings	PolicyDB Name : DATABASE_NAME
Entry Name :	TIMEOUT_NAME	Enterprise Name : YOUR_ENTERPRISE
Action	Policy Name	Policy Description
	DESCRIPTION	Enter description
	TIMEOUT VALUES	Define timeout values

	WHERE USED	List Systems linked to this entry
	COPY	Copy data from an existing entry

Figure 168. Policy Selection Panel for Timeout Settings

The unique policy item for *Timeout Settings* is:

TIMEOUT VALUES

SA OS/390 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 169 on page 249 shows the definition panel for a Timeout Settings.

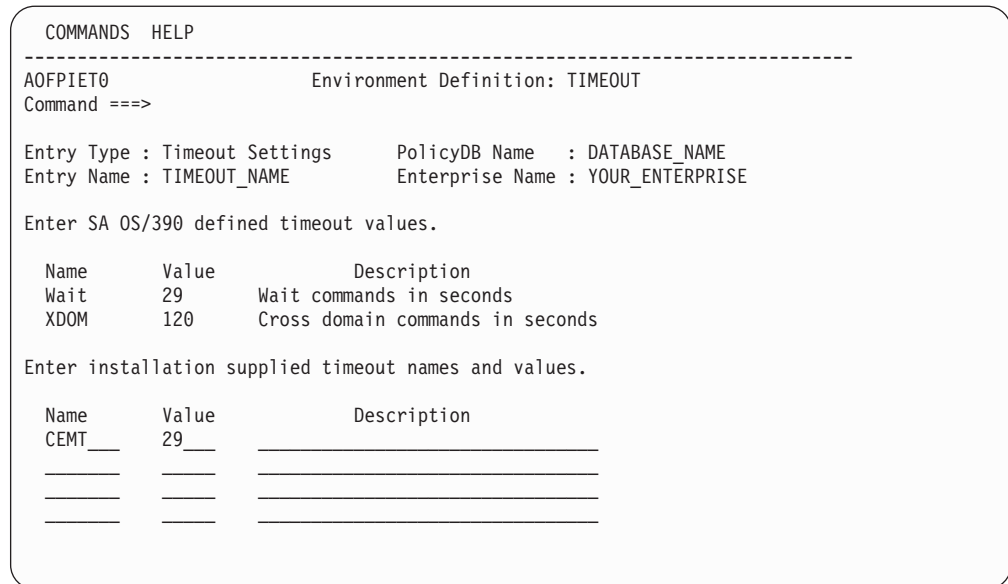


Figure 169. Environment Definition: TIMEOUT Panel

The following timeout values can be entered:

Wait Defines how long automation processing waits for a response after issuing a system, NetView, or SA OS/390 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 OS/390 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.

Note: You must ensure that the names of any global variables you create do not clash with SA OS/390 external or internal global variable names. SA OS/390 external global variables are documented in *System Automation for OS/390 Customizing and Programming*. In addition, you should not use names beginning with EHK, AOF, SUBS, ING or ISQ.

Timeout Settings Policy Object

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 OS/390 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 OS/390 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 OS/390 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 jobname requiring the response to the IEF238D message. You can specify a trailing asterisk (*) to allow for generic jobname 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 OS/390 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 MAT 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 OS/390 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 170 on page 251 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.

Tape Attendance Policy Object

```

ACTIONS  HELP
-----
AOFGEPOL          Policy Selection
Command ==>>>                                SCROLL==>> HALF

Entry Type : Tape Attendance      PolicyDB Name : DATABASE_NAME
Entry Name : TAPE_ATTEND_NAME    Enterprise Name : YOUR_ENTERPRISE

Action   Policy Name      Policy Description
        DESCRIPTION      Enter description
        ATTEND TIME      Define tape drive attend time
-----
        WHERE USED      List Systems linked to this entry
        COPY            Copy data from an existing entry
  
```

Figure 170. Policy Selection Panel for Tape Attendance

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 171 for more information on this policy item.

Figure 171 shows the *Environment Definition: TAPES* panel.

```

COMMANDS  HELP
-----
AOFPIETA          Environment Definition: TAPES
Command ==>>>                                SCROLL==>> PAGE

Entry Type : Tape Attendance      PolicyDB Name : DATABASE_NAME
Entry Name : TAPE_ATTEND_NAME    Enterprise Name : YOUR_ENTERPRISE

Indicate if tapes are attended. If attended indicate if GMFHS monitoring of
tape mounts is required, and the warning and alert delay times.

Tapes Attended ==> YES Yes or No      Monitor Tape Mounts ==> YES Yes or No

Warning Delay ==> 01:00 (MM:SS)      Alert Delay ==> 02:00 (MM:SS)

Day   ----- Unattended Times -----
*     Begin  End    Begin  End    Begin  End    Begin  End
-----
_     _      _      _      _      _      _      _      _
_     _      _      _      _      _      _      _      _
_     _      _      _      _      _      _      _      _
_     _      _      _      _      _      _      _      _
  
```

Figure 171. 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 OS/390 workstation. If you enter YES in this field, SA OS/390 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 OS/390 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
SSSS.VVVVVV_DDD(D)

where:

SSSS

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 OS/390.

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 OS/390 data sets, resources and facilities. These OS/390 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 (OS/390 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 172 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.

ACTIONS		HELP

AOFGEPOL		Policy Selection
Command ==>>		SCROLL==>> PAGE
Entry Type : MVS Component		PolicyDB Name : TEST_PDB
Entry Name : MVS_COMPONENT		Enterprise Name : ALL_PRIMARY
Action	Policy Name	Policy Description
	DESCRIPTION	Enter description
	MVSDUMP THRESHOLDS	Define MVS dump specifications and thresholds
	SMFDUMP THRESHOLDS	Define SMF dump specification and thresholds
	LOGREC THRESHOLDS	Define LOGREC specification and thresholds
	SYSLOG THRESHOLDS	Define SYSLOG specification and thresholds
	WTOBUF RECOVERY	Define WTOBUF recovery specifications
	MESSAGES/USER DATA	Define MVS messages and user data
	MINOR RESOURCE FLAGS	Define MVSESA sub-component flags

	WHERE USED	List Systems linked to this entry
	COPY	Copy data from an existing entry

Figure 172. Policy Selection Panel for MVS Component

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 254.

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 254.

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 254.

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 254.

WTOBUF RECOVERY

This option allows you to customize recovery from write-to-operator (WTO) buffer shortage and full conditions at an operator console.

MVS Component Policy Object

Note: When SA OS/390 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 173 on page 255 for more information on this policy item.

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 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 258.

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 259. For information on the relation of minor resources and automation flags refer to “Automation Flags and Minor Resource Flags” on page 47.

More About the THRESHOLD Policy Items

For various threshold policy items, SA OS/390 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 41.

More About the WTOBUF RECOVERY Policy Item

Figure 173 shows the panel that is displayed when the WTOBUF RECOVERY policy item is selected.

```

COMMANDS  HELP
-----
AOFPIEW0          Environment Definition: WTOBUF Overflow
Selection ==>

Entry Type : MVS Component          PolicyDB Name  : DATABASE_NAME
Entry Name  : MVS_COMP_NAME        Enterprise Name : YOUR_ENTERPRISE

Enter information and selection to restore console after WTOBUF overflow.

NBUF ==>          Threshold to initiate recovery of console
Retry ==>         Delay time between initiating recovery passes (hh:mm:ss)

----- Recovery Phase Selections -----
1 RECOVERY        Console recovery passes (R)
2 NORMCONS        Normal console mode after recovery (N)

```

Figure 173. Environment Definition: WTOBUF Overflow Panel

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 specifies the time between each phase of the recovery process. The range is from 00:00:00 to 23:59:59. After performing a set of recovery actions, the recovery process waits the specified time period before trying the next set of recovery actions. The default is 1.

RECOVERY

This option specifies recovery operations. A panel similar to Figure 174 on page 256 is displayed.

MVS Component Policy Object

```
COMMANDS  HELP
-----
AOFPIEW1          WTOBUF Recovery Passes          SCROLL==> PAGE
Command ==>

Entry Type : MVS Component          PolicyDB Name : DATABASE_NAME
Entry Name : MVS_COMPONENTS        Enterprise Name : YOUR_ENTERPRISE

Enter commands to be executed when a WTO buffer shortage is detected.
Commands may contain the variables &EHKVAR1 and &EHKVAR2.
&EHKVAR1 = the Console ID in trouble  &EHKVAR2 = CUU of console in trouble

Pass
Command Text
1
MVS K S,RTME=001,DEL=RD,L=&EHKVAR1_____

1
MVS K E,D,L=&EHKVAR1_____

1
MVS K A,NONE,L=&EHKVAR1_____

2
MVS K S,RTME=001,DEL=R,L=&EHKVAR1_____
```

Figure 174. Policy Selection Panel for MVS Component

where:

- 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 K A,NONE,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.

MVS Component Policy Object

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 175 is displayed.

When a WTOBUF shortage has been relieved by SA OS/390, 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:

```
COMMANDS  HELP
-----
AOFPIEW2      Console Mode After WTOBUF Recovery
Command ==>                                     SCROLL==> PAGE

Entry Type : MVS Component      PolicyDB Name  : DATABASE_NAME
Entry Name  : MVS_COMPONENT_NAME Enterprise Name : YOUR_ENTERPRISE

Enter parameters to reset console after WTO buffer shortage is relieved.

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 OS/390

Console Areas:  A   B   C   D   E   F   G   Total Area Allocated
Size           ==>                                     0

Additional Commands to Restore Normal Console Operation.
_____
_____
_____
_____
```

Figure 175. Console Mode After WTOBUF Recovery Panel

The following fields can be specified:

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.

MVS Component Policy Object

- ROLL- 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.

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 172 on page 253), you come to the *Message Processing* panel shown in Figure 176.

```
COMMANDS  ACTIONS  HELP
-----
AOFPISSM          Message Processing
Command ==>>>          SCROLL==>>> PAGE

Entry Type : MVS Component      PolicyDB Name  : TEST_PDB
Entry Name  : MVS_COMPONENT     Enterprise Name: ALL_PRIMARY

Subsystem   : MVSESA

Enter messages issued by this resource that will result in automated actions.
Actions:  CMD = Command  REP = Reply  CODE = CODE  USER = User defined values

Action  Message ID          Cmd  Rep  Code  User
        Description
-----  -----
        MVSDUMPTAKEN          1
        Dump is taken
        MVSDUMPFULL          1
        Dump is full
        SYSLOG
        LOGREC
```

Figure 176. Message Processing Panel

MVS Component Policy Object

The policy item MESSAGES/USER DATA allows you to specify message IDs, also the special ones like MVSDUMPTAKEN, MVSDUMPFULL, SMFDUMP, SYSLOG, LOGREC. It is possible to define up to 99 messages for each message ID.

On this panel, you can define how SA OS/390 should react to an issued message by typing one of the four alternatives C(md), R(ep), C(ode), U(ser) into the *Action* column. 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 163.

The SA OS/390 sample data bases (see “Using Sample SA OS/390 Policy Databases” on page 20) provide an example of how to automate AMRF buffer recovery using the MESSAGES/USER DATA policy item.

AMRF buffer shortages generate three OS/390 system messages that are trapped by SA OS/390 using the message table sample AOFMMVS. These three OS/390 system messages, IEA359E AMRFSHORT, IEA360A AMRFFULL and IEA361I AMRFCLEAR cause module AOFRSA0G to be driven. AOFRSA0G scans the automation control file (ACF) for MVSESA AMRFSHORT, MVSESA AMRFFULL and MVSESA AMRFCLEAR policy definitions and executes any user supplied commands or command lists specified for these message definitions.

The SA OS/390 sample policy databases include an entry in MESSAGES/USER DATA for AMRFFULL and AMRFCLEAR. AMRFFULL causes the OS/390 command CONTROL M,AMRF=N to be issued to clear the AMRF buffers when they reach 100% and AMRFCLEAR causes CONTROL M,AMRF=Y to be issued to reactivate AMRF after the buffer shortage has been relieved. AMRFSHORT has not been included in the sample. If required, you must add and tailor it.

More About Policy Item MINOR RESOURCE FLAGS

The *Minor Resource Selection* panel appears if you select policy item *MINOR RESOURCE FLAGS* from the *Policy Selection* panel for entry types *Application* or *MVS Component*.

MVS Component Policy Object

```
COMMANDS  ACTIONS  HELP
-----
AOFPIR4          Minor Resource Selection
Command ==>>>          SCROLL==>> PAGE

Entry Type : MVS Component      PolicyDB Name : DATABASE_NAME
Entry Name : MVS_COMPONENTS    Enterprise Name : YOUR_ENTERPRISE

Major Resource: MVSESA

Select minor resource to be altered.

Action Minor Resource
- LOGREC _____
- MVSDUMP _____
- SCU0040 _____
- SCU0050 _____
- SCU0060 _____
- SCU0070 _____
- SCU0080 _____
- SCU0090 _____
- SMFDUMP _____
- SYSLOG _____
- TAPES _____
- WTOBUF _____

F1=HELP   F2=SPLIT   F3=END     F4=RETURN  F5=RFIND   F6=RCHANGE
F7=UP     F8=DOWN    F9=SWAP    F10=LEFT   F11=RIGHT  F12=RETRIEVE
```

Figure 177. Minor Resource Selection Panel

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 upcoming *Flag Automation Specification* panel.

You can also define a hierarchy of minor resources. For example, for the minor resource TAPES from our example in Figure 177, 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 _____
- _____
```

MVSCOMP Defaults Policy Object

MVSCOMP Defaults allow you to set default values for MVS Component sets. Figure 178 on page 261 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.

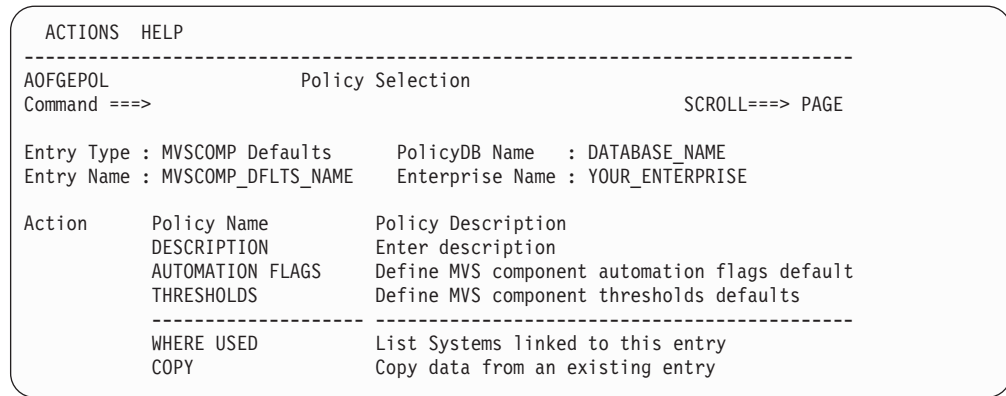


Figure 178. Policy Selection Panel for MVSCOMP Defaults

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 42 for more information on 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 OS/390 Uses Error Thresholds” on page 39 for more information on error thresholds.

Only one MVSCOMP Default may be attached to a system.

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 179 on page 262 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.

System Defaults Policy Object

ACTIONS		HELP

AOFGEPOL		Policy Selection
Command ==>		SCROLL==> HALF
Entry Type :	System Defaults	PolicyDB Name : DATABASE_NAME
Entry Name :	SYS_DEFAULTS_NAME	Enterprise Name : YOUR_ENTERPRISE
Action	Policy Name	Policy Description
	DESCRIPTION	Enter description
	AUTOMATION FLAGS	Define System automation flag defaults
	THRESHOLDS	Define System thresholds defaults
	ENVIRONMENT	Define System environment defaults

	WHERE USED	List Systems linked to this entry
	COPY	Copy data from an existing entry

Figure 179. Policy Selection Panel for System Defaults

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 42 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 OS/390 Uses Error Thresholds” on page 39 for more information on error thresholds.

ENVIRONMENT

This option allows you to set various defaults for applications you define to the system. The values entered in the upcoming *Subsystem Defaults* panel are used as default values for *Application* objects which are attached to this system and do not have this information specified.

Only one *System Defaults* set may be attached to a system.

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 180 on page 263 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.

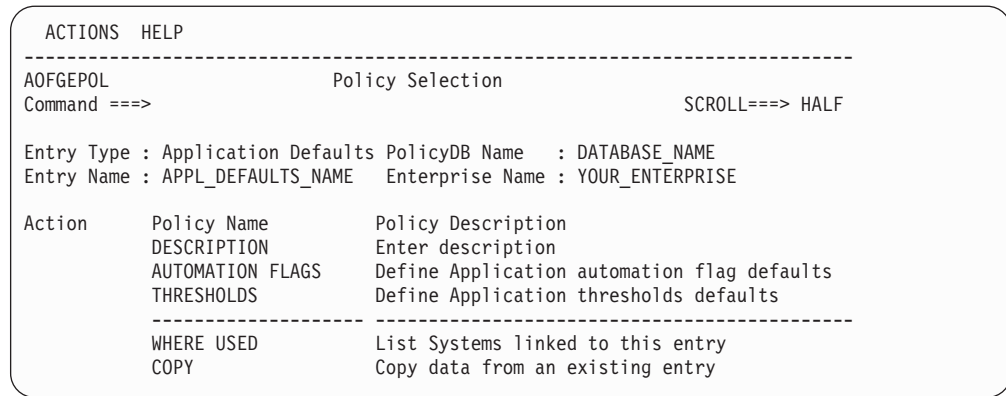


Figure 180. Policy Selection Panel for Application Defaults

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 42 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 OS/390 Uses Error Thresholds” on page 39 for more information on setting thresholds.

Only one *Application Defaults* set may be attached to a system.

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 181 on page 264 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.

Auto Operators Policy Object

```

ACTIONS  HELP
-----
AOFGEPOL          Policy Selection
Command ==>>>          SCROLL==>>> HALF

Entry Type : Auto Operators      PolicyDB Name : DATABASE_NAME
Entry Name : AUTO_OPERATORS_NAME Enterprise Name : YOUR_ENTERPRISE

Action   Policy Name      Policy Description
        DESCRIPTION
        OPERATORS        Enter description
                          Define automation operators
-----
        WHERE USED      List Systems linked to this entry
        COPY            Copy data from an existing entry

```

Figure 181. Policy Selection Panel for Auto Operators

The unique policy item for *Auto Operators* is:

OPERATORS

Automation operators are crucial to the operation of SA OS/390. The SA OS/390 sample automation control file supplies definitions that are required in order to start SA OS/390. 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 OS/390 defines an *automated function name* for each automation operator it uses. Certain automated function names must be defined. These are listed in Table 11. The automated function names are included in the sample policy databases supplied with SA OS/390. 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 11 lists the automated function names that need to be defined.

Table 11. SA OS/390 Mapping of NetView Automation Operator to SA OS/390 Automated Function

Automated Function	Automation Operator	Description
AUTO1	AUTO1	Used for SA OS/390 initialization
AUTO2	AUTO2	Used for SA OS/390 initialization
BASEOPER	AUTBASE	Base automation: Backs up all other automation operators.
SYSOPER	AUTSYS	System automation: Acts as the primary system automation operator.
LOGOPER	AUTLOG	Used for message writing and for unassigned messages.
MSGOPER	AUTMSG	Message automation: Formats and routes messages.
AOFWRKxx	AUTWRKxx	Used for subsystem processing, for example startup, shutdown
NETOPER	AUTNET1, AUTNET2	VTAM and NetView automation: Acts as the network products automation operator.

Table 11. SA OS/390 Mapping of NetView Automation Operator to SA OS/390 Automated Function (continued)

Automated Function	Automation Operator	Description
JESOPER	AUTJES	JES automation: Acts as the JES2 or JES3 product automation operator.
MONOPER	AUTMON	Monitoring and startup automation: Performs active monitoring of resources.
RECOPER	AUTREC	Recovery automation: Acts as the primary recovery operator.
RPCOPER	AUTRPC	Used for XCF requests
SHUTOOPER	AUTSHUT	Shutdown automation: Acts as the base shutdown operator.
GSSOPER	AUTGSS	General automation. Acts as the operator for applications using SA OS/390 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.
GATOPER	GATAOF01	Gateway: Acts as the outbound gateway operator for automation notification message forwarding
MVSCONS <i>i</i>	AUTCON	Used for the MVS and NetView console interface. The MVS console with which it is associated can be changed in the customization dialog.
PPIOPER	AOFARCAT	ARM PPI receiver
HBOPER	AUTHB	Used by heartbeat function
XCFOPER	AUTXCF	Used for enhanced Parallel Sysplex automation functions.
XCFOPER2	AUTXCF2	Used for enhanced Parallel Sysplex automation functions.

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 OS/390. You may define additional automation operators, to suit your automation needs.

When defining SA OS/390 automated functions, you must associate the SA OS/390 automated function names with actual automation operator IDs defined in member DSIOPF in the NetView DSIPARM data set. For example, you can associate the SA OS/390 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 OS/390 Customizing and Programming* for more information on synonyms.

Auto Operators Policy Object

Figure 182 shows an example of an *Automation Operator Definition* panel.

```
COMMANDS  ACTIONS  HELP
-----
AOFPIA00          Automation Operator Definitions
Command ==>>>                                SCROLL==>>> PAGE

Entry Type : Auto Operators          PolicyDB Name : DATABASE_NAME
Entry Name : BASE_AUTOOPS            Enterprise Name : YOUR_ENTERPRISE

Actions: S = Select  M = Move  B = Before  A = After  I = Insert

Action      Automated
Function    Messages for this Operator (* notation ok)
BASEOPER
GSSOPER
JESOPER
MONOPER
MSGOPER     AOF*
NETOPER
RECOOPER    IEA*,IEE*,IOS*
SHUTOPER
SYSOOPER    IEE889I
LOGOPER     *
PPIOOPER
GATOPER
MVSCONSi
HBOPER
AOFWRK01
AOFWRK02
AOFWRK03
AOFWRK04
AOFWRK05
```

Figure 182. Automation Operator Definition Panel

Messages are entered in the *Messages for this Operator* field. You can specify:

- A single message, such as IEE115I
- 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 OS/390 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 OS/390 searches this list of automation operators from the top to find which automation operator is associated with the message. To make sure SA OS/390 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.

Auto Operators Policy Object

The following automated functions should not have messages assigned to them on the *Automation Operator Definition* panel:

- BASEOPER
- GATOPER
- MVSCONSi (where *i* is blank, A-Z, or 0-9)
- SHUTOPER
- HBOPER
- AOFWRKxx
- GSSOPER

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 183 provides an example.

```
COMMANDS  HELP
-----
AOFPIA01          Automation Operator NetView Userids
Command ==>>>

Entry Type : Auto Operators          PolicyDB Name : DATABASE_NAME
Entry Name : BASE_AUTOOPS           Enterprise Name : YOUR_ENTERPRISE

Automated Function: BASEOPER
Messages assigned:

Enter automation operators and &net; operator(s) to receive messages.

Automation Operators          NetView Operators
Primary ==> A&DOMAIN.xyz ____  ==> _____
Backup  ==> X&SYSPROC.______  ==> _____
                                     ==> _____
                                     ==> _____
                                     ==> _____

Primary ==> AUTJES              ==>
Backup  ==>                    ==>
                                     ==>
                                     ==>
                                     ==>
```

Figure 183. Automation Operator NetView Userids Panel

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

Auto Operators Policy Object

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 message 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 SYSCONE variables and the &DOMAIN. variable for the automation operators and NetView operators specification.

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 OS/390 displays a different panel if MVSCONS_i is entered on the *Automation Operator Definition* panel. Figure 184 provides an example.

```
COMMANDS  HELP
-----
AOFPIA03          Automation Operator: MVSCONS
Command ==>>>

Entry Type : Auto Operators      PolicyDB Name  : DATABASE_NAME
Entry Name  : AUTO_OPERATORS    Enterprise Name: YOUR_ENTERPRISE

Automated Function:

Enter information.

MVS Console ID   ==> ___      Console-id eligible to enter NetView commands
Automation Operator ==>

or

MVS Console NAME ==>          Console-name eligible to enter NetView cmds
Automation Operator ==>      Operator-id (optional)
```

Figure 184. Automation Operator: MVSCONS Panel

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 OS/390 detects attached to the machine. With the default ID of AUTCON and three consoles attached to the machine, autotasks AUTCON, AUTCON01 and AUTOCON02 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 OS/390 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, MVSCONS1..., MVSCONS9).

Structure of AOFWRKxx Operators

AOFWRKxx operators are used to distribute the work on different NetView tasks, so that they can run in parallel. AOFWRKxx operators are used as dedicated automated function responsible for all the work on a particular subsystem. However, 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 AOFWRKxx 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 AOFWRKxx operators.

Behavior of SHUTDOWN, STARTUP and GSSOPER

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

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

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

Auto Msg Classes Policy Object

The *Auto Msg Classes* policy object allows you to define your own message classes. Message classes defined here can be assigned to operators with the NOTIFY OPERATORS policy item of the system object. See "System Policy Object" on page 99 for information on how to assign message classes. Figure 185 on page 270 shows the policy items that are displayed when a new Auto Msg Class entry is created or when an existing Auto Msg Class entry is selected from the *Entry Name Selection* panel.

Auto Msg Classes Policy Object

ACTIONS		HELP

AOFGEPOL		Policy Selection
Command ==>		SCROLL==> HALF
Entry Type : Auto Msg Classes		PolicyDB Name : DATABASE_NAME
Entry Name : AUTO_MSG_CLASS_NAME		Enterprise Name : YOUR_ENTERPRISE
Action	Policy Name	Policy Description
	DESCRIPTION	Enter description
	CLASSES	Define message classes

	WHERE USED	List Systems linked to this entry
	COPY	Copy data from an existing entry

Figure 185. Policy Selection Panel for Auto Msg Classes

The unique policy item for Auto Msg Class sets is:

CLASSES

This option allows you to define 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 "System Policy Object" on page 99 for more information on the message classes that SA OS/390 defines. No duplicate message classes are allowed.

Network Policy Object

When multiple OS/390 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.
- Define Gateway sessions connecting systems.

A gateway session uses autotasks both inbound to and outbound from a system to send data between systems.

Using gateway sessions allows you to send messages, commands, and responses between systems without requiring individual NetView-NetView task (NNT) sessions for each logged-on operator. A gateway definition in your automation policy defines which system is being connected to and the logon and password used to log a gateway autotask on to the other system.

- Manage passwords for gateway autotasks.
- 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 OS/390 NetView Interface. If a TAF session is not defined to start automatically, you can initialize it using the OPER dialog.
- Define user NNT sessions and specify whether or not each of these should be initiated automatically when the operator logs on to the SA OS/390 NetView Interface. If an NNT session is not defined to start automatically you can initialize it using the OPER dialog.

“Chapter 8. Automating Networks” on page 327 provides more information on the definition of automation networks. Figure 186 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.

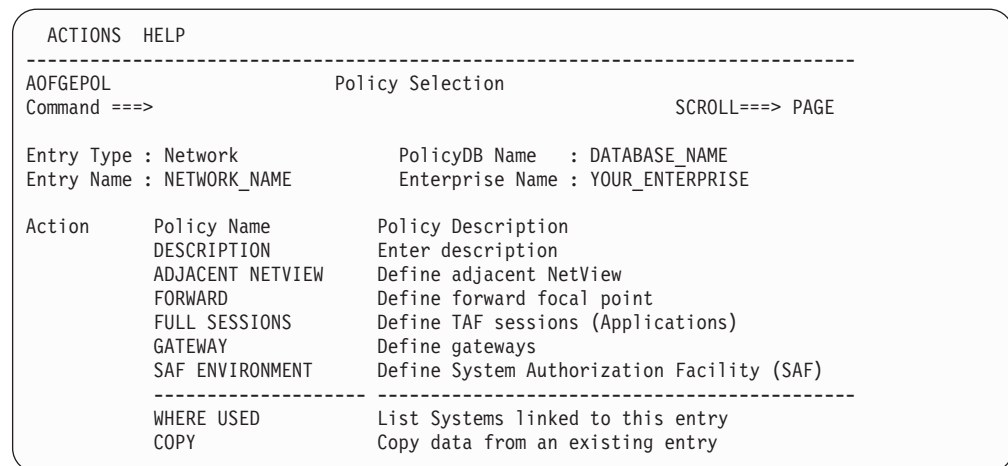


Figure 186. Policy Selection Panel for Network

The unique policy items for *Network* are:

ADJACENT NETVIEW

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:

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

Network Policy Object

messages to the current one without it 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

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:

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 RACFNNT can be entered if RACF is to be used to control passwords.

Logmode

The logmode that SA OS/390 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

When the System Authorization Facility (SAF) is identified as being used to manage gateway access, SA OS/390 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 OS/390 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 OS/390 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 OS/390 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 | (A-Z) |
| B - Alphabetic/National | (A-Z,@,#,\$) |
| C - Consonant (Alphabetic - No Vowels) | |
| E - Even Numerics | (0,2,4,6,8) |
| N - Numeric only | (0-9) |
| O - Odd Numerics | (1,3,5,7,9) |
| V - Vowels only | (A,E,I,O,U) |
| W - Alphanumeric - No Vowels | |
| X - Alphanumeric | (A-Z,0-9) |
| Y - Alphanumeric/National | (A-Z,0-9,@,#,\$) |
| Z - Alphanumeric/National - No Vowels | |

The following mask specification would cause SA OS/390 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

This policy object is used to define definitions for NetView to NetView sessions. Each *NNT Sessions* set can contain multiple defined NNT sessions. Figure 187 on page 274 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.

NNT Sessions Policy Object

```

ACTIONS  HELP
-----
AOFGEPOL          Policy Selection
Command ==>>>          SCROLL==>>> HALF

Entry Type : NNT Sessions      PolicyDB Name : DATABASE_NAME
Entry Name : NNT_SESS_NAME     Enterprise Name : YOUR_ENTERPRISE

Action   Policy Name      Policy Description
        DESCRIPTION
        SESSIONS        Enter description
                          Define NNT sessions (NetView)
-----
        WHERE USED      List Systems linked to this entry
        COPY            Copy data from an existing entry

```

Figure 187. Policy Selection Panel for NNT Sessions

The unique policy item for NNT is:

SESSIONS

When you select this option, a panel similar to Figure 188 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.

```

COMMANDS  ACTIONS  HELP
-----
AOFGNNOPI          NNT Sessions Definition
Command ==>>>          Scroll ==>>> PAGE

Entry Type : NNT Sessions      PolicyDB Name : DATABASE_NAME
Entry Name : NNT_SESS_NAME     Enterprise Name : YOUR_ENTERPRISE

Enter the Userid used to establish NNT sessions.

Action  Userid  Description
        XXAOC05  SHIFT LEADER
        XXAOC06  SENIOR OPERATOR
        XXAOC07  OTHER SENIOR OPERATOR
        XXAOC43  CONTRACT OPERATOR

```

Figure 188. NNT Sessions Definition Panel

To complete the definition for each user ID, enter an S (Select) in the Action column. A panel similar to Figure 189 on page 275 is displayed.

```

COMMANDS  HELP
-----
AOFGNNDM          NNT Sessions Definition          Scroll ==> PAGE
Command ==>

Entry Type : NNT Sessions          PolicyDB Name : DATABASE_NAME
Entry Name : NNT_SESS_NAME        Enterprise Name : YOUR_ENTERPRISE

UserId : XXAOCXX

Enter the NetView domains used to establish NNT sessions.
Password: PROMPT or Password
Initialize: YES          =Initiated automatically
              NO          =Initiated thru operator interface

Domain  Password  Initialize  Description

```

Figure 189. NNT Sessions Definition Panel

The fields are:

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

A resident CLIST is a CLIST that is loaded into and kept in OS/390 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 OS/390 Automated Operations Network User's Guide* for more information on using the LOADCL command for CLISTs.

The customization dialog provides several sets of recommended resident CLISTs in the sample policy databases, for example, AOF_BASIC for system operations basic routines. Refer to the description provided for each set to find out which is appropriate for you.

Resident CLISTs Policy Object

If a new automation control file is loaded after SA OS/390 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 190 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.

```
ACTIONS  HELP
-----
AOFGEPOL          Policy Selection
Command ==>>>          SCROLL==>>> PAGE

Entry Type : Resident CLISTs      PolicyDB Name  : DATABASE_NAME
Entry Name  : AOF_BASIC           Enterprise Name : YOUR_ENTERPRISE

Action      Policy Name      Policy Description
DESCRIPTION
CLISTS      CLISTS              Define command lists to load to main storage
-----
WHERE USED  List Systems linked to this entry
COPY        Copy data from an existing entry
```

Figure 190. Policy Selection Panel for Resident CLISTs

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 OS/390.

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 OS/390-provided defaults. The entries specified here define statuses and the priorities assigned to those statuses. These entries are used by SA OS/390 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.

Status Details Policy Object

Figure 191 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.

```
ACTIONS  HELP
-----
AOFGEPOL          Policy Selection
Command ==>>>                                SCROLL==>>> HALF

Entry Type : Status Details      PolicyDB Name : DATABASE_NAME
Entry Name : STATUS_DETAILS_NAME Enterprise Name : YOUR_ENTERPRISE

Action   Policy Name      Policy Description
        DESCRIPTION    Enter description
        SDF DETAILS    Enter status display facility details
-----
        WHERE USED      List Systems linked to this entry
        COPY            Copy data from an existing entry
```

Figure 191. Policy Selection Panel for Status Details

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 OS/390 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*.

```
COMMANDS  HELP
-----
AOFGSCR          Status Display Facility Details
Command ==>>>                                SCROLL==>>> PAGE

Entry Type : Status Details      PolicyDB Name : DATABASE_NAME
Entry Name : AOFSDF             Enterprise Name : YOUR_ENTERPRISE

Status  Priority Highlight  Color  Clear      Srv Req(noadd)
RWTOR          (Y,RV)    NOADD
SPLGONE        (Y,RV)    NOADD
COMMLOST 110   REVERSE  RED       Y
BROKEN  120   REVERSE  RED       (Y,RV*)
BREAKING 130   BLINK    RED       (Y,RV*)
INACTIVE 140   UNDERSCORE RED       (Y,RV*)
IWTOR    150   RED      Y
SPLFULL  150   RED      (Y,RV)
```

Figure 192. Status Display Facility Details Panel

The fields are:

Status Any required name. The name must start with an alphabetic or numeric

Status Details Policy Object

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 OS/390 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 OS/390 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 OS/390, 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 OS/390 common routines described in *System Automation for OS/390 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:

Status Details Policy Object

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.

Communications Task Policy Object

The *Communications Task* policy object enables you to define and customize the communications tasks used by the processor operations functions of SA OS/390. Communications Tasks are used to establish a session between a processor operations focal point and a processor operations PC.

Creating a New Communications Task

```
COMMANDS  HELP
-----
A0FGCT00          Define New Entry
Command ===>

To define a new entry, specify the following information:
Type. . . . . Communications Task
Name . . . . . IPUFMCT3          (name of Comm Task LU)
Task Number. . . . . 3

Short Description. . . . .
Extended Description . . . . .
. . .
. . .
. . .
```

Figure 193. Define New Entry Panel for a Communications Task

This panel's unique fields are:

Name Is required. This gives the LU 6.2 name assigned to the communications task. The LU name is defined by the VTAM coordinator.

Task Number

Is already specified. Its value depends upon how many communications tasks are currently defined. With each addition of a communications task, this number is incremented by one. This field cannot be altered.

Communications Task Policy Object

Policy Items for Communications Task

Figure 194 shows the policy items that are displayed when a new Communications Task entry is defined or when an existing Communications Task entry is selected from the *Entry Name Selection* panel.

```
COMMANDS  HELP
-----
AOFGEPOL          Policy Selection
Command ==>>>                                SCROLL==>>>PAGE

Entry Type : Communications Task PolicyDB Name:
Entry Name :                               Enterprise Name:

Action      Policy Name      Policy Description
DESCRIPTION Enter description here
CONTROL PARAMETERS Define parameters for communications tasks
CONTROLLED PS/2S  Assign PS/2s to communications task
-----
COPY                               Copy data from an existing entry
```

Figure 194. Policy Selection Panel for Communications Task

The panel contains the following policy items:

CONTROL PARAMETERS

The *Communications Task Control Parameters* panel (Figure 195) is displayed.

CONTROLLED PS/2S

The *Controlled PS/2 List* panel (Figure 196 on page 282) is displayed.

More About Policy Item CONTROL PARAMETERS

Use the panel shown in Figure 195 to define parameters for SA OS/390 communications tasks.

```
ISQDPD06          Communications Task Control Parameters
Command ==>>>

Communications task: 3
Commtask LU name:   RE00NJW3

Enter the control parameters.

LOGMODE            => DSIL6MOD

Retry count        => 1

Event check time interval => 5           0-32700 seconds
Poll interval      => 60             0-32700 seconds
PS/2 poll response limit => 20       0-32700 seconds
PS/2 command wait limit  => 60       0-32700 seconds
Session start time limit => 180      0-32700 seconds
PASSTHRU idle time limit => 15       0-00545 minutes
```

Figure 195. Communications Task Control Parameters Panel

The panel contains the following fields:

Communications task

Gives the number of the communications task.

Commtask LU name

Shows the LU name of this communications task.

LOGMODE

Enter the LOGMODE name of the DLOGMODE value defined in the VTAM Network Control Program (NCP). Names can contain from 1 to 8 alphanumeric characters, starting with an alphabetic character, #, \$, or @. Use logmode DSIL6MOD for communication.

Retry count

Indicates the number of times the selected communications task will retry an operation when certain problems are detected. Valid values are from 0 to 99. Zero indicates no retry processing.

Problems which will be retried include:

- Restarting the selected communications task after a complete communications task failure.
- LU 6.2 communications failures where VTAM indicates that the problem may be resolved with a retry.

Note: Session establishment is a synchronous process within a communications task. While session establishment is in progress for one PS/2, the communications task cannot perform communications to other PS/2s. Specifying a large number of retries here may, in the case of a session establishment problem to one PS/2, result in timeouts for other PS/2s controlled by the same communications task.

Event check time interval

Enter the number of seconds (integers only) the system is to wait before checking each of the following intervals. The default is 5 seconds.

Poll interval

Enter the maximum length of time in seconds (integers only) that a PC session can remain idle before the focal point system requests verification that the PC is still active. The default is 60 seconds.

PS/2 poll response limit

Enter the length of time in seconds (integers only) that the focal point system is to wait for the PC to respond to the poll. This value should account for network delay. A value between 20 and 60 seconds is recommended. The default is 20 seconds.

PS/2 command wait limit

Enter the length of time in seconds (integers only) that the system is to wait for a response to a command sent to the PC processor before generating a PC Does Not Respond status. A value of 60 seconds or greater is recommended. The default is 60 seconds.

Session start time limit

Enter the length of time in seconds (integers only) that the system is to wait for a Start PC Session command to complete. A value between 180 and 240 seconds is recommended. The default is 180 seconds.

PASSTHRU idle time limit

Enter the length of time in minutes (integers only) that a pass-through session can be idle before being ended. This applies only to pass-through

Communications Task Policy Object

sessions initiated by users by means of the ISQTC command pass-through facility. The default is 15 minutes. You can disable the pass-through idle time limit by specifying all zeros in this field.

More About Policy Item CONTROLLED PS/2S

Use the panel shown in Figure 196 to add or delete a PC from the current communications task.

```
ISQDPD05          Controlled PS/2 List          Scroll ==> PAGE
Command ==>>>

Communications task: 3

Delete a PS/2 using the action column.  Enter a PS/2 name on the add line.
Action:  D = Delete PS/2

Action   PS/2 Name   PS/2 LU Name
Add   =>   _____

-        STEVE      RE00NJW2
-        TEDDY      RE00JW1
-        CALVIN     RNUS123
-        WOODROW    RNUS321
```

Figure 196. Controlled PS/2 List Panel

The panel contains the following fields:

Communications task

Gives the number of the communications task.

Action

Presents the action you can select. Enter D, which is the only valid value, to delete a PC from the communications task.

Add Enter the name of the PC to add to the communications task. When you add a PS/2 name, the new name is added to the list shown on this panel.

PS/2 name

Gives the name of the PC assigned to the communications task.

PS/2 LU name

Gives the LU name associated with the PS/2 name.

Communications Task Commands

Three commands facilitate the task of assigning processor operations PS/2 computers to communications tasks. These commands are found in the Commands pull-down menu of both the communications path *Entry Name Selection* panel and the communications task *Entry Name Selection* panel. These are:

- DISTRPS2 (T)
- DISPUNAP (P)
- DISPASGN (A)

DISTRPS2

When DISTRPS2 is selected, the panel shown in Figure 197 on page 283 is displayed. Use this panel to select how to distribute PCs not currently assigned to a communications task. Unassigned PCs can be distributed in three ways:

- Evenly among all communications tasks

Communications Task Policy Object

- Evenly among all empty communications tasks
- Evenly among the tasks you select

If you choose to distribute the PCs among the tasks you select, use this panel to identify those communications tasks.

```
ISQDPD08      Processor Operations PS/2 Distribution
Command ==>                                     Scroll ==> PAGE
Enter YES in one of the distribution fields A, B or C.
Total tasks:      10      Total PS/2s:      18
Tasks without PS/2s: 2      Unassigned PS/2s: 8

Do automatic (A), (B) or selective (C) distribution of unassigned PS/2s.

A. Enter YES if distribution of PS/2s is across all tasks      => NO_
B. Enter YES if distribution of PS/2s is across empty tasks    => NO_
C. Enter YES if distribution of PS/2s is across selected
  tasks.                                                        => YES
  Select the tasks by placing an X in the action field.

Action  Task Number  Task LU Name  Number of PS/2s
-       1             RE00NJW0     2
-       2             RE00NJW2     2
-       3             RE00NJW3     1
-       4             RE00NJW1     1
-       5             1            1
-       6             1            1
```

Figure 197. Processor Operations PS/2 Distribution Panel

The panel contains the following fields:

Total tasks

Gives the total number of communications tasks.

Tasks without PS/2s

Gives the number of communications tasks without PCs.

Total PS/2s

Gives the total number of PCs.

Unassigned PS/2s

Gives the number of PCs not assigned a communications task.

Choose option A, B, or C and type YES or Y in the corresponding input field.

A. Enter YES if distribution of PS/2s is across all tasks

To distribute PCs among all defined tasks, enter YES or Y. Otherwise, type NO or N.

B. Enter YES if distribution of PS/2s is across empty tasks

To distribute PCs among all empty tasks, enter YES or Y. Otherwise, type NO or N.

1. If you choose option A or B, press the End key to distribute the PS/2 computers.
 - a. Type YES or Y in the input field for option A or B.
 - b. Type NO or N in the next field. After selection is complete, enter YES.

Communications Task Policy Object

- c. After selecting the communications tasks, type an X in the *Action* field opposite the communications tasks you want.
- d. Press PF3 (End key) to distribute the PCs.

C. Enter YES if distribution of PS/2s is across selected tasks.

To distribute PCs among selected tasks, enter YES or Y. Use the action column to select the tasks. Otherwise, type NO or N.

1. If you choose option C:
 - a. Type YES or Y in the input field for option C.
 - b. Type NO or N in the next field. After selection is complete, enter YES.
 - c. After selecting the communications tasks, return to the *Action* field opposite the communications tasks you want.
 - d. Press PF3 (End key) to distribute the PCs.

The following are output fields:

Task Number

Gives the number assigned to the communications task.

Task LU Name

Gives the LU 6.2 name assigned to the communications task.

Number of PCs

Gives the number of PCs currently assigned (through the dialogs) to the communications task.

Note: Assign every PS/2 communications path to a communications task.

DISPUNAP

When DISPUNAP is selected, the following panel is displayed.

```
ISQDPD03                               Unassigned PS/2 List                               Scroll ==> PAGE
Command ==>>>

Enter END to leave PS/2 list.

      PS/2 Name      PS/2 LU Name
      ALANMPS2      RE00NJW2
      CHARLYB2      RE00JW5
      PS2000A1      RE00B000
      PS2000A2      RE00B001
      PS2000A3      RE00B002
      PS2000A4      RE00B003
      PS2000A5      RE00B004
      PS2000A6      RE00B005
      PS2000A7      RE00B006
      PS2000A8      RE00B007
      PS2000A9      RE00B008
      VINNYPS2      RE00JW1
      WILLYPS2      RE00JW0
      WOLFTSTA      RE05E0IN
```

Figure 198. Unassigned PS/2 List Panel

This panel contains the following fields:

PS/2 Name

Lists the PCs not assigned to a communications task. This field is the path name that was specified for the communications path PS/2 computer using the customization dialog.

PS/2 LU Name

Gives the LU names of PCs not assigned to a communications task. This field is the LU name that was specified for the communications path PS/2 computer using the customization dialog.

DISPASN

When DISPASN is selected, the following panel is displayed.

ISQDPD04 Processor Operations - Task-PS/2 Assignment
 Command ==> Scroll ==> PAGE

Total tasks: 7 Existing PS/2s: 71
 Tasks without PS/2s: 3 Unassigned PS/2s: 0

Task Number	Task LU Name	Number of distributed PS/2s
1	RE00NJW0	12
2	RE00NJW2	16
3	RE00NJW3	15
4	RE00NJW1	28
5	M02ST091	0
6	M02ST092	0
7	M02ST093	0

Figure 199. Processor Operations - Task-PS/2 Assignment Panel

The panel contains the following fields:

Total tasks

Gives the total number of communications tasks.

Tasks without PCs

Gives the number of communications tasks without PCs.

Existing PS/2s

Gives the number of PS/2 computers.

Unassigned PS/2s

Gives the number of defined PS/2 computers not assigned a communications task.

Task Number

Gives the number assigned to the communications task. This is an output field.

Task LU Name

Gives the LU 6.2 name assigned to the communications task. The LU name is available from your VTAM coordinator. This is an output field.

Number of distributed PCs

Gives the number of PCs currently assigned (through the dialogs) to the communications task. This is an output field.

Includes Policy Object

Includes Policy Object

This policy object is available only for clean-up or for down-level SA OS/390 systems and will no longer be supported. The BUILD function will still include this data set member into the automation control file, but SA OS/390 2.1 will ignore all user defined members and issue 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 OS/390.

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 OS/390 policy object. Using the automation control file to store policy-specific information helps keep your automation procedures generic.

Figure 200 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.

```

ACTIONS  HELP
-----
AOFGEPOL          Policy Selection
Command ==>>>          SCROLL==>> HALF

Entry Type : User E-T Pairs      PolicyDB Name : DATABASE_NAME
Entry Name : USER_ET_NAME      Enterprise Name : YOUR_ENTERPRISE

Action   Policy Name      Policy Description
        DESCRIPTION      Enter description
        E-T DATA        Define User Entry-Type pairs
-----
        WHERE USED      Lists Systems linked to this entry
        COPY            Copy data from an existing entry

```

Figure 200. Policy Selection Panel for User E-T Pairs

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 201 on page 287 shows the *UET Entry-Type Selection* panel.


```

COMMANDS  ACTIONS  VIEW  HELP
-----
AOFGUET          UET Entry-Type Selection          Row 20 from 40
Command ==>>>          SCROLL==>>> PAGE

Entry Type : User E-T Pairs          PolicyDB Name : DATABASE_NAME
Entry Name : JOBCARDS                Enterprise Name : YOUR_ENTERPRISE

Action  Entry Name          Type
      JOBCARD                DEFAULTS

```

Figure 201. UET Entry-Type Selection Panel

The *Entry Name* and *Type* fields are used to define a unique set of key-data pairs. In general, the *Entry Name* field can contain any name except subsystem names existing in the policy database. The *Type* field can be used to identify groups of information for the entry name. You can define wild cards and types by appending the asterisk (*) character to the type value.

When you enter an E for Edit in the *Action* column, another panel, similar to Figure 202 on page 288, 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
- Imbedded single quotation marks must appear in pairs.

User E-T Pairs Policy Object

```
COMMANDS  ACTIONS  HELP
-----
A0FGUKD          UET Keyword-Data Specification          Row 48 from 130
Command ==>>>          SCROLL==>> PAGE

Entry Type : User E-T Pairs          PolicyDB Name : DATABASE_NAME
Entry Name : JOBCARDS                Enterprise Name : YOUR_ENTERPRISE
Entry-Type : JOBCARD                  DEFAULTS

Action  Keyword/Data(partial)
      JOBNAME
      HAIAOC3*
      CLASS
      A
      MSGCLASS
      N
      MSGLEVEL
      (1,1)
      REGION
      4096K
```

Figure 202. 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 80.

Chapter 6. Building and Distributing Control Files

How to Define Data Sets for Build Processing	289	Building NetView and VTAM Information Used by Processor Operations	301
How to Build Data and Control Files from the Build Functions Menu	289	Generate NetView and VTAM Information	301
Building the Processor Control File	290	Files That Are Defined	302
Usage Notes.	291	Building PS/2 Profile Information	303
LAN Passthru Function	292	What Are PS/2 Profiles?.	303
Building an Automation Control File and an Automation Manager Configuration File	292	Contents of PS/2 Profiles	304
System Operations Control Files Build Options	293	Generating PS/2 Profiles	305
BUILD Error Messages	296	Customizing the PS/2 Computer for the Standalone Function	306
Distributing the System Operations Control Files	298		

After you have completed the SA OS/390 customization process, the following data and control files need to be built:

- Processor operations control file (needed for processor operations functions)
- System Operations control files (ACF and AMC file) (needed for automation)
- NetView and VTAM information (needed for processor operations functions)
- PS/2 profile information (needed for processor operations functions)

Note: You must use the customization dialog to create the system operations control files. Do not edit any of the system operations control 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 OS/390).

On the *Customization Dialog Primary Menu* panel of the customization dialog, you must select the *Settings* option. This brings up the *Customization Dialog Settings Menu* (see Figure 220 on page 352).

There, choose option 11, DS NAMES. This brings up the Data Set Customization Details panel (see Figure 223 on page 356).

For more information, refer to “Defining Data Sets for Batch Processing” on page 356.

How to Build Data and Control Files from the Build Functions Menu

Exiting the customization dialog does not automatically build these files. You build using the BUILD (B) command from the COMMANDS pull-down menu or from the command line from the *Policy Data Base Selection* panel or the *Entry Type Selection* panel. Then a panel as shown in Figure 203 on page 290 is displayed.

Building and Distributing Control Files

```
COMMANDS  HELP
-----
AOFGBLDK          Build Functions Menu
Command ==>>>

 1 ALL              - Build all files
 2 Proc-Ops        - Build Processor Operations Control File
 3 Sys-Ops         - Build System Operations Control Files
 4 NetView/VTAM    - Build NetView and VTAM information
 5 PS/2 Profiles   - Build PS/2 profile information
   PS/2 profile data set. . . .
```

Figure 203. Build Functions Menu Panel

The *Build Functions Menu* offers five functions to create control files:

1. Building all files
Causes all of the files required by all functions of SA OS/390 to be built. These files are the processor operations control file, the system operations control files, NetView and VTAM information, and PS/2 profile information. Selecting 1 is equivalent to selecting 2 through 5 sequentially.
2. Building 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 control files because the name of the processor operations file to be used for automation is placed in the ACF.
3. Building the system operations control files
Causes enterprise automation data for SA OS/390 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, and the automation manager configuration file (AMC file)
4. Building NetView and VTAM information
Causes NetView autotask definitions, NetView user task definitions, and VTAM application definition files for use by processor operations to be generated.
5. Building PS/2 profile information
Causes PS/2 profiles for use with the PS/2 stand-alone function of SA OS/390 processor operations to be generated. To create processor operation PS/2 profiles, type in the PS/2 Profile data set name. PS/2 profile information should be built only after successful completion of the Processor Operations control file build.

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, processors, communications paths,

Building the Processor Control File

and communications tasks 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 204 to identify the fully qualified names of both the SA OS/390 processor control file and the control file message log.

```
COMMANDS  HELP
-----
ISQDPG01      Processor Control File Generation
Command ==>

1 Generate Processor Control File

    Verify the fully qualified data set names for the Processor
    Control file and Control file log:
    Control file. . . . 'CONTROL.CTRF'
    Control file log. . . 'CONTROL.CTRF.LOG'

2 Browse Control file log
```

Figure 204. Processor Control File Generation Panel

For SA OS/390 processor operations, the processor control file and control file log must reside in a sequential data set.

The panel contains the following fields:

Control file

Identifies the data set containing configuration information for processor operations under NetView. Enter the fully qualified name of the control file for processor operations. The control file displayed on this panel is taken from what you entered in the panel shown in Figure 51 on page 91. The control file name cannot be DEBUG.

Control file log

Identifies the data set containing information about errors in the configuration definition. Enter the fully qualified name of the message log for the control file generation. When you generate a control file, this error file contains the errors or messages created during the generation procedure. The control file displayed on this panel is taken from what you entered in the panel shown in Figure 51 on page 91. Check this file after each control file generation.

Usage Notes

The SA OS/390 processor control file contains information specific to your SA OS/390 configuration and is used to start SA OS/390 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 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. For more information about allocating and cataloging data sets for the processor control file and processor control file log, see *System Automation for OS/390 Planning and Installation*.

Building the Processor Control File

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 the control file and/or control file log has been changed (using PF key PF3), the values are stored into the enterprise policy database. To avoid this, leave 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 ISQDPG01. This is option 2 in Figure 204 on page 291. To determine the correct response for any processor control file log messages, see *System Automation for OS/390 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 OS/390 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.

LAN Passthru Function

If you plan to use the LAN passthru function, the processor control file must be downloaded to the SA OS/390 workstation. After you have successfully built the processor control file, download the entire file (in binary) to the workstations, using an appropriate download tool. The file should be placed in the path that you have defined as your user data path.

Building an Automation Control File and an Automation Manager Configuration File

To construct a new enterprise automation configuration, that is, a set of new system operations control files (automation control file and automation manager configuration file), you have the following possibilities 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 203 on page 290).

Building an Automation Control File and an Automation Manager Configuration File

- Use option 3 on the *Build Functions Menu* (Figure 203 on page 290). This will bring you to the *ACF and AMC Build Parameters* panel shown in Figure 205 on page 294.
- 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 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: BUILDF SYS to build all defined system objects. You are presented with the *ACF and AMC Build Parameters* panel (see Figure 205 on page 294 with the entry fields filled in according to your previous selection. The build function builds all systems irrespectively 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 *ACF and AMC Build Parameters* panel (see Figure 205 on page 294) 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 configuration files are placed in the data sets you have defined in the Output Data Set field.

Note: Throughout this book the term **automation control file** is used to refer to the master include file and all the members it includes.

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 build and the AMC build will always be invoked together. This is due to the fact that the configuration data between the ACF and the AMC file 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 OS/390 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 OS/390. It starts with prefix ACF followed by four alphanumeric characters.

System Operations Control Files Build Options

The *ACF and AMC Build Parameters* panel allows you to specify the information required to build the system operations control files (ACF and AMC file) required for automation. You can build these system operations control files for stand alone

Building an Automation Control File and an Automation Manager Configuration File

systems, individual sysplexes or your complete enterprise, using the following options:

```

MENU  OPTIONS  HELP
-----
AOFGBLDP                ACF and AMC Build Parameters
Option ==>

 1 Build a complete enterprise
 2 Build sysplex group or stand alone system
   Sysplex / System name . . . . . (*, ?, or name)
 3 Build entry type or entry name
   Entry Type. . . . . SYS          (*, ?, or type)
   Entry Name. . . . . *           (*, ?, or name)
 4 View build report

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

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

```

Figure 205. ACF and AMC Build Parameters Panel

- 1 builds all enterprise automation control files and also the corresponding automation manager configuration files.

Note: This option must at least be successfully completed once before you can use options 2 or 3.

- 2 allows the selection of a single sysplex group or a stand alone system.

This build option performs a complete build of all objects linked to the selected sysplex or stand alone system. For a big policy database, this options 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 rebuild. 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. 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*, *Group*, it is checked whether data that affect the automation manager have been changed. In this case, building only the ACF fragment for the requested entry would produce a situation where your previously generated AMC files would no longer be compatible with your current ACF files. To avoid this situation, a complete build (using build type MOD) will automatically be invoked. Before the build actually starts, a confirmation panel will be displayed explaining this situation and allowing you to defer the build to a later point in time.

Building an Automation Control File and an Automation Manager Configuration File

If you enter a '?' into these field, SA OS/390 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 get a complete set of system operations control files (automation control file and automation manager configuration file). 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 stand-alone systems and sysplexes still remain in the configuration files. For example, a deleted sysplex or stand-alone 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 have been deleted, to clean up unnecessary entries from the configuration files.

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 blanks).

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
- 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.

Building an Automation Control File and an Automation Manager Configuration File

If for example a new timer is linked to one system, only the system include member for that system is rebuilt, while for all other system include members only the header timestamps are updated, and the AMC build is skipped, because timer data does not go to the automation manager.

If for example 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 for that sysplex.

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 stand-alone system are rebuilt.

The default value is MODIFIED (if this field is set to blanks).

The very first build after installing SA OS/390 2.1 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 pop-up 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.

Building an Automation Control File and an Automation Manager Configuration File

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 bottom of the build report, and specific messages showing 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 on 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 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).

Building an Automation Control File and an Automation Manager Configuration File

It is possible to tailor the detail level of the messages for the trigger and application group warning. When invoking the new option *0 Settings*, there is a selection *3 MESSAGE*, which 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 pop-up 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 INGBBLD of the SINGSAMP library. Build return codes are:

- 0 successful
- 4 build complete, however, some warning messages are issued.
- 20 severe error; build terminated without usable configuration

Distributing the System Operations Control Files

The SA OS/390 automation configuration consists of two components:

- the automation manager's automation manager configuration file (AMC)
- the automation agents' automation control files (ACF)

These files are also referred to as *system operations control files* in the customization dialog.

The partitioned data set generated by the BUILD contains both the automation control file and the automation manager configuration file. SA OS/390 runs on a sysplex basis, that is, the same ACF and AMC files must be available to all of the automation agents and automation managers in the sysplex.

The automation agent automatically finds the automation control file 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 OS/390 partitioned data sets in *System Automation for OS/390 Planning and Installation*.) The automation control file must either be shared or have exactly the same name on all systems within a sysplex. It can be shared within an enterprise.

For the automation manager you have a choice:

Building an Automation Control File and an Automation Manager Configuration File

- 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 OS/390 and at least one automation manager needs to run per sysplex and one per stand-alone system. Automation managers can manage systems sysplex-wide. Figure 206 shows a sample automation configuration and presents the build process.

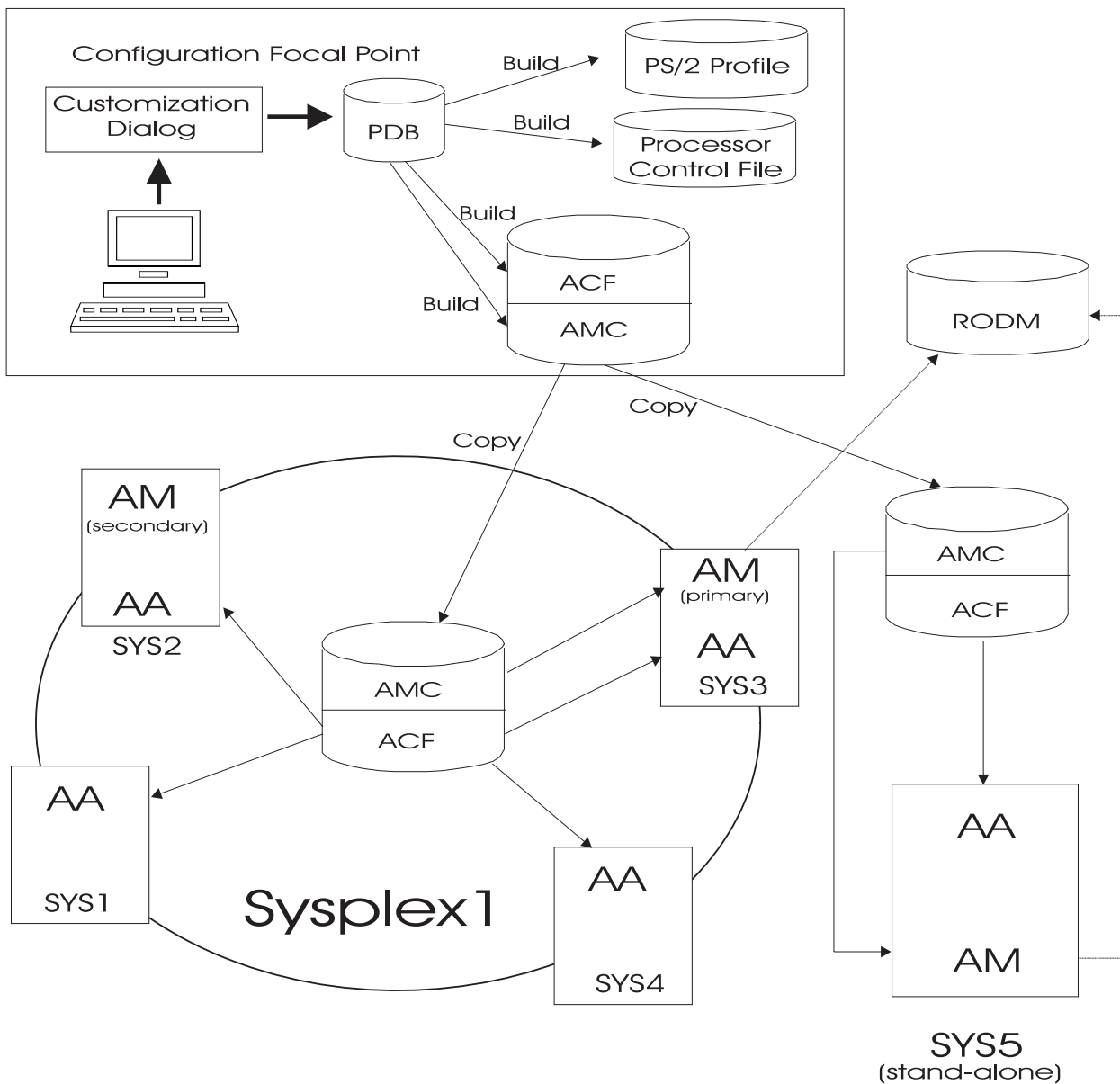


Figure 206. SA OS/390 Automation Configuration

Figure 206 shows a sample enterprise consisting of one sysplex *Sysplex1* and one stand-alone system *SYS5*. The sysplex consists of four systems *SYS1*, *SYS2*, *SYS3*,

Building an Automation Control File and an Automation Manager Configuration File

SYS4 and has a primary and a secondary automation manager (AM). The sysplex shares a copy of the system operations control files and the stand-alone 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 stand-alone system must have one automation manager (AM) and one automation agent (AA).

The enterprise wide policy database for your sysplex and the stand-alone system is created and maintained with the customization dialog on any suitable system. For the purpose of the diagram shown in Figure 206 on page 299, 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 PS/2 Profile
- the Processor Operations Control File
- the System Operations Control Files consisting of the automation control file (ACF) and the automation manager configuration (AMC).

You could share one common automation control file in your enterprise, or you can copy the automation control 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 OS/390 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 agents' 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 control files must be distributed to each system in the SA OS/390 enterprise where SA OS/390 system operations functions are to run. In general, this is all of the OS/390 systems in the SA OS/390 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 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.

Building NetView and VTAM Information Used by Processor Operations

To access the NetView and VTAM information to build, you can use option 4 of the *Build Functions Menu*.

When you request a build of NetView and VTAM information, the following are generated:

- NetView autotask definitions
- NetView data service task definitions
- VTAM application definition files

Generate NetView and VTAM Information

Use the panel shown in Figure 207 to specify the NetView and VTAM definitions.

```

ISQDPN01                Generate NetView and VTAM Information
Selection ==>

Make a selection and supply input.

1 OPERATOR                Generate NetView Operator Definitions
  Data set name           => _____
  Number of monitor tasks   => 2__
  Number of target control tasks => 3__

2 TASK                    Generate Task Definitions
  Data set name           => _____
  Number of communications tasks => 5__

3 APPLICATION              Generate VTAM Application Definitions
  Data set name           => _____
  
```

Figure 207. Generate NetView and VTAM Information Panel

The panel contains the following fields:

Selection

Enter the number or keyword of one of the following options. Complete any related fields, and press the Enter key.

Option Description

- | | |
|---|--|
| 1 | Generate NetView Operator Definitions. |
| 2 | Generate Task Definitions. |
| 3 | Generate VTAM Application Definitions. |

Data set name

Enter the names of the data sets to receive the definition statements. Follow TSO naming conventions in choosing names for the data sets. If the data set names are not fully qualified, the customization dialog uses the current value of your TSO PREFIX as the high-level qualifier for the data set names. The high-level qualifier is the current TSO PREFIX value. If the existing data sets are partitioned, you must specify member names. If the data sets do not exist, SA OS/390 creates sequential data sets unless you specify member names.

Building NetView and VTAM Information

Number of monitor tasks

Specify the number of monitor tasks for this focal point system. Accept the value displayed or enter a different value. The displayed value indicates the number of tasks that will be generated. This value is the result of an entry you made in panel *Processor Operations Information* (ISQDPD02) when you defined processor operations policy for an enterprise object. See Figure 51 on page 91. The valid range is from 1 to 999.

Number of target control tasks

Specify the number of target control tasks for this focal point system. Accept the value displayed, or enter a different value. The displayed value gives the number of tasks that will be generated. This value is the result of an entry you made in panel *Processor Operations Information* (ISQDPD02 found in Figure 51 on page 91) when you defined processor operations information for an enterprise object. The valid range is from 1 to 999. The number of target control tasks should be equal to the maximum number of active target hardware systems.

Number of communications tasks

Specify the number of communications tasks for this focal point system. Accept the value displayed, or enter a different value. The displayed value gives the number of tasks that will be generated. This value is the result of entries you made when you defined Communications Task Policy. The valid range is from 1 to 999.

In configurations where no processor operations PCs are defined, the number of communication tasks is zero.

Files That Are Defined

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 OS/390 processor operations.

To retain any operator and autotask definitions that you already use, make sure you generate the SA OS/390 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.

NetView Task Definition File

Each SA OS/390 communications task is a NetView data service task. These tasks must be defined to the NetView program to enable SA OS/390 communication.

Integrate the NetView task definitions into the DSIDMN member (for supported NetView below 1.4) or the CNMSTYLE member (for NetView1.4 and above) of the NetView DSIPARM data set.

VTAM Application Definition File

Each processor operations PC in your SA OS/390 configuration is a partner logical unit (LU) for its associated communications task. The

Building NetView and VTAM Information

communication task LUs must be defined to VTAM as VTAM applications to establish the LU 6.2 session between the focal point system and the PCs. The application IDs of the generated VTAM definitions are the Commtask LU names entered in the *Define New Entry* panel when you define Communications Task policy.

Integrate the VTAM application definitions into the SYS1.VTAMLST VTAM initialization data set.

This process does not generate the PS/2 computer LU definitions. The PS/2 computer should be defined to the focal point host as an independent LU 6.2 by your VTAM coordinator.

The AOFDMN and AOFOPF files contain commented %INCLUDE statements for files called AOFDMNPO and AOFOPFPO. If those %INCLUDE statements are uncommented, the generated definition files should be integrated into these AOFDMNPO and AOFOPFPO files. For NetView 1.4 and above, the AOFDMNPO task definitions should be integrated into the AOFSTYLE DSIPARM member as AOFDMN no longer exists.

Building PS/2 Profile Information

Choose Option 5 to build PS/2 profile information. When you do, you need to enter the processor operations PS/2 profile data set name. Following that, the PS/2 profiles for use with the PS/2 standalone function are generated.

What Are PS/2 Profiles?

When SA OS/390 processor operations is running with the stand-alone function, a connection is not required between the SA OS/390 processor operations personal computer and its focal point system. Instead of a session with the focal point system to provide the required data, the information is obtained from a profile on the SA OS/390 processor operations personal computer.

In general, a profile describes the ports of a SA OS/390 processor operations personal computer, so that SA OS/390 processor operations PC code can exchange data with the ports. The port description sets up a relationship between the hardware for that port (for example, a 3270 connection card) and the type of console connected to that card (for example, an MVS console). This information is required by the SA OS/390 processor operations code to service the ports correctly.

When you initialize the standalone function on the PS/2 computer, a customized PS/2 profile is executed. The PS/2 profile contains all the commands that are needed to configure and activate the target system consoles. An operator at the PS/2 computer can now operate the target system.

The profile is stored as a file on the SA OS/390 processor operations personal computer. PS/2 profiles are generated on the focal point system using the SA OS/390 customization dialog. Before the profiles can be used, you must download them to the SA OS/390 processor operations personal computer. See "Customizing the PS/2 Computer for the Standalone Function" on page 306 for more information about downloading the PS/2 profiles and customizing the SA OS/390 processor operations personal computer to run with the stand-alone function.

On the focal point system, the PS/2 profiles are stored in a partitioned data set. One member in the partitioned data set is allocated for each processor operations

Building PS/2 Profile Information

personal computer in your processor operations configuration. The member name of the profile is the name of the processor operations personal computer.

The PS/2 profiles can be modified either by rerunning the customization dialog or by editing the focal point system profile data set or the PS/2 profile files. These alternatives let you choose the method that is most convenient for a given task, whether it is installing and testing a new target system processor and an associated processor operations personal computer or involves widespread changes throughout your enterprise. When taking advantage of this flexibility, however, it is important to keep the PS/2 configuration data on both the processor operations PCs and the focal point system synchronized. Refer to *System Automation for OS/390 Planning and Installation* for more information about changing profiles and associated PS/2 configuration data on the focal point system.

Contents of PS/2 Profiles

A PS/2 profile consists of a set of commands accepted by the processor operations PC code. For a generated profile, the base for these commands is the PS/2 definitions in the customization dialog. The contents of a generated PS/2 profile can be divided into the following groups:

- **Port definition of the PS/2 control port**
A processor operations PS/2 control port must be defined. When using the processor operations stand-alone function, this port is needed for entering commands to other target system ports and for entering commands to control the PS/2 control port itself.
- **Start of optional WWV external clock**
If a WWV external clock is connected to your processor operations personal computer, the command to start the external clock routine is placed in the generated profile. The WWV parameters, specified in the customization dialog for the processor operations PS/2, are used in this command.
- **Port definitions and port settings for target ports**
The 3270 or ASCII ports of a processor operations personal computer are defined using the *Processor Operations Port Configurations* panel of the customization dialog. If a PS/2 profile is generated, these definitions are used to build a set of commands that reflect the definitions.
- **Set the PS/2 control port as the active port for all other ports**
When a processor operations personal computer is operating using the stand-alone function, the PS/2 computer's display and keyboard serve as the system console or operator display console or both for the target systems being controlled. To display messages from the target ports on the PS/2 display and to have commands entered on the PS/2 keyboard sent to the target ports requires that the PS/2 control port become the processor operations active port. A command in the profile establishes this relationship.

See *System Automation for OS/390 Programmer's Reference* for information about the commands and parameters related to the standalone function.

Generating PS/2 Profiles

When the standalone function is installed on your focal point system, the PS/2 profiles can be produced by generating them in the processor operations customization dialog on the focal point system. Perform the following steps to generate the PS/2 profiles for your processor operations environment:

1. Allocate the profile data set.

Before starting the BUILD, you must allocate a partitioned data set for the PS/2 profiles. A sample job in member INGEALLO of the SINGSAMP sample library shows the recommended allocation:

```
//PS2PROF DD DSN=ISQ.PS2.PROFILES,VOL=SER=??????,UNIT=????,
//          DISP=(NEW,CATLG),SPACE=(TRK,(10,5,10)),DSORG=PO,
//          DCB=(LRECL=80,RECFM=FB,BLKSIZE=6400)
//*-----
```

2. Invoke the *Build Functions Menu*.
3. Enter the PS/2 profile data set name.

To generate PS/2 profiles, the PS/2 Profile data set name field must contain the name of the preallocated partitioned data set for the profiles. If you do not want to generate PS/2 profiles, leave this field blank. Otherwise, type in the name of the PS/2 profile partitioned data set. The fully qualified data set name must be specified. Make sure your TSO user ID is authorized to update this data set.

Building PS/2 Profile Information

Note: All PS/2 definitions should be completed before exiting the PS/2 subset of the customization dialog. When you exit the customization dialog, incomplete PS/2 definitions will produce a profile with unpredictable results.

If any errors occur while generating the control file, unpredictable results may occur. In this case, you should generate the PS/2 profiles again.

If you modify the PS/2 definitions, you must download the profiles again.

4. Select Option 5 to build the profile.

The PS/2 profiles are generated when you exit this panel. If you have no PS/2 definition in your processor operations configuration, no profile is generated. Exit the panel by pressing the *End* key.

Customizing the PS/2 Computer for the Standalone Function

After the standalone function code is installed on the PS/2 computer, you must customize the OS/2[®] functions on the processor operations PC. After the customization, you can have the standalone function start automatically each time the processor operations personal computer is booted. This section describes how to implement the automatic start of processor operations on the personal computer. Also, information in this section can help you download PS/2 profiles from the focal point system.

The basic customization steps are as follows:

1. Create a subdirectory for the processor operations PS/2 stand-alone profiles.

For example, if the processor operations PS/2 code is to be installed on the C drive in subdirectory ISQPS2, select the root directory of drive C as the target for the ISQUSR subdirectory, as follows:

```
C:  
CD\  
MD ISQUSR
```

2. Download the PS/2 profile.
 - a. Make sure that the partitioned data set that contains your PS/2 profile members exists on the focal point system.
 - b. Make sure that the Communications Manager/2 is started with a configuration file customized for processor operations.
 - c. To prepare for downloading the profile, log on to TSO at the focal point system through the LU2 session of the processor operations personal computer.
 - d. To download the profile, use the RECEIVE command or the file transfer panels of Personal Communications. The PS/2 profiles are stored on the focal point system in EBCDIC format. When you download the profile, you must specify the appropriate file transfer parameters to convert the PS/2 profile to its ASCII format and add carriage return and line feed characters. Ensure that the default code page on the focal point system and the default used on your processor operations personal computer provide the correct character translation from EBCDIC to ASCII. If not, you must specify the focal point system file code page and the PC file code page when you download the profile. For more information about the file transfer, see the Communications Manager help panels or the appropriate Communications Manager manuals.

Building PS/2 Profile Information

Enter a RECEIVE command in an OS/2 session of the processor operations personal computer as follows:

```
RECEIVE x:\direct\member.pro y:'partition_dataset(member)' ASCII CRLF
```

where:

x Is the PS/2 fixed drive where the profile is to be stored.

direct

Is the subdirectory where the profile is to be stored.

member.pro

Is the member name of the profile.

y Is the focal point system session identifier for downloading.

partition_dataset

Is the partitioned data set where the profile is stored.

ASCII CRLF

Indicate that profile is to be converted from EBCDIC to its ASCII format.

An example of a RECEIVE command entered in an OS/2 session of the processor operations PC follows:

```
RECEIVE C:\ISQUSR\CHICAGO.PRO A:'ISQ.PS2.PROFILES(CHICAGO)' ASCII CRLF
```

The profile is in the partitioned data set ISQ.PS2.PROFILES. The member name of the profile is CHICAGO. This is also the name of the processor operations personal computer. On the processor operations personal computer, the profile is stored in the subdirectory ISQUSR of the C drive as file CHICAGO.PRO. The focal point system session for downloading the profile has the identifier A. The parameters ASCII CRLF are specified to convert the profile from EBCDIC to its ASCII format. Because the default code pages on the focal point system and on the processor operations personal computer provide the correct character conversion, no further parameters are specified.

3. Update the OS/2 STARTUP.CMD file (optional).

To enable the standalone function automatic startup, modify your STARTUP.CMD file. This file is executed each time the processor operations PC is booted. Insert the following lines in your STARTUP.CMD file: **The ISQXEC SW statement must be coded before the ISQXEPC statement.**

```
START CM tscfcfg  
x:\ISQPS2\ISQXEC SW  
x:\ISQPS2\ISQXEPC . EXEC path\filename
```

where:

x Is the PS/2 fixed disk drive where the processor operations PS/2 code is installed

Dot (.)

Is the temporary processor operations PS/2 port name

EXEC

processor operations PS/2 command executes a profile

path

Specifies the path of the processor operations PS/2 profile

Building PS/2 Profile Information

Backslash (\)

Separates the path and file name specification

filename

Indicates the file name with file extension of the PS/2 profile

tscfcfg

Indicates the Communication Manager profile

Samples of the processor operations autostart statements follow:

```
START CM TSCFCFG
C:\ISQPS2\ISQXEC SW
C:\ISQPS2\ISQXEPC . EXEC C:\ISQUSR\CHICAGO.PRO
```

The processor operations PC code is installed on hard disk drive C in the ISQPS2 subdirectory. The PS/2 profile is stored as file CHICAGO.PRO in the subdirectory ISQUSR of drive C.

For more information on PS/2 profiles, see *System Automation for OS/390 Planning and Installation*.

Chapter 7. Migration Functions

This chapter provides information about functions provided by SA OS/390 to migrate from earlier releases to the current release. In detail, the following topics are discussed:

- “Migrate Automation Control File”
- “Merge TSCF Configuration” on page 324

Migrate Automation Control File

With the introduction of the automation manager, it is necessary to create the new automation manager configuration file, which can only be built using the customization dialog. So 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 **NEW**. On the follow-on panel, a model policy database (PolicyDB) name is required. 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 OS/390 Policy Databases” on page 20. This allows to select a sample that best matches the 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 MIGRATE command for the policy database into which you want to copy the automation control file data. Then the *System Migration Definitions* panel will be displayed. On this panel, the entry name for the system that will be used as the target, and the MVS sysname 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.

Furthermore, in the *Source Data Set* field, enter the data set name together with the member name containing 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 *System Migration Definitions* panel has to be selected. This gives

Migration Functions

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 *System Migration Definitions* panel.
6. All data belonging to one entry in the ACF are copied into one entry within the target PolicyDB. Dependent 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 '.

You have the following alternative methods of invoking the migration function:

- From the *Customization Dialog Primary Menu* (Figure 5 on page 14) select option 5 *Migrate*.
- From the *Policy Data Base Selection* panel, you can select action **Migrate (G)** against the policy database to which you want to migrate your data.

In both cases, you reach the *System Migration Definitions* panel shown in Figure 208 on page 311 that you can use to migrate an automation control file into a policy database of the current release of SA OS/390.

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 available are described in the following.


```

MENU  HELP
-----
AOFGMSYN                System Migration Definitions
Option ==>>

  1 Select Migration Objects
  2 Start Online Migration
  3 Start Batch Migration
  4 View Migration Report

Source definitions:
  Dataset & member. . 'HART.SH.POLICYDB(ACFZ998) '

Target definitions:
  PolicyDB name      : SCENARIO
  Dataset name       : 'HART.SCENARIO'
  Report member . . . $FOC1RPT           Member name in target data set
  Overwrite system. . NO                 YES NO
  System Entry Name . SAMPLE_SYSTEM_01   Entry Name of new/changed system
  MVS Sysname . . . . FOC1               MVS System Name
  Short Description .

Job statement information: (used for BATCH build)
//KEY3$MIG JOB (DE03160,,),'KEY3',CLASS=A,USER=KEY3, _____
//      MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=KEY3_____
//*

```

Figure 208. System Migration Definitions Panel

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 a selective migration of objects from ACF fragments. It leads you to the *System Object Migration* panel (Figure 209 on page 313) 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 even select certain existing entry names that you want to migrate. To achieve this, type the command *EDIT (E)* against a selected entry type. Then the *Entry Name Migration* panel (Figure 210 on page 314) is displayed.

After you have selected all objects (entry types and entry names) for migration, return to the *System Migration Definitions* 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 will start the migration process in the background. In this case, you need to provide a valid JOB statement under *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 356.

Migration Functions

- **4 View Migration Report**

This option invokes the ISPF VIEW function to view the migration report that had been 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 from 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 with reading the ACF data.
4. If option 1 is used again after the selections have been canceled, processing starts again with reading the ACF data.

Below these options, you specify:

- the *Source definitions*, that is, the data set and member of the automation control file that you want to migrate, for example, 'KEY3.SH.POLICYDB(ACFZ998) '
- the *Target definitions*, that is, all required information about the targets of your migration. *PolicyDB name* and *Dataset name* are already specified. Their values are taken from the policy database you have selected as policy database where the data will 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:

In the field *Overwrite system*, you can prevent SA OS/390 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 which already exists in the policy database. To proceed, you must either change the system name or set *Overwrite system* to YES.

Exiting the *System Migration Definitions* panel will cause any previously saved data to be deleted.

For detailed information about the individual input fields on this panel, refer to the online help.

```

COMMANDS  ACTIONS  HELP
-----
AOFGMOBJ          System Object Migration
Command ==>>>                                SCROLL==>> PAGE

Entry Name  : SAMPLE_SYSTEM_01      PolicyDB Name  : DATABASE_NAME
System Name : SYS1                  Enterprise Name : YOUR_ENTERPRISE

Target DSN   : 'KEY3.DOCU.PDB'
Source DSN   : 'KEY3.SH.POLICYDB(ACFZ998)'
```

Action	Migr-Status	Type	Description
		ENT	Enterprise
		GRP	Groups
		SBG	Sub_groups
		APG	Appl_groups
	SELECTED	APL	Applications
		TMR	Timers
		TMO	Timeout_Settings
		TPA	Tape_Attendance
		MVC	MVS_components
		MDF	MVSCOMP_defaults
		SDF	System_defaults
		ADF	Appl_Defaults
		AOP	Auto_Operators
		AMC	Auto_Msg_Classes
		NTW	Network
		NNT	NNT_Sessions
		RES	Resident_Clists
		SCR	SDF_Status_Details
		EVT	Events
		SVP	Service_Periods
		TRG	Triggers
		UET	User_ET_Pairs

Figure 209. System Object Migration Panel

You reach this panel, if you select option **1 Select Migration Objects** from the *System Migration Definitions* panel shown in Figure 208 on page 311. On this panel, initially all entry types are selected. You can apply the following actions against 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 210 on page 314).

If you enter command **CANCEL**, all changes made on this panel and on panel *Entry Name Migration* (which is displayed if action "EDIT" is selected) are deleted and the migration process starts as if it was being invoked for the first time.

Migration Functions

```

COMMANDS  HELP
-----
AOFGMENM          Entry Name Migration
Command ==>>>                                SCROLL==>> PAGE

Entry Name  : SAMPLE_SYSTEM_01      PolicyDB Name  : DATABASE_NAME
System Name : FOC1                  Enterprise Name : YOUR_ENTERPRISE

You may specify the name and description used to migrate the APL entries
into the Policy DB. To avoid migration blank out the name.

Migration Name  Dup  Description
APPC            *   APPC/MVS automation policy
ASCH            *   ASCH automation policy
JES2            *   JES2 policy definition
JES3            *   JES3 automation policy
TRANSIENT1      *   Transient job 1 policy details
TRANSIENT2      *   Transient job 2 policy details
TSO             *   TSO automation policy
VLF             *   VLF automation policy
VTAM            *   VTAM policy definition

```

Figure 210. Entry Name Migration Panel

An * in column *Dup* denotes 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 panel *System Migration Definitions* 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 OS/390 automation control file. The information from your old automation control file is migrated into an SA OS/390 policy database. To build an SA OS/390 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 12 describes how SA OS/390 creates names and descriptions for the entry types to be migrated.

Table 12. Creation of Entry Names

Entry Type	Entry Name	Description
ENT	From table AOFTENT (ENTNAME)	From table AOFTENT (ENTDESCS)
SYS	From panel AOFGMSYN	From panel AOFGMSYN
GRP	From ACF line 'GROUP entryname'	From ACF line '* Desc : short_description' or '' if the '* Desc' line is not available

Table 12. Creation of Entry Names (continued)

Entry Type	Entry Name	Description
SBG	From ACF line 'SUBGROUP entryname'	From ACF line '* Desc : short_description' or '' if the '* Desc' line is not available
APG	From ACF line 'APPLGROUP entryname'	From ACF line '* Desc : short_description' or '' if the '* Desc' line is not available
APL	From ACF line '* Built : userId yy/mm/dd hh:mm:ss entryname' or from ACF line 'SUBSYSTEM_sysname subsystemname' if the '* Built' line is not available. The name created will be: 'subsystemname_sysname'.	From ACF line '* Desc : short_description' or '' if the '* Desc' line is not available
TMR	From ACF line '* Built : userId yy/mm/dd hh:mm:ss entryname' or 'TIMER_sysname' if the '* Built' line is not available	From ACF line '* Desc : short_description' or '' if the '* Desc' line is not available
TMO	From ACF line '* Built : userId yy/mm/dd hh:mm:ss entryname' or 'TMO_SET_sysname' if the '* Built' line is not available	From ACF line '* Desc : short_description' or '' if the '* Desc' line is not available
TPA	From ACF line '* Built : userId yy/mm/dd hh:mm:ss entryname' or 'TAPE_ATT_sysname' if the '* Built' line is not available	From ACF line '* Desc : short_description' or '' if the '* Desc' line is not available
MVC	From ACF line '* Built : userId yy/mm/dd hh:mm:ss entryname' or 'MVS_COMP_sysname' if the '* Built' line is not available	From ACF line '* Desc : short_description' or '' if the '* Desc' line is not available
MDF	From ACF line '* Built : userId yy/mm/dd hh:mm:ss entryname' or 'MVSCOMP_DFT_sysname' if the '* Built' line is not available	From ACF line '* Desc : short_description' or '' if the '* Desc' line is not available
SDF	From ACF line '* Built : userId yy/mm/dd hh:mm:ss entryname' or 'SYSTEM_DFT_sysname' if the '* Built' line is not available	From ACF line '* Desc : short_description' or '' if the '* Desc' line is not available
ADF	From ACF line '* Built : userId yy/mm/dd hh:mm:ss entryname' or 'APPL_DFT_sysname' if the '* Built' line is not available	From ACF line '* Desc : short_description' or '' if the '* Desc' line is not available
AOP	From ACF line '* Built : userId yy/mm/dd hh:mm:ss entryname' or 'AUTOOPS_sysname' if the '* Built' line is not available	From ACF line '* Desc : short_description' or '' if the '* Desc' line is not available
AMC	From ACF line '* Built : userId yy/mm/dd hh:mm:ss entryname' or 'AUTOMSG_CLS_sysname' if the '* Built' line is not available	From ACF line '* Desc : short_description' or '' if the '* Desc' line is not available

Migration Functions

Table 12. Creation of Entry Names (continued)

Entry Type	Entry Name	Description
NTW	From ACF line '* Built : userId yy/mm/dd hh:mm:ss entryname' or 'NETWORK_sysname' if the '* Built' line is not available	From ACF line '* Desc : short_description' or '' if the '* Desc' line is not available
NNT	From ACF line '* Built : userId yy/mm/dd hh:mm:ss entryname' or 'NNT_SESSION_sysname' if the '* Built' line is not available	From ACF line '* Desc : short_description' or '' if the '* Desc' line is not available
RES	From ACF line '* Built : userId yy/mm/dd hh:mm:ss entryname' or 'RES_CLISTS_sysname' if the '* Built' line is not available	From ACF line '* Desc : short_description' or '' if the '* Desc' line is not available
SCR	From ACF line '* Built : userId yy/mm/dd hh:mm:ss entryname' or 'STAT_DETAIL_sysname' if the '* Built' line is not available	From ACF line '* Desc : short_description' or '' if the '* Desc' line is not available
EVT	From ACF line '@EXTCOND entryname'	From ACF line '* Desc : short_description' or '' if the '* Desc' line is not available
SVP	From ACF line '@SERVICE entryname'	From ACF line '* Desc : short_description' or '' if the '* Desc' line is not available
TRG	From ACF line '@TRIGGER entryname'	From ACF line '* Desc : short_description' or '' if the '* Desc' line is not available
PRD	not applicable	not applicable
UET	From ACF line '* Built : userId yy/mm/dd hh:mm:ss entryname' or 'USER_ET_sysname' if the '* Built' line is not available	From ACF line '* Desc : short_description' or '' if the '* Desc' line is not available

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 12 on page 314 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.

Note:

- 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*.
- 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*.

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 OS/390 2.1. So 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 SA OS/390 2.1. If the definitions are not known to SA OS/390 2.1, 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:

Migration Functions

```
SA OS/390 UET Keyword-Data Specification      Row 83 from 194
Command ==>>>                               SCROLL==>>> PAGE

Entry Type : User E-T Pairs      PolicyDB Name  : DATABASE_NAME
Entry Name  : HSM                Enterprise Name : YOUR_ENTERPRISE
Entry-Type  : HSMMAIL           BCDS

Action      Keyword/Data
-----
MAILLIMIT
      80%
MAILOP
      (I-SC.DATABASESYSTEM)
MAILTXT
      (BACKUP CONTROL DATASET OF HSM)
MAILTXT
      (IS 80% FULL)
LIMIT
      97%
```

Figure 211. 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:

Migration Functions

```
*                                SUBSYSTEM SPECIFICATION
SUBSYSTEM CICS4A,
  PARENT=(CICS4),
  SHUTDLY=00:03:00,
  STRTDLY=00:03:00,
  JOB=CICS4A,
  SDESC='TEST-CICS'
*                                DEFINES CICS
CICSCNTL CICS4A,
  APPLID=IPUAJCIA,
  MAJNODE=IPUJBC,
  AUTOOPS=CICSOP01,
  KEYPOINTREQ=YES,
  SMFID=215
*                                HEALTH CHECKING
HEALTHCHK CICS4A,
  FUNCTION=(0,EVECHLTH,00:10:00,30,AUTO,'SAMPLE COBOL HEALTH')
*                                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
*                                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),
  INFR=(5,24:00)
*
```

Figure 212. 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 SA OS/390 2.1.

The following screens show how the ACF entries from Figure 212 will be migrated into the customization dialog.

Migration Functions

SUBSYSTEM SPECIFICATION

```
SUBSYSTEM CICS4A,  
  PARENT=(CICS4),  
  SHUTDLY=00:03:00,  
  STRTDLY=00:03:00,  
  JOB=CICS4A,  
  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,  
  APPLID=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:

```
COMMANDS  HELP  
-----  
CICS control specifications  
Command ==>  
  
Entry Type : Application      PolicyDB Name  : DATABASE_NAME  
Entry Name  : CICS4A_sysname  Enterprise Name: YOUR_ENTERPRISE  
  
Subsystem name: CICS4A  
  
Enter or update the following fields:  
  
  APPLid. . . . IPUAJCIA  
  Major node. . IPUJBC  
  Keypoint req. YES      Keypoint request (Yes/No)  
  SMFID . . . . 215  
  
Auto operator          ( SA13 only )  
Start timer            ( SA13 only )  
Cold timer             ( SA13 only )  
Warm timer             ( SA13 only )  
Emerg. timer           ( SA13 only )
```

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:

```

COMMANDS  ACTIONS  HELP
-----
Command ==>>                Message Processing                Row 1 to 8
                               SCROLL==>> PAGE

Entry Type : Application      PolicyDB Name : DATABASE_NAME
Entry Name : CICS4A_sysname   Enterprise Name : YOUR_ENTERPRISE

Subsystem : CICS4A

Enter messages issued by this resource that will result in automated actions.
Actions: CMD = Command REP = Reply CODE = CODE USER = User defined values

Action  Message ID                Cmd  Rep  Code  User
-----  -
Description
ABCODESYSTEM                34
Specifications for ABCODESYSTEM
ABCODETRAN                  4
Specifications for ABCODETRAN
user  HEALTHCHK                    1
Specifications for HEALTHCHK

```

Selecting message HEALTHCHK with action USER will present the following panel:

```

COMMANDS  HELP
-----
Command ==>>                User Defined Data                Row 1 to 4 of 21
                               SCROLL==>> CSR

Entry Type : Application      PolicyDB Name : DATABASE_NAME
Entry Name : CICS4A_sysname   Enterprise Name : YOUR_ENTERPRISE

Subsystem : CICS4A Message ID : HEALTHCHK

To change keyword-data pair, specify the following:

Keyword
Data
FUNCTION
(0,EVECHLTH,00:10:00,30,AUTO,'SAMPLE COBOL HEALTH')

```

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:

Migration Functions

```

COMMANDS  HELP
-----
                                CICS connection links
Command ==>

Entry Type : CICS link           PolicyDB Name : DATABASE_NAME
Entry Name : CICS4AIM4A         Enterprise Name : YOUR_ENTERPRISE
-----

Enter or update the following fields:

Connection id . . . . . IM4A           CICS sysid
Type. . . . . IMS                     (IMS/CICS)
Echo. . . . . (X,60)                 (txn/pgm,seconds)
Time zone offset. . . . . 0           (hh:mm,WEST/EAST)
Check interval. . . . . 00:05         (hh:mm)
Max repair tries. . . . . 5           (0-9)
Release delay . . . . . 15            (4-99)
Acquire delay . . . . . 15            (4-99)
Critical link (SDF) . . YES           YES      NO

```

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 213.

```

COMMANDS  HELP
-----
                                Link monitoring service period
Command ==>

Entry Type : Monitoring period     PolicyDB Name : DATABASE_NAME
Entry Name : CICS4AIM4A           Enterprise Name : YOUR_ENTERPRISE

Note: Review help panel for complete description of data entry fields.
      Valid start and stop times are 00:00-23:59, ***, and DOWN.

      Start/Stop   Start/Stop   Start/Stop   Start/Stop   Start/Stop
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Weekday 08:30 15:00
Weekend 09:00 18:00
Daily

```

Figure 213. Monitoring Service Period

The migration also links the generated monitoring period (see Figure 213) 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:

```

COMMANDS ACTIONS VIEW HELP
-----
A0FGXC4E          Monitoring period for CICS link          Row 1 to 2 of 2
Command ==>          SCROLL==> PAGE

Entry Type : CICS link          PolicyDB Name : DATABASE_NAME
Entry Name : CICS4AIM4A        Enterprise Name : YOUR_ENTERPRISE

Action      Status      Monitoring period/Short Description
          SELECTED      CICS4AIM4A
                               Migrated from connection CICS4AIM4A
    
```

SA OS/390 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:

```

COMMANDS ACTIONS VIEW HELP
-----
          CICS link for Application          Row 1 to 3 of 3
Command ==>          SCROLL==> CSR

Entry Type : Application        PolicyDB Name : DATABASE_NAME
Entry Name : CICS4A_sysname     Enterprise Name : YOUR_ENTERPRISE

Action      Status      CICS link/Short Description
          Selected      CICS4AIM4A
                               IMS521 (IPUAHI51) on AOC2
***** Bottom of data *****
    
```

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 ==>

Entry Type : Application        PolicyDB Name : DATABASE_NAME
Entry Name : CICS4A_sysname     Enterprise Name : YOUR_ENTERPRISE

Resource:   CICS4A.TRAN
Description:

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

----- Levels -----
Resource      Critical      Frequent      Infrequent
              Number  Interval  Number  Interval  Number  Interval
              (hh:mm)  (hh:mm)  (hh:mm)  (hh:mm)
CICSK4D.TRAN  02    00:05    04    01:00    05    24:00
    
```

SERVICE PERIODS

Migration Functions

```
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 as appropriate as described in the sample itself.

Merge TSCF Configuration

Use option 6 **MERGE** from the panel shown in Figure 5 on page 14 to add an existing TSCF configuration. SA OS/390 will display a panel similar to the following:

```
COMMANDS  HELP  
-----  
AOFGMTSC                                     Merge TSCF configuration  
Command ==>>>  
  
PolicyDB name : DATABASE_NAME               Enterprise name: YOUR_ENTERPRISE  
Target DSN    : 'DOCU.SH.POLICYDB'  
  
TSCF Table Library Name . . . .  
Report. . . . . $TSCFRPT
```

Figure 214. Merge TSCF Configuration Panel

In this panel, you specify the data set to be migrated from TSCF.

TSCF Table Library Name

Fully qualified name of the partitioned data set containing the TSCF configuration to be migrated.

Report

Member name of the migration report, placed in the TARGET data set.

Migration Functions

The COMMANDS choice on the action bar allows you to either enter commands on the command line using the full command name or fast key symbol as described in “Migrate Automation Control File” on page 309.

Messages similar to the ones printed below will be displayed on your screen:

```
-----  
Starting migration process.  
Migrating Processors.  
Migrating Communications Paths.  
Migrating Communications Tasks.  
Migrating Profile Information.  
Migrating Systems.  
Checking for duplicate names.
```

Migration Functions

Chapter 8. Automating Networks

Automation Network Definition Process	327	DSIPARM Definitions Required for Gateway Sessions	333
Defining a Focal Point System	328	Defining Adjacent NetView Paths	334
Defining Gateway Sessions	329	Defining Automatically Initiated TAF Full-Screen Sessions	335
Defining an Outbound Gateway Autotask	330	Defining Automatically Initiated User NNT Sessions	336
Defining Gateway Sessions with Other Systems	330		
Securing Gateway Sessions	332		

Automation Network Definition Process

This section summarizes the steps for defining an automation network to SA OS/390. 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 directly connected systems.
“Defining Gateway Sessions” on page 329 describes how to define gateway sessions.
 - Adjacent NetView path descriptions to allow message forwarding between systems that are not directly connected by a gateway session.
“Defining Adjacent NetView Paths” on page 334 provides details on adjacent NetView path descriptions to allow message forwarding between systems that are not directly connected by a gateway session.
2. Modify NetView DSIPARM data set definition members DSIOPF and DSIDMN to reflect your automation network configuration. The chapter on how to install SA OS/390 on host systems in *System Automation for OS/390 Planning and Installation* provides details.

For an example of the automation network definition process, also refer to the chapter about installing SA OS/390 on host systems in *System Automation for OS/390 Planning and Installation*.

These definitions create a path allowing message forwarding from target systems to the focal point system. If your definitions include adjacent NetView path descriptions, you can also use this path to send messages, commands, and responses between target systems.

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.

Automating Networks

If the message forwarding path is not yet fully implemented on all systems in an automation network, messages are displayed to notification operators on the target systems. Once the message forwarding path is fully implemented, notifications for any system in the network are also 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 a 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 215 shows an example automation network. In this example, the primary focal point system is CHI01. The backup focal point is CHI02.

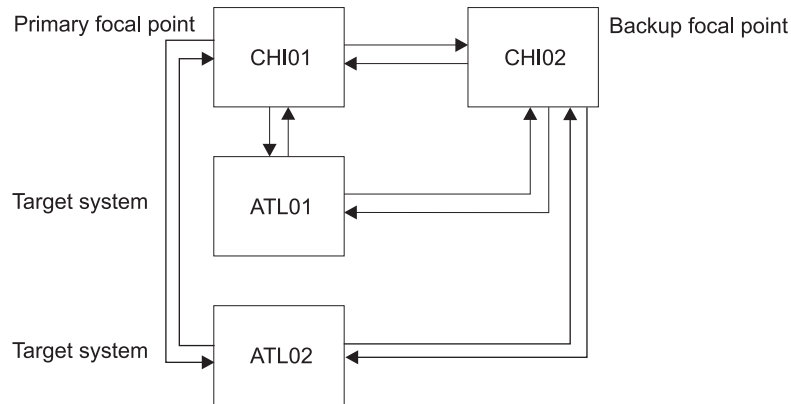


Figure 215. Focal Point Forwarding Definitions for Systems

The corresponding focal point forwarding definitions in the automation policy for systems CHI02, ATL01 and ATL02 are as follows. You reach the required panel selecting the FORWARD policy item of a *Network* entry type.

- CHI02 automation policy:

```

COMMANDS  HELP
-----
AOFINE2          Notification Forwarding
Command ==>>>

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
    
```

- ATL01 automation policy:

```

COMMANDS  HELP
-----
AOFpine2      Notification Forwarding
Command ==>

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

```

- ATL02 and ATL02 automation policy:

```

COMMANDS  HELP
-----
AOFpine2      Notification Forwarding
Command ==>

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

```

In Figure 215 on page 328, the automation policies for system CHI01 do not have any forward focal point definitions. This is because SA OS/390 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” on page 330 for details.
2. On the SA OS/390 Network policy object, use the GATEWAY policy item to define the destination systems to which the originating system connects, and the password used to log on inbound gateway autotasks from the originating system at destination systems. Coordinate the password values here with passwords in DSIOPF entries for gateway autotask operator IDs on all interconnected systems. See “Defining Gateway Sessions with Other Systems” on page 330 and the chapter on how to install SA OS/390 on host systems in *System Automation for OS/390 Planning and Installation* for details.

Optionally, use the SA OS/390 password protection feature to generate and maintain gateway autotask passwords. See “Securing Gateway Sessions” on page 332 for details.

3. Define operator IDs used for all inbound and outbound gateway autotasks used on the system in the NetView DSIPARM data set member DSIOPF. Coordinate the password values on these DSIOPF entries with the passwords specified on the GATEWAY policy item. See “Defining Gateway Sessions with

Automating Networks

Other Systems” and the chapter on how to install SA OS/390 on host systems in *System Automation for OS/390 Planning and Installation* for details.

4. Adjust values in NetView DSIPARM data set member DSIDMN as needed to accommodate gateway autotask definitions. See *System Automation for OS/390 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. You do not have to specify any messages for GATOPER. The *Automation Operator NetView Userids* panel is displayed automatically.

If GATOPER has previously been defined, then select it by entering an **S** in the ACTION column.

Enter the NetView operator ID 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 OS/390 Planning and Installation*, the following values are specified:

```
COMMANDS  HELP
-----
AOFPIA01      Automation Operator NetView Userids
Command ==>>

Entry Type : Auto Operators      PolicyDB Name  : DATABASE_NAME
Entry Name  : CHI010PS           Enterprise Name : YOUR_ENTERPRISE

Automated Function: GATOPER
Messages assigned:

Enter automation operators and NetView operator(s) to receive messages.

Automation Operators      NetView Operators
Primary ==> GAT&DOMAIN.   ==>
Backup  ==>                ==>
                               ==>
                               ==>
                               ==>
```

Defining Gateway Sessions with Other Systems

The next step in defining gateway sessions is done using the GATEWAY policy item of the Network policy object. Using this policy item you define:

- Other systems to which a system connects. These systems are identified in the DOMAIN field.
- The passwords used to log on an incoming gateway autotask from the originating system at a destination system. These passwords are specified in the PASSWORD field.

You must coordinate password values on this panel with the password values for incoming gateway autotask operator IDs in the DSIOPF members on the

destination systems. (DSIOPF member entries are discussed in more detail in the chapter on how to install SA OS/390 on host systems in *System Automation for OS/390 Planning and Installation*.)

If you use the SA OS/390 password protection feature, in which SA OS/390 retrieves the password for the gateway autotask from a specialized SA OS/390 password data set, specify RACFNNT, or MAXIMUM, in the Password field.

If you use the SA OS/390 password protection feature to generate the passwords, you do not have to coordinate the panel and DSIOPF password values. However, there are activities you must perform to set up and use the SA OS/390 password protection feature. For descriptions of these activities, and descriptions of the OWNER and SHARE fields on the Gateway Definitions panel, see “Securing Gateway Sessions” on page 332.

Following is an example of defining gateway sessions using hardcoded passwords for the gateway autotasks. Refer to the chapter about installing SA OS/390 on host systems in *System Automation for OS/390 Planning and Installation* to see how the SA OS/390 password protection feature affects password values specified on panels and in DSIOPF entries.

Suppose the gateway autotasks connecting systems shown in *System Automation for OS/390 Planning and Installation* use hardcoded passwords instead of ones supplied by the SA OS/390 password protection feature. The gateway definitions for system CHI01 specify connections to two systems, ATL01 and ATL02. The password values for gateway definitions in the CHI01 automation policy must be coordinated with password values in the DSIOPF members on ATL01 and ATL02. These passwords are the ones used to log inbound gateway autotask GATCHI01 on systems ATL01 and ATL02. The *GATEWAY Definitions* panel for system CHI01 looks like this:

```

COMMANDS  HELP
-----
AOFPIE4          GATEWAY Definitions
Command ==>>>                                SCROLL==>>> PAGE

Entry Type : Network          PolicyDB Name : DATABASE_NAME
Entry Name : NETWORK_NAME     Enterprise Name : YOUR_ENTERPRISE

Enter the following information for each NetView domain that commands or
responses are forwarded to. Use RACFNNT as the password if SA OS/390 is
to retrieve the password from RACF.

Domain Password Logmode Description
ATL01 LILCITY AOF00001 Atlanta System 1 _____
ATL02 BOBDOBBS AOF00002 Atlanta System 2 _____
_____
_____
_____

```

Figure 216. GATEWAY Definitions Panel

The passwords on the *GATEWAY Definitions* panel must match the password for the incoming gateway autotask operator ID GATCHI01 in the DSIOPF entries on systems ATL01 and ATL02.

In system ATL01, the DSIOPF entry for GATCHI01 looks like this:

Automating Networks

```
GATCHI01 OPERATOR PASSWORD=LILCITY
          PROFILEN AOFPRFAO
:
```

In system ATL02, the DSIOPF entry for GATCHI01 looks like this:

```
GATCHI01 OPERATOR PASSWORD=BOBDOBBS
          PROFILEN AOFPRFAO
:
```

Securing Gateway Sessions

Gateway sessions are initiated by gateway autotasks automatically logging on to other systems. You may want to secure gateway sessions by using the SA OS/390 password protection feature on gateway autotask passwords.

The SA OS/390 password protection feature does the following:

- Generates and maintains a random 8-character password for the gateway autotask.
- Changes the password automatically every 30 days.
- Stores the password in an encrypted format in a VSAM data set called the *SA OS/390 password data set*. This is a separate data set from the SAF data set and is separately allocated and installed during SA OS/390 installation. The SAF database contains entries for all user IDs and their passwords for a particular system, including user IDs for gateway operators *inbound* from other systems. In contrast, the SA OS/390 password data set contains user IDs and passwords for gateway operator IDs to log on to other systems in the automation network. That is, it contains user IDs and passwords for gateway operator IDs that are *outbound* from the originating system and, in turn, *inbound* on the destination systems. When gateway sessions are activated, this SA OS/390 password data set is accessed to retrieve passwords for logging on gateway operators for the originating system at the destination systems.

When you use the SA OS/390 password protection feature instead of hardcoding passwords, you do not have to coordinate specific password values between the customization dialog and DSIOPF entries.

Installing the SA OS/390 Password Protection Feature: To use the SA OS/390 password protection feature for SA OS/390 gateway autotasks, you must install it, then use the INGPW command, described in *System Automation for OS/390 Programmer's Reference* to initialize it. Because the INGPW command retrieves passwords from the SA OS/390 password data set, use scope checking on the INGPW command to set limitations on which users can issue it. Installing the SA OS/390 password protection feature involves communicating with the MVS security administrator to obtain passwords for the gateway operator IDs. Further details on installing the SA OS/390 password protection retrieval feature are in *System Automation for OS/390 Planning and Installation*.

Managing Passwords If a RACF Data Set Is Shared: When systems in an automation network share a RACF data set, you may have to specify additional information using the SAF Environment policy item of the Network policy object. You need to specify this additional information when the following conditions are met:

1. The system has gateway operators log on to two or more other systems.
2. The systems onto which the originating system logs on share the same RACF data set.

When these two conditions are met, you need to use the OWNER and SHARE fields on the *SAF Environment Definition* panel.

```

COMMANDS  ACTIONS  HELP
-----
AOFPIE6          SAF Environment Definition          SCROLL==> PAGE
Command ==>>>

Entry Type : Network          PolicyDB Name  : DATABASE_NAME
Entry Name  : NETWORK_NAME    Enterprise Name: YOUR_ENTERPRISE

If a standard password format has been established, enter the corresponding
Password Generation Mask. . . _____

If you have NetView domains that share the same SA OS/390 password data set,
enter the following information.
Actions: M = Move  B = Before  A = After  I = Insert

Action  Owner  Group  Share
-----  -
_____ IPSNM  ___   IPSNN,IPSNO,IPSNP,IPVNH,IPUNA,IPUNB,IPUNC,IPUND,IPUNM
_____  ___   ___   IPUN8,IPUN9_____
_____  ___   ___   _____
_____  ___   ___   _____

```

Figure 217. SAF Environment Definition Panel

The fields shown above are:

Owner

Enter the domain ID, or system name, of the originating system responsible for logging on to the two or more systems that share the same RACF data set in the OWNER field.

The OWNER value must match the *domain_name* parameter value on the INGPW command issued from the originating system. The OWNER value, in combination with the *gateway_id* parameter value on the INGPW command, serves as a key to the appropriate operator ID entry in the SA OS/390 password data set on the originating system.

Group If multiple shared RACF databases are in use, the *group_id* for each shared RACF database may also be required.

Share Enter the names of the destination systems that share the same RACF data set in the SHARE fields.

For an example of an automation network with systems sharing a RACF data set, refer to the chapter about installing SA OS/390 on host systems in *System Automation for OS/390 Planning and Installation*.

Managing Passwords If SAF Is Not Used: If your system does not use the SA OS/390 password protection feature, you must hard-code passwords for gateway operator IDs in the Password field on the Gateway Definitions panel of the customization dialog. Using hardcoded passwords requires you to coordinate passwords specified in the customization dialog with passwords specified in DSIOPF member entries on other systems. See “Defining Gateway Sessions with Other Systems” on page 330 for an example.

DSIPARM Definitions Required for Gateway Sessions

To complete defining a gateway session, you must add or modify definitions in NetView DSIPARM data set members DSIOPF and DSIDMN. This task is done

Automating Networks

outside of the customization dialog. The chapter on how to install SA OS/390 on host systems in *System Automation for OS/390 Planning and Installation* provides details.

Defining Adjacent NetView Paths

Adjacent NetView paths are defined using the ADJACENT NetView policy item of the *Network* policy object. The definition specifies:

- The destination system to which this system will route commands. This value is specified in the *To Domain* field.
- The system used as a path to the To Domain. This value is specified in the *Adjacent Domain* field.
- The system used as adjacent domain when the gateway of the actual adjacent domain is not active.
- An optional description of the destination system, or of how the adjacent NetView system is used as a path to the destination system. This value is specified in the *To Domain Description* field.

Note: The adjacent NetView path capability is limited to passing through only one adjacent system on the way to the destination system. You cannot define a path consisting of multiple intervening systems between originating and destination systems.

For example, in Figure 218, to route commands and responses between systems ATL02 and ATL04, the automation policy for both systems must specify adjacent NetView path descriptions identifying the gateway session with system ATL03 as a path to the destination system.

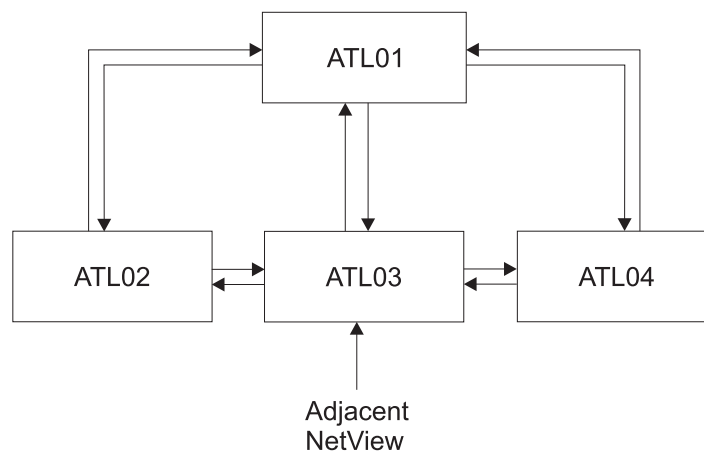


Figure 218. Example Adjacent NetView Path

You reach the panel to define adjacent NetView path using the ADJACENT NETVIEW policy item of a *Network* policy object. In system ATL02, the automation policy values defining an adjacent NetView path are:


```

COMMANDS  HELP
-----
AOFpine1          Adjacent Netview
Command ==>>>                                SCROLL==>> PAGE

Entry Type : Network          PolicyDB Name : DATABASE_NAME
Entry Name : FOCAL_NETWORK    Enterprise Name : YOUR_ENTERPRISE

Enter the following information for automated message forwarding.

  To      Adjacent  Alternate  To-Domain Description
  Domain  Domain    Adjacent
-----
ATL04     ATL03     ATL01     Path to ATL04
-----
-----
-----

```

In system ATL04, the automation policy values defining an adjacent NetView path are:

```

COMMANDS  HELP
-----
AOFpine1          Adjacent NetView
Command ==>>>                                SCROLL==>> PAGE

Entry Type : Network          PolicyDB Name : DATABASE_NAME
Entry Name : FOCAL_NETWORK    Enterprise Name : YOUR_ENTERPRISE

Enter the following information for automated message forwarding.

  To      Adjacent  Alternate  To-Domain Description
  Domain  Domain    Adjacent
-----
ATL02     ATL03     ATL01     Path to ATL02
-----
-----
-----

```

Defining Automatically Initiated TAF Full-Screen Sessions

You can automatically establish Terminal Access Facility (TAF) full-screen sessions for applications that SA OS/390 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 OS/390 Customization Guide*

Once TAF full-screen sessions are set up, they can be managed using the TAF Fullscreen Menu in the SA OS/390 Operator Interface. See *System Automation for OS/390 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:

Automating Networks

- 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.
- 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
-----
AOFPI3E3      Fullscreen TAF Application Definition
Command ==>>                                SCROLL==>> PAGE

Entry Type : Network          PolicyDB Name  : DATABASE_NAME
Entry Name  : FOCAL_NETWORK   Enterprise Name: YOUR_ENTERPRISE

Enter the applications with which SA OS/390 operators can establish TAF
sessions automatically using the operator interface.

Session Name  Application ID  System
  TSO          TAIN1        CHI01
  _____  _____  _____
  _____  _____  _____
  _____  _____  _____
```

Defining Automatically Initiated User NNT Sessions

A *user NNT session* is a dedicated NetView-NetView task session for one operator. It is an additional type of connection beyond gateway autotasks established between systems. Individual operators can use these NNT sessions for routing commands and responses between systems.

User NNT sessions are useful for routing commands and responses to systems running versions of AOC/MVS earlier than Release 3. It is recommended that you use RMTCMD-based command routing for routing commands between AOC/MVS Release 3 or later systems or between SA OS/390 systems.

User NNT sessions can be defined by selecting the NNT Sessions policy object from the *Entry Type Selection* panel of the customization dialog. You can define:

- The NetView users who will have user NNT sessions established
- The systems with which they will communicate
- Whether you want the user NNT session automatically initiated, or started using the Cross Domain Entry Menu of the SA OS/390 operator dialogs

More information on editing the DSICMD member can be found in the *Tivoli NetView for OS/390 Installation and Administration Guide*

When user NNT sessions are defined, they can be managed using the Cross Domain Entry Menu in the SA OS/390 operator dialogs. See *System Automation for OS/390 User's Guide* for more information on managing user NNT sessions.

To define a user NNT session on the *NNT Sessions Definition* panel, specify the following:

- The user ID for which the user NNT session is to be established. This user ID must be the same on both systems.
- The system with which the user NNT session is to be established (the Domain field).
- The password used for logging the user on to the other system. You can also specify the keyword PROMPT, which causes an SA OS/390 logon panel to be displayed, where the user can key in the password.
- Whether the user NNT session should be automatically initialized when the user logs on to the current system or started on the Cross Domain Entry Menu in the operator dialogs
- A description of the system with which the user NNT session is to be established. This is an information-only field, and is displayed in the Description field on the Cross Domain Entry Menu operator panel.

For example, the following panels establish a user NNT session allowing user ID USER999 to log on to system CHI01 from another system (the user ID must be USER999 on both system CHI01 and the other system). The user is prompted to enter the logon password, and the user NNT session is automatically initialized.

```

COMMANDS  ACTIONS  HELP
-----
AOFGNNOP                                NNT Sessions Definition
Command ==>>>                                Scroll ==>> PAGE

Entry Type : Network                    PolicyDB Name : DATABASE_NAME
Entry Name : FOCAL_NETWORK              Enterprise Name : YOUR_ENTERPRISE

Enter the Userid used to establish NNT sessions.

Action  Userid  Description
      USER999  An example NNT Session
  
```

```

COMMANDS  HELP
-----
AOFGNNDM                                NNT Sessions Definition
Command ==>>>                                Scroll ==>> PAGE

Entry Type : Network                    PolicyDB Name : DATABASE_NAME
Entry Name : FOCAL_NETWORK              Enterprise Name : YOUR_ENTERPRISE

UserId : USER999

Enter the NetView domains used to establish NNT sessions.
Password: PROMPT or Password
Initialize: Yes =Initiated automatically
           No  =Initiated thru operator interface.

Domain  Password  Initialize  Description
CHI01   PROMPT     YES        NetView CHI01 NNT_____
          _____
          _____
          _____
  
```

Chapter 9. How System Operations Coordinates with Automatic Restart Manager

SA OS/390 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 OS/390 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 OS/390 fallback capabilities for applications running on sysplexes. SA OS/390 continues to automate an application after it has been moved to a fallback system, provided SA OS/390 is installed on that system. If it is not installed on the fallback system, SA OS/390 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 OS/390 resolves Automatic Restart Manager statuses to SA OS/390 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 OS/390, see "Application Policy Object" on page 141.

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.ARM* can be used to tailor automation behaviour during ARM restart processing. As an example, a *subsystem.ARM* minor resource could be specified with a RESTART EXIT enabled to drive a user supplied

How System Operations Coordinates with Automatic Restart Manager

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 OS/390 will not allow ARM to attempt a restart of the application.

Rather than use the *subsystem.ARM* 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.

For more information on how to define MOVE groups refer to "Creating a New ApplicationGroup" on page 133.

Chapter 10. Conversion Function

This chapter provides information on what you should be aware of before using the new release of SA OS/390. In detail, the following topics are discussed:

- “Deleted Entry Types and Policy Items”
- “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.

Deleted Entry Types and Policy Items

Note that the following entry types available in earlier SA OS/390 releases have been removed from SA OS/390 2.1:

- *OperatorView*
- *Volume*
- *VolumeGroup*

As a consequence, the related policy items in other entry types have been removed, like for example, the policy item *USED IN VIEWS*.

Furthermore, according to the new design of SA OS/390 2.1, the following policy items have been removed from the related entry types:

- policy items related to I/O exception monitoring
- policy items related to enterprise monitoring
- policy items related to RODM load
- policy items related to monitoring via RMF data

The following policy objects required for RODM have also been removed:

- For GRP, SBG, SYS APG and APL: policy object DEFAULTS
- For APG and APL additionally: policy object SYSTEM
- For APL: policy objects *SYSTEM, *BATCH, *TSO, *STC, *MASTER* and all APLs with generic names

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 is a special conversion option available via selection of *0 Settings* on the *Customization Dialog Primary Menu* and *4 CONVERT* on the subsequent *PDB Convert Option* panel shown in Figure 219 on page 342.

This option takes effect for the initial conversion only. Subsequent conversions which might be triggered due to usage of the MIGRATE function do not accept this option.

Note: This conversion option should be set to YES. Otherwise SA OS/390 cannot generate resources for these entries. The default value is NO.

Conversion Function

On this panel, SA OS/390 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 OS/390 can generate resources from such entries.

```
COMMANDS  HELP
-----
AOFGCOPT          PDB Convert Option
Command ==>>>

Specify whether the following data should be set up during initial PDB convert.
This option will be saved in your ISPF profile pool and will be used for the
initial converts on all Policy Databases until it is set again.

APG Automation Name. . . YES          Set up name during convert (YES NO)
```

Figure 219. PDB Convert Option Panel

In the field *APG Automation Name* you specify whether an automation name should be set up during the initial PDB conversion. YES indicates that the automation name for each APG entry in the PDB will be set up with the entry name of the APG entry if it is less than or equal 11 characters. If the entry name has more than 11 characters in length, the automation name will be forced to blank.

NO indicates that the automation name for each APG entry in the policy database will be filled with blank.

Converted Data

During initial conversion the following data will be converted:

- The PARENTS of all applications will be converted to HASPARENT 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 definition plus special satisfy conditions of the created HASPARENT relationships
- The SHUTOPTIONS definitions of all applications will be converted to EXTERNAL SHUTDOWN definition plus special conditions specified in the new HASPARENT relationships

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 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 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 OS/390 2.1:

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)
```

- 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

Conversion Function

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 *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.
- Resync of ACF prefix value for each entryname in the PDB with table AOFTEP (formally done during each ACF build).
- With 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.

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:

COMMANDS	ACTIONS	VIEW	HELP

AOFGXREL	Relationship Selection List		
Command ==>			
Entry Type : Application	PolicyDB Name : DATABASE_NAME		
Entry Name : APPC	Enterprise Name : YOUR_ENTERPRISE		
Action #	Type	Supporting Resource	Auto Chain
_____ _1	HASPARENT	SUBSYS1/APL/=	
		StartsMe	
_____ _2	HASPARENT	SUBSYS2/APL/=	
		StartsMeAndStopsMe	
_____ _3	HASPARENT	SUBSYS3/APL/=	
		StartsMeAndStopsMe	

Cross Reference of SA OS/390 1.3 Feature Keywords to SA OS/390 2.1 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* pairs via the customization dialog and had SA OS/390 build the ACF.

Now whatever method you used to use, you either need to convert your old UET entries to the new SA OS/390 2.1 customization dialog entries and thus to the SA OS/390 2.1 policy database format or migrate your old ACF to the SA OS/390 2.1 format. Both these functions are now done by SA OS/390 2.1.

All feature keyword conversions either take place during the Conversion function, when a policy database in SA OS/390 1.3 format is opened or in the Migration functions, when an ACF is migrated (and type PRD is selected). Table 13 contains details of all the entries that are migrated/converted for Product Automation.

Some explanations:

Format change

means the data entry format is slightly modified from the original ACF entries, refer to the Product Automation dialog documentation.

New means that a new dialog function has been implemented for this definition.

Obsolete

means that the original entry type no longer exists.

Table 13. Cross Reference of SA OS/390 1.3 Feature Keywords to SA OS/390 2.1 Dialog Entries

Keyword	SA OS/390 2.1 Migration/Conversion Target
CICS Keywords , where Application Messages is stated, this refers to the MESSAGES/USER DATA policy item of the appropriate CICS subsystem definition.	
ACORESTART	Application Messages
Automation Flags	Application Automation info
AUTOOPS	Auto Operators (not in CICSCNTL any more)
SDF	SDF
ENVIRON SETUP CICS	Obsolete
ENVIRON TIMEOUT	TIMEOUT
SHUTDOWN	Application
STARTUP	Application
SUBSYSTEM	Application (Application type = CICS)
THRESHOLDS	New <i>Application</i> policy item (CICS) RESOURCE THRESHOLDS
ABCODESYSTM	Application Messages
ABCODETRAN	Format change, Application Messages
AREA	Obsolete, new CICS State/ Action, link to APL
CICSCNTL	New <i>Application</i> policy item CICS CONTROL
CICSGROUP	Obsolete, handled in base code
CONNECTION	New define connection, link to APL
EXTCOND	Migrated to entry type <i>Events</i>

Conversion Function

Table 13. Cross Reference of SA OS/390 1.3 Feature Keywords to SA OS/390 2.1 Dialog Entries (continued)

Keyword	SA OS/390 2.1 Migration/Conversion Target
HEALTHCHK	Format change, Application Messages
LISTSHUT	Format change, Application Messages
PRODUCT	Obsolete, new CICS State/ Action, link to APL
RCVRAUTO	Format change, Application Messages
RCVRSOS	Format change, Application Messages
RCVRTRAN	Format change, Application Messages
RCVRVIOL	Format change, Application Messages
SERVICE	Migrated to entry type <i>Service Periods</i>
TRIGGER	Migrated to entry type <i>Triggers</i>
IMS Keywords , where Application Messages is stated, this refers to the MESSAGES/USER DATA policy item of the appropriate IMS subsystem definition.	
ABCODEPROG	Format change, Application Messages
ABCODES, ACTCODES, ALTCODES	Application Messages
ABCODETRAN	Format change, Application Messages
ACORESTART	Application Messages
AREA	Obsolete, new IMS State/ Action, link to APL
AUTOOPS	Auto Operators (not in IMSCNTL any more)
BRO	Application Messages
CHE	Application Messages
CQS0031A	Application Messages
CQS0032A	Application Messages
CQS0033A	Application Messages
CQSET	Application Messages
DFS2142	Application Messages
DFS2161I	Application Messages
DFS2169I	Application Messages
DFS3258A	Application Messages
DFS3869A/DFS3869I	Application Messages
DFS554A	Application Messages
DFS810A	Application Messages
DFS989I	Application Messages
DFS994I	Application Messages
ENVIRON	New <i>Application</i> policy item IMS ENVIRON
ENVIRON SETUP	Obsolete
EXTCOND	Migrated to entry type <i>Events</i>
FORCE	Application Messages
HOLDQ	Application Messages
IMSCNTL	New <i>Application</i> policy item IMS CONTROL
IMS CRITMSG	Obsolete

Table 13. Cross Reference of SA OS/390 1.3 Feature Keywords to SA OS/390 2.1 Dialog Entries (continued)

Keyword	SA OS/390 2.1 Migration/Conversion Target
IMSGROUP	no longer used
IMSRSENM	New, IMS RESOURCE NAME Policy
INITSTART	Application Messages
IPS	Application Messages
NTFYOP	Application Messages
OLDS	Application Messages
POSTCHKP	Application Messages
PRECHKP	Application Messages
PRODUCT	Obsolete, new IMS State/Action, link to APL
RECONS	Application Messages
RECOVERY	Application Messages
RELEASEQ	Application Messages
RESTART	Application Messages
RESTARTABORT	Application Messages
SDF	SDF policies
SERVICE	Migrated to entry type <i>Service Periods</i>
SHUTINIT	Application Messages
SHUTNORM	Application Messages
SHUTIMMED	Application Messages
SHUTFORCE	Application Messages
SHUTYPES	Application Messages
SNAPQ	Application Messages
START	Application Messages
STARTUP	Application Messages
STOPBMPREGION	Application Messages
STOPFPREGION	Application Messages
STOPREGION	Application Messages
STSFILE	New, ISF entry type, policy object IMS status file, policy item IMS STATUS FILE
SUBSYSTEM	Application Policy (TYPE=IMS)
TCO	Application Messages
TCOMEMBERS	Application Messages
THRESHOLDS	New IMS resource thresholds
TPABEND	Application Messages
TRIGGER	Migrated to entry type <i>Triggers</i>
UNLKAVM	Application Messages
UNLOCK	Application Messages
UP	Application Messages
USERSTART	Application Messages

Conversion Function

Table 13. Cross Reference of SA OS/390 1.3 Feature Keywords to SA OS/390 2.1 Dialog Entries (continued)

Keyword	SA OS/390 2.1 Migration/Conversion Target
VTAMTERMS	Application Messages
%INCLUDE	Obsolete (no longer permitted)
\$PI	Application Messages
\$SI	Application Messages
\$TI	Application Messages
OPC Keywords , where Application Messages is stated, this refers to the MESSAGES/USER DATA policy item of the appropriate OPC subsystem definition.	
OPCA (CODE)	Application Messages
OPCACMD	Application Messages
OPCA DOMAINID	New, WORKSTATION DOMAINS policy
ENVIRON OPCA0	New, OPC SYSTEM DETAILS policy
OPCAPARM	Application Messages
OPCA PCS	New, CONTROLLER DETAILS policy
OPCA OPCSRSTAT	New, SPECIAL RESOURCES policy linked to CONTROLLER DETAILS
OPCCNTL	New, Application OPC CONTROL policy

Selective Policy Database Conversion

After entries have been migrated into the policy database, the convert is done to convert the old data which might have been migrated into the policy database into the new format. The migrate functions automatically calls the convert function.

Note: The Convert function, after it is called after a migration, will not call the product automation conversion as this has already been intercepted and performed by the migration.

Appendix A. Concurrent Customization by Multiple Users

Note:
Concurrent user access is supported for *Application* and *ApplicationGroup*.

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* and *Application* 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* or *ApplicationGroup* with several users, or performing commands other than **OPEN** or **SELECT** against a policy database while an application or an application group is 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 220 on page 352), 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 OS/390 policy items for modification, this will prevent other users from accessing any policy item and an appropriate message is displayed.

Policy Data Base Selection

If you want to issue a **BUILD**, **DELETE**, **MIGRATE**, or **REPORT** command on the *Policy Data Base Selection* panel (AOFGPDB), while another user has accessed the same policy database, the command is rejected and an appropriate message is displayed.

Concurrent Customization by Multiple Users

On the other hand, 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 (with **S (Select)** or **O (Open)**) successfully, the *Entry Type Selection* panel (AOFGETYP) 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* or *Application* is selected, and another user 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* and *ApplicationGroups*.

Entry Name Selection

On the *Entry Name Selection* panel (AOFGENAM) a specific application can be selected, and a **BUILD** can be requested.

If you try a **BUILD** action, and the policy database is 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, which means, changing the displayed data is not allowed. In this case, however, it is possible to select that application and to look at the data. A warning message is displayed on the *Policy Selection* panel AOFGEPOL, 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.

You can only add or remove such links, if you exclusively access the entry types *Application* or *ApplicationGroup*. 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 or application group to a trigger.

Appendix B. More about Using the Customization Dialog

Getting Help	351	User-written Functions	360
Changing Your Customization Dialog Environment	351	Alternate Configuration Support	361
Extending Policy Definitions	353	How to Define Alternate Configurations for	
Defining Data Sets for Batch Processing	356	Systems	362
Define ISPF Temporary Data Sets for Batch		How to Build Alternate Configurations.	362
Build	357	Restrictions	362
Creating Reports about Policy Databases	358		

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. This displays help text for that panel.

When selecting **HELP** from the action bar, the pull-down choice contains two new selections:

- **Automation Help:** This help provides on-line 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 SA OS/390 2.1.

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 (Figure 220 on page 352) allow you to change the customization dialog environment for your user ID. This is a usability function and has no effect on the policy database contents.

More about Using the Customization Dialog

```
MENU  HELP
-----
AOFGCUST          Customization Dialog Settings Menu
Option ==>>

The following options apply only to your userid:

  1 ALIAS          Define alias for standard commands      ( AOFALIAS )
  2 DISPLAY        Define display characteristics        ( AOFDISPL )
  3 MESSAGE        Define message and logging options    ( AOFMSG   )
  4 CONVERT        Define PDB conversion options

The following options apply to all SA OS/390 users:

 10 POLICY        Modify SA OS/390 policy definitions    ( AOFPOLCY )
 11 DSNAME        Define SA OS/390 data set names        ( AOFDSNAM )
 12 WORKDSN       Define ISPF temporary data sets for batch build

F1=HELP  F2=SPLIT  F3=END   F4=RETURN  F5=RFIND  F6=RCHANGE
F7=UP    F8=DOWN   F9=SWAP  F10=LEFT  F11=RIGHT F12=RETRIEVE
```

Figure 220. SA OS/390 Customization Dialog Settings Menu

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 356.

CONVERT

This option lets you specify whether SA OS/390 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 341 for more information.

POLICY

see “Extending Policy Definitions” on page 353;

DSNAMES

see “Defining Data Sets for Batch Processing” on page 356;

WORKDSN

see “Define ISPF Temporary Data Sets for Batch Build” on page 357;

Extending Policy Definitions

To create new or modify existing policy items:

1. Choose option **10 POLICY** on the *Customization Dialog Settings Menu* (Figure 220 on page 352). The *Policy Definition Entry Selection* panel is displayed.

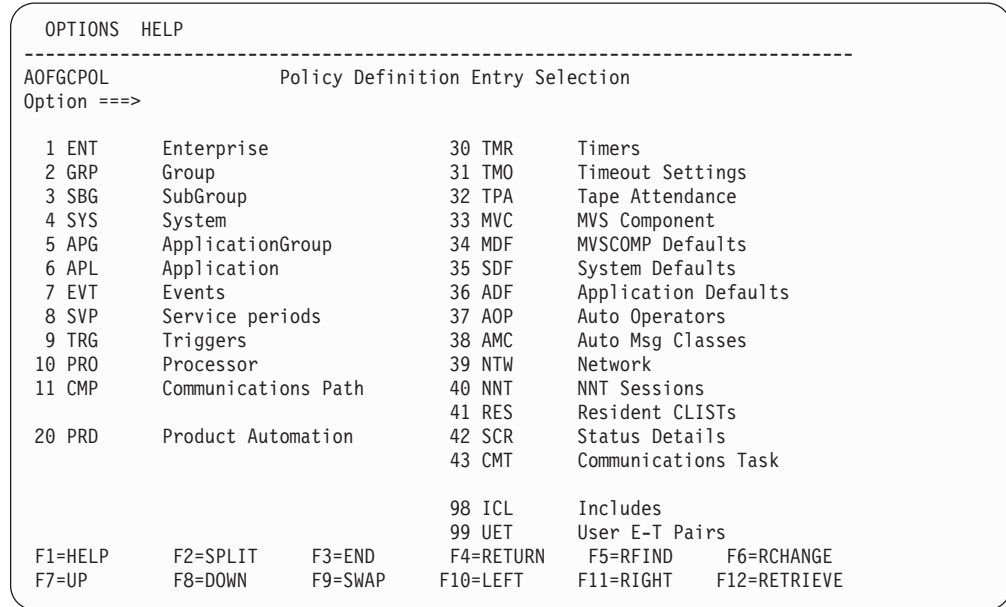


Figure 221. Policy Definition Entry Selection Panel

2. Select the entry type which you wish to deal with. The *Edit Policy Selection* panel is displayed, listing all current policy items for this entry type.
3. From the *Edit Policy Selection* panel you can:
 - Edit an existing policy item (not recommended)
 - Delete a policy item (not recommended)
 - Insert a new policy item
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. In the example from Figure 222 on page 354, you want to edit policy item GROUP INFO from entry type *Group*.

More about Using the Customization Dialog

```

COMMANDS  HELP
-----
AOFGCP02          Define Policy Selection
Command ==>>>

Entry Type : Group

Enter or update the following to define the Policy Definition:

Name . . . . . GROUP INFO
Description. . Display Group information
Help Action. . PGM(ISPTUTOR) PARM(AOFHHG03)
Edit Action. . CMD(AOFF#POL AOFGGPA0,GRP)
User Note. . .
              . . .
              . . .
              . . .
              . . .

Filter attributes . .

F1=HELP      F2=SPLIT      F3=END        F4=RETURN     F5=RFIND      F6=RCHANGE
F7=UP        F8=DOWN        F9=SWAP      F10=LEFT     F11=RIGHT    F12=RETRIEVE

```

Figure 222. 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 create your own ISPF input panel for the new policy item.

Note:

Any modifications to the existing policy definitions are solely your responsibility.

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 OS/390 and differ based on the entry type, as follows:

ENTRY TYPE	ATTRIBUTE	VALUE
Group	Group type	(blank)
		SYSPLEX
		MVS
System	Operating System type	(blank)
		MVSSNMP
		NONSNMP
ApplicationGroup	ApplicationGroup type	(blank)

More about Using the Customization Dialog

ENTRY TYPE	ATTRIBUTE	VALUE
		SYSPLEX
		SYSTEM
Application	Object type	CLASS
		INSTANCE
	Application type	(blank)
		JES2
		JES3
		CICS
		IMS
		OPC
		DB2
Processor	Processor type	(blank)
		SCREEN
		OCF
Communications Path	Path type	(blank)
		PS2
		NVC
Others	none	

The following rules apply to the filtering attributes:

1. If no attributes are specified (indicated by "(blank)" in the table above), 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.

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.

JES2 JES3

This never displays 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.

The following COMMANDS are supported:

COPY (P)

Saves the data on this panel into a clipboard

PASTE (T)

Copies the data from the clipboard to the policy being defined

More about Using the Customization Dialog

Defining Data Sets for Batch Processing

Use Option 11 on the *Customization Dialog Settings Menu* panel to define the batch processing data sets. This needs to be done only once, when you install SA OS/390. The details of how to do this are given in the *System Automation for OS/390 Planning and Installation*.

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.

On the *Data Set Customization Details* panel, a table is displayed with groups of

COMMANDS		HELP			

AOFGCUSD		Data Set Customization Details			
Command ==>>				SCROLL==> PAGE	
DDname	Seq	Bas	Data set name	Volume	Unit
AOFEXEC	98	YES	'ING.SINGIREX'	_____	_____
AOFEXEC	99	YES	'SYS1.ISPF.ISPEXEC'	_____	_____
AOFIPDB	99	YES	'ING.SINGIPDB'	_____	_____
AOFLLIB	98	YES	'ING.SINGIMOD'	_____	_____
AOFLLIB	99	YES	'SYS1.ISPF.ISPLOAD'	_____	_____
AOFMLIB	98	YES	'ING.SINGIMSG'	_____	_____
AOFMLIB	99	YES	'SYS1.ISPF.ISPMENU'	_____	_____
AOFPLIB	98	YES	'ING.SINGIPNL'	_____	_____
AOFPLIB	99	YES	'SYS1.ISPF.ISPPENU'	_____	_____
AOFSLIB	98	YES	'ING.SINGISKL'	_____	_____
AOFSLIB	99	YES	'SYS1.ISPF.ISPSLIB'	_____	_____
AOFSTEP	99	YES	'ING.SINGIMOD'	_____	_____
AOFTLIB	96	NO	'HART.ISPF.ISPPROF'	_____	_____
AOFTLIB	97	NO	'HART.USER.SINGITBL'	_____	_____
AOFTLIB	98	YES	'ING.SINGITBL'	_____	_____
AOFTLIB	99	YES	'SYS1.ISPF.ISPTENU'	_____	_____
F1=HELP	F2=SPLIT	F3=END	F4=RETURN	F5=RFIND	
F7=UP	F8=DOWN	F9=SWAP	F10=LEFT	F11=RIGHT	

Figure 223. SA OS/390 Data Set Customization Details Panel

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.

More about Using the Customization Dialog

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 OS/390 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 223 on page 356, these are:

```
'HART.ISPF.ISPPROF'  
'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 OS/390 displays the following panel:

```
-----  
MENU  OPTIONS  HELP  
-----  
AOFGWKDS      ISPF Temporary Data Sets Preallocation for Batch  
Command ==>>>  
  
More:  +  
Specify ISPF work data sets, used by ISPF for processing file tailoring  
services during batch build.  
  
//ISPWRK1 DD DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(5,10)),  
//          DCB=(LRECL=256,RECFM=FB)  
//ISPWRK2 DD DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(5,10)),  
//          DCB=(LRECL=256,RECFM=FB)  
//ISPWRK3 DD DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(5,10)),  
//          DCB=(LRECL=256,RECFM=FB)  
//ISPWRK4 DD DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(5,10)),  
//          DCB=(LRECL=256,RECFM=FB)  
//ISPCTL1 DD DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(5,10)),  
//          DCB=(LRECL=80,RECFM=FB)  
  
F1=HELP      F2=SPLIT      F3=END        F4=RETURN     F5=RFIND     F6=RCHANGE  
F7=UP        F8=DOWN        F9=SWAP      F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

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.

More about Using the Customization Dialog

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.

Creating Reports about Policy Databases

If you select option 3 *Report* from the *Customization Dialog Primary Menu* shown in “How to Start the Customization Dialog” on page 14, you reach the following panel from which you can create reports about various SA OS/390 objects.

```
MENU  HELP
-----
A0FGREPK                      Report Selection Menu
Option ==>>

 1 Report                      Create Policy Data Base Report
 2 Unlinked                    Create Report for entries that are not referenced
 3 ViewUnlinked                View Unlinked Entries Report
 4 Resources                    Create List of generated Resources
 5 ViewResources                View List of generated Resources
```

Figure 224. SA OS/390 Report Selection Menu

Select from the following alternatives:

1 Report

With this option, you can create a report which includes all automation data from the selected policy database. SA OS/390 displays the *Policy Data Base Report Parameters* panel as shown in Figure 225 on page 359 which offers two options to report either the full automation policy, or to select a specific entry type and name to be reported.

As report generation may run for a long time, this function can also be run in batch mode. As the output may be comprehensive, a separate data set is recommended.

This function creates a member \$RPTPDB in the specified output data set, that needs to be processed by SCRIPT/VS to produce readable output. The *Command Progress Display* panel will inform you about the progress of the report generation.


```

MENU  OPTIONS  HELP
-----
AOFGRPTP          Policy Data Base Report Parameters
Option ==> 1 _____

 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 . . . . 'HART.SCENARIO' _____
Mode . . . . . ONLINE (ONLINE BATCH)

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

```

Figure 225. Policy Data Base Report Parameters Panel

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
- all relationships that are not used during a build

It does not include the entry types related to processor operations: *Processor* (PRO), *Communications Path* (CMP), and *Communications Task* (CMT).

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.

More about Using the Customization Dialog

```
File Edit Confirm Menu Utilities Compilers Test Help
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
VIEW      HART.SCENARIO($RPTRES) - 01.00                      Columns 00001 00072
Command ==>                                                Scroll ==> PAGE
***** ***** Top of Data *****
000001
000002 Application resources for system: SYSTEM1
000003
000004 DB2IRLM/APL/SYSTEM1      DB2IRLM      APL
000005 DB2MAIN/APL/SYSTEM1      DB2MAIN      APL
000006 DWH/APL/SYSTEM1           DWH          APL
000007 TCP/IP/APL/SYSTEM1      TCP/IP       APL
000008 VTAM/APL/SYSTEM1        VTAM         APL
000009
000010 Application resources for system: SYSTEM2
000011
000012 CICS/AOR1/APL/SYSTEM2      CICS/AOR1    APL
000013 CICS/FOR/APL/SYSTEM2      CICS/FOR     APL
000014 CICS/MAIN/APL/SYSTEM2     CICS/MAIN    APL
000015 CICS/STOR/APL/SYSTEM2     CICS/STOR    APL
000016 DWH/APL/SYSTEM2          DWH          APL
000017
000018 ApplicationGroup resources for system: SYSTEM1
000019
000021 DB2/APG/SYSTEM1           DB2          APG
000022 NET/APG/SYSTEM1          NET          APG
000023 WBSRVAPG/APG/SYSTEM1     WBSRVAPG     APG
000024
000025 ApplicationGroup resources for system: SYSTEM2
000026
000027 NET/APG/SYSTEM2          NET          APG
000028
000029 Sysplex ApplicationGroup resources
000030
000031 CICS/APG                 CICS        SYSPLEX1
000032 CICS_APG/APG            CICS_APG    SYSPLEX1
000033 CICS_APG2/APG          CICS_APG2   SYSPLEX1
000034 CICS_APG3/APG          CICS_APG3   SYSPLEX1
000035 CICS_APG4/APG          CICS_APG4   SYSPLEX1
000036 DWHAPG/APG             DWHAPG      SYSPLEX1
```

Figure 226. List of Generated Resources

User-written Functions

Option **U User** on the *Customization Dialog Primary Menu* invokes the *User-written Functions Selection Menus*. This menu allows you to add installation specific functions to the customization dialog.

As an example, one selection has already been provided:

- 1 - Transmit sends the build output data set to a specified user and node.

If you select this option 1, SA OS/390 displays the *Send Policy Data Base* panel (see Figure 227 on page 361), which allows you to 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.

More about Using the Customization Dialog

The *Send Policy Data Base* panel allows you to transmit a build output data set to a specified node and user ID.

```
MENU  HELP
-----
AOFGCM DX                      Send Policy Data Base
Command ==>

To transmit the build output data set to a remote node,
specify or revise the following values:

Data set name . . . 'HART.DOCU.PDB'

Target:
User ID or nickname . . . SMITH      Node ID . . KEY1

Transmit command parameters:
Command parameters. . . .

To issue the transmit command, enter the SEND command ( PF6 ).
To exit from this panel without transmitting data, enter the CANCEL
or END command.
```

Figure 227. Send Policy Data Base Panel

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.

More about Using the Customization Dialog

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 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 62 on page 111).

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 *ACF and AMC Build Parameters* panel (see Figure 205 on page 294) 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. So the generated automation control file and AMC file can be used for automation by just 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 AOFACFMP 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 stand alone system. In each sysplex, it contains only the systems that are marked for the selected configuration, and stand alone system members are created only for systems of 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,

More about Using the Customization Dialog

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 “Creating Reports about Policy Databases” on page 358) 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 OS/390 Customizing and Programming*.

Appendix C. DB2 Automation for System Automation for OS/390

Automation has been produced to provide automated functions for the DB2 software product.

Unlike other SA OS/390 automation products (CICS, IMS and OPC), DB2 Automation has been designed as part of base automation. Consequently DB2 is treated as a normal SA OS/390 application, relying heavily on base functionality. Therefore the material provided here should be read in conjunction with base documentation. Only DB2 Automation-specific information is provided in this document.

Product Overview

Automated functions provided by DB2 Automation are implemented using two distinct methods. The first being line mode invocation which allows for an operator (or OPC) initiated task to be performed on an on-demand basis. The second method is via event-driven functions such as timer expiration and NetView message automation table traps. Timed commands are mainly used to provide for connection monitoring of links to IMS and CICS. Automation table commands support the dynamic discovery of IMS and CICS connections as well as Critical Event Monitoring.

Line Mode Functions

DB2 Automation offers the following operator line command functions:

- **maintenance start**

This provides the ability to start DB2 in a non-standard mode. This is a command line function that will allow for an ACCESS(MAINT) type start and/or a PARM(modname) start.

- **terminate threads**

This provides the ability to stop threads attached to DB2 in order to free DB2 to perform special tasks (backup).

- **start/stop tablespace**

This provides the ability to stop or start a specific Tablespace.

- **event-driven functions**

DB2 Automation event driven functions are via Timer commands or NetView message automation table (AT) Traps.

- **connection monitoring (timer and AT driven)**

This function will facilitate the monitoring of IMS and CICS connections. This is done at connection level via either an ACF entry definition or via dynamic self discovery, or both. If required, a recovery command will be issued to re-establish a lost connection.

- **critical events (AT driven)**

This function is handled at 2 levels, each of which can forward messages to SDF:

1. Specific event (for example, excessive logging).
2. General events using a NetView AT entry to drive a message threshold/recovery process.

Planning, Installation, and Defining Automation Policy

Planning Requirements

DB2 Automation requires SA OS/390 2.1 and its associated prerequisites. For more information on this topic refer to *System Automation for OS/390 Planning and Installation*.

For dynamic discovery of connections, certain messages must be available:

IMS

For IMS, connection monitoring requires that messages DSNM001I, DSNM002I and DSNM003I are available to automation in order to detect the current status of a DB2 connection with IMS. This requires that IMS Automation is installed. For non DBCTL regions the IMS Automation EVISPINM member must be updated with the above mentioned messageIDs so as to expose them to automation via the IMS Automation AOI user exit. For more information please refer to *System Automation for OS/390 IMS Automation Programmer's Reference and Operator's Guide*. The IMS non-DBCTL regions should also be defined to SA OS/390 in order for the dynamically discovered connections to have recovery commands issued as a reply to the correct subsystem.

CICS

For CICS V4, connection monitoring requires that messages DSN2023I, DSN2025I and DSN2016I are available to automation. As this is not possible dynamic discovery is not available to this level of CICS and the ACF CONN entry must be used. For CICS TS, connection monitoring requires that messages DFHDB2023I, DFHDB2025I and DFHDB2037 are available to automation. These are exposed to automation automatically and therefore dynamic discovery is available.

Installation

Once the SMP/E apply has completed, the following steps need to be completed before you are able to use DB2 Automation.

Automation Control File (ACF)

Samples of DB2 Automation are contained in the policy database samples (all except the *DEFAULT sample). The DB2 entries are contained in CLASS/INSTANCE application (APL) relationships. Please refer to "Defining Automation Policy" on page 367 for detailed information on how to implement these and other (SCR) entries into your automation policy.

NetView Message Automation Table (AT)

DB2 Automation AT member INGMDB2 is already included in the standard NetView message automation table provided by SA OS/390. This member contains DFHDB* message traps. If you have CICS product automation installed, then these entries should appear inside the CICS DFH message block. IMS Automation users should merge the provided DFS989I, DFS994I, and DFS4444I message trap entries with any existing entries or place them inside the DFS message block.

Defining Automation Policy

To customize DB2 Automation to your needs, you need to perform the following tasks:

- “Creating Your PDB”
- “Tailoring Your DB2 ACF Entries”

Creating Your PDB

Sample ACF entries are delivered with DB2 Automation. They can be accessed by using the updated ACF sample policy databases. The `PRODUCT_SAMPLES` sample policy database includes the DB2 Automation ACF entries. Regardless of whether you require to add DB2 Automation sample ACF entries to an existing PDB, or create a new PDB containing these entries, you will first need to create a policy database based on one of the updated samples.

“Creating a New Policy Database” on page 17 describes how to create a new policy database using a PDB sample. You can choose either sample PDB as your base PDB, depending on the one that best suits your needs.

Having created a new policy database, based on a sample PDB containing DB2 Automation, if you do not require to add these entries to an existing PDB you can jump to “Tailoring Your DB2 ACF Entries”. Otherwise to add these entries to an existing PDB, perform the following steps to merge DB2 Automation sample ACF entries into your current PDB:

- Edit your sample based PDB and create a new *System* (SYS) policy object.
- Create a new *ApplicationGroup* (APG) policy object and use the WHERE USED policy item to select the new system.
- Enter the *Application* (APL) resources and use the WHERE USED policy item to select the new application group for each of the following entries:

`DB2I_DB2, DSNDBM1I_DB2, DSNDISTI_DB2, DSNIRLMI_DB2, DSNSPASI_DB2`

- Enter the *Status Details* (SCR) resources and use the WHERE USED policy item to select the new system for the `STAT_DETAIL_DB2` entry.
- After saving these PDB changes, perform a full BUILD for all objects so that the new system contains all DB2 Automation sample entries.
- You must then perform a migration of this new system into your PDB that requires DB2 Automation defined.

“Migrate Automation Control File” on page 309 describes how to migrate an automation control file.

Tailoring Your DB2 ACF Entries

After you have created your PDB you will need to edit it in order to add your specific DB2 Automation requirements. To do this, follow the subsequent steps:

1. Select option **4 Policies** from the *Customization Dialog Primary Menu* (Figure 5 on page 14).
2. Select the required policy database that is to contain the DB2 subsystem from the *Policy Data Base Selection* panel (Figure 6 on page 18).
3. Select entry type *Application* from the *Entry Type Selection* panel (Figure 2 on page 8).

DB2 Automation for System Automation for OS/390

4. From the *Entry Name Selection* panel for *Applications*, enter "NEW *entryname*" on the command line and press ENTER in order to create a new policy object that will represent the DB2 MSTR subsystem.
5. On the *Define New Entry* panel, you will need to enter the DB2 master subsystem name, an application type of DB2, the subtype (one of: MSTR SPAS IRLM DBM1 DIST WLMS) and the MVS jobname, where the *db2id* represents the prefix used when you defined your DB2 jobnames to OS/390:

```
Subsystem Name. . . . . subsystem
Application Type. . . . . DB2
Subtype . . . . . MSTR
Job Name. . . . . db2idMSTR
```

6. Press END to save this information. This will bring you to the *Policy Selection* panel for *Applications*.
7. From here select policy item LINK TO CLASS.

Note: One policy item that is inherited at this point is the SHUTDOWN NORM command. This command looks like:

```
INGRDTTH &SUBSAPPL S
```

This has the effect of notifying of, and cancelling, any outstanding threads prior to DB2 shutdown. If you would prefer that threads are not cancelled then this command should be changed to read:

```
INGRDTTH &SUBSAPPL S N
```

8. From the list presented, select DB2C_DB2 and press END, returning to the *Policy Selection* panel.
9. From here select policy item AUTOMATION INFO.
10. When presented with panel *Application Automation Definition*, enter the command prefix character(s) for this DB2 subsystem in the entry:

```
Command Prefix . . cmdprfx Console command character(s)
```

11. Press END to save this information.
12. For *connection monitoring* you must enter CMD and CODE entries for a CONN message to describe any connections that require a forced monitoring action at each monitoring cycle (for example, CICS V4 connection). Refer to "Connection Monitoring" on page 380 for further details on how to add these entries to the MESSAGES policy item:

```
MESSAGES Define Application messages
```

If you wish to force connection status refresh at NetView restart, then add an extra parameter "Y" to the end of the command to be issued for the ACORESTART message entry, for example:

```
AFTER 00:00:10,INGRDCNM &SUBSAPPL Y
```

This has the effect of ensuring that any lost connections during a NetView outage are recovered. This option is not required if you can rely on connections being automatically re-established (that is, CICS/TS).

13. Press END to save this information.
14. Select the extended DB2 subsystem information policy:

DB2 Automation for System Automation for OS/390

DB2 CONTROL Define DB2 Control entries

15. On the *DB2 control entries* panel define DB2 specific subsystem information. The DB2Id should be defined to indicate the subsystem ID and the Active log data set name should be entered.

DB2 subsystem id *db2id*
Active log dataset name. . . . *dsname*

16. Press END to return to the *Policy Selection* panel.
17. Press END again to return to the *Entry Name Selection* panel for *Applications*.
18. Now enter "NEW *entryname*" on the command line and press ENTER in order to create a new policy object that will represent the DB2 DBM1 subsystem. On *Define New Entry* panel AOFGLN00 you will need to enter the subsystem name and the MVS jobname:

Subsystem Name. *subsystem*
Job Name. *db2idDBM1*

19. Press END to save this information and the *Policy Selection* panel will appear.
20. From here select policy item LINK TO CLASS.
21. From the list presented select DSNDBM1C_DB2 and press END, returning to the *Policy Selection* panel.
22. From here select policy item RELATIONSHIPS.
23. Enter "New HASPARENT" on the command line and press Enter. In the *Supporting Resource* field of the upcoming *Define Relationship* panel, enter the subsystem name that you gave for the *db2idMSTR* job and press Enter. For the *Condition* field enter "StartsMeAndStopsMe". The relevant entries on the *Define Relationship* panel should now look like:

Relationship Type. HASPARENT
Supporting Resource. *subsystem/APL/=*
Condition StartsMeAndStopsMe

24. Press END until you reach the *Entry Name Selection* panel for *Applications*.
25. Now enter "NEW *entryname*" on the command line and press ENTER to create a new policy object that will represent the DB2 DIST subsystem. Perform the steps described for the DBM1 subsystem accordingly now for the DIST subsystem.
26. Create a new policy object that will represent the DB2 IRLM subsystem. Perform the steps described for the DBM1 subsystem accordingly now for the IRLM subsystem.
27. Create a new policy object that will represent the DB2 SPAS subsystem. Perform the steps described for the DBM1 subsystem accordingly now for the SPAS subsystem.
28. After the MSTR, DBM1, DIST, IRLM, and SPAS subsystems have been created, they must be linked to an application group (APG).
29. The application group should be linked to the required system(s) that this DB2 subsystem is to be automated on.
30. To support the SDF requirements of DB2 Automation, the status details (SCR) settings are required and should be linked to the same system(s). The policy item is:

DB2 Automation for System Automation for OS/390

STAT_DETAIL_DB2

DB2 Automation status settings

31. After all the relevant ACF definitions have been entered, the ACF can be created using the BUILD command (see “Chapter 6. Building and Distributing Control Files” on page 289).

DB2 Automated Functions – Line Command Functions

Once a DB2 subsystem has been defined to automation, then there are a number of functions that can be performed against it. These are either invoked by a line command or are event-driven (timer or NetView AT).

Command Handler

Purpose

“INGDB2” is the only line command delivered by DB2 Automation. This is referred to as the “Command Handler”.

Syntax

```

▶▶—INGDB2—request—subsystem—┬──┬──▶
                               └─,parm─┘ └─,TARGET=domid─┘
  
```

Parameters

request

START— DB2 startup (in non-standard mode)
 TERM — Terminate threads
 TABLE — Start/Stop Tablespace

For more information refer to “Command Requests” on page 372.

subsystem

The DB2 subsystem that the request is for

parm

A comma delimited positional parameter string.

The number of parameters depends on the command request.

START

parm1 MAINT (for maintenance startup)
 or an asterisk (*, for standard start)

parm2 An optional module name for a non-standard startup.

TERM

None

TABLE

parm1
 START (to start a tablespace) or
 STOP (to stop a tablespace)

parm2
 a database name

parm3
 a tablespace name

domid

A domain within the sysplex on which you wish this command to be invoked (default is current domain). DB2 Automation must be installed on each of the domains that would be required to act as a target.

Messages

AOF010I WRONG NUMBER OF PARAMETERS ENTERED
 AOF204I *time* : EXPECTED PARAMETERS MISSING OR INVALID FOR REQUEST *clist_name* - *parameter_name*
 AOF332I SUBSYSTEM *name* COULD NOT BE LOCATED ON *target*

Command Requests

This is a detailed description of each of the requests that can be invoked via the Command Handler.

Maintenance Start

Purpose

This function will start a DB2 subsystem in a non-standard startup mode.

Using this function, a DB2 subsystem may be started in the following non-standard startup modes:

- Maintenance mode using the default module
- Maintenance mode using a custom module
- Normal startup mode using a custom module.

If a DB2 subsystem is started in maintenance mode then:

- The DDF will be stopped to inhibit any further connections
- Connection monitoring will be suppressed

To perform a standard startup (normal startup mode using default module) the SA OS/390 SETSTATE command should be used to take full advantage of base automation features.

By using the INGREQ command it is possible to start DB2 with the necessary maintenance parameters entered on the "Appl Parm" field. For instance an applparm field value of:

```
ACCESS(MAINT), PARM(DSNxxxxx)
```

for a normal start would be the equivalent of a command line maintenance start, through substitution of the EHKVAR1 parameter. If the maintenance start parameters are consistent, then they may be entered into the ACF via the flexible startup policy. This provides for a variety of start types to be identified and initiated from the INGREQ request panel. See the INGREQ command documentation for further information.

The Maintenance Start function may also be invoked from OPC Automation as a Non subsystem Operation. For further information regarding invocation from the OPC product, refer to *System Automation for OS/390 OPC Automation Programmer's Reference and Operator's Guide*.

Syntax

```
INGDB2 START subsystem,start_type,modname
```

Parameters

subsystem

Name of the DB2 subsystem to be started in a non-standard startup mode.

start_type

Specify MAINT for a DB2 subsystem to be started in maintenance mode.

Specify * for a DB2 subsystem to be started in maintenance mode (*modname* must be specified for this option.)

modname

If *modname* is supplied then this module name will be used to start up DB2 subsystem, otherwise the default module name will be used to start up the DB2 subsystem (this parameter must be specified if *start_type* is *).

Maintenance Start

Restrictions and Limitations

Maintenance Start can only be used when

- SA OS/390 is initialized
- The DB2 subsystem is defined to SA OS/390
- When a standard start is not required
- The OPC interface can only be used if OPC Automation is installed.

Note: When DB2 is started in maintenance mode, the SPAS (stored procedures) address space is not started by DB2. However, SA OS/390 is not aware of this and expects it to be an external startup. As a result the status of SPAS goes into an ambiguous STARTING status. The status will return to normal, when DB2 is next re-started in normal mode. For sites that frequently start up DB2 in maintenance mode, a separate DB2 application group could be defined and used in which the SPAS application is excluded.

Usage

This command may be used to start a DB2 subsystem in a non-standard startup mode, it may be invoked by the INGDB2 command handler or via OPC Automation.

Input parameters are validated for accuracy and any errors found are logged and the process is terminated. The requested subsystem is then checked to determine if automation is enabled. Once these preliminary checks have been successfully completed the requested function is initiated.

Example(s)

The type of startup performed will depend on the invocation parameters.

To startup a DB2 subsystem called DB2P in a non-standard startup mode, enter one of the following commands on the command line:

Example 1

```
INGDB2 START DB2P MAINT
```

Start a DB2 subsystem DB2P in maintenance mode using the default module.

Example 2

```
INGDB2 START DB2P MAINT,DSNMOD1
```

Start a DB2 subsystem DB2P in maintenance mode using custom module DSNMOD1.

Example 3

```
INGDB2 START DB2P *,DSNMOD1
```

Start a DB2 subsystem DB2P in normal mode using custom module DSNMOD1.

Policy Entries

The following DB2 startup command can be found in the sample DB2C_DB2 subsystem class STARTUP policy:

```
MVS &SUBSCMDPFX START DB2 &EHKVAR1
```

The specified command is appended with the required invocation parameters depending on the parameters provided.

Maintenance Start

To invoke the maintenance start function via the OPC Automation interface, the following startup command is required to be coded against an "Automation Function" of UXxxxxxx.

```
INGRDMST &EHKVAR1
```

Where UXxxxxxx must match the first token of the operation text as specified in the OPC plan.

The SHUTDLY ACF entry is used to delay maintenance start should the DB2 subsystem be ACTIVE when this function is invoked.

The STRTDLY ACF entry is used to decide how long to wait before checking to see if the DB2 subsystem is UP once this function is invoked.

Messages

```
AOF014I SPECIFIED PARAMETER parameter INVALID
AOF146I PARAMETER MUST BE NUMERIC
AOF204I EXPECTED PARAMETERS MISSING OR INVALID FOR REQUEST INGRDMST - text
AOF289I SUBSYSTEM subsystem HAS EXCEEDED NORMAL STARTUP INTERVAL.
AOF313I START FOR SUBSYSTEM subsystem (JOB jobname) WAS NOT ATTEMPTED - text
AOF332I SUBSYSTEM subsystem COULD NOT BE LOCATED ON domain
AOF583I AUTOMATION FOR SUBSYSTEM subsystem (JOB jobname) IS SET OFF - AUTOMATION NOT ATTEMPTED
```

Return Codes

```
8      Process failure – see accompanying message.
4      Automation not allowed.
0      Normal End.
-1     Command, instruction or nested command list encountered an error.
-5     Command list cancelled.
```

Error Codes Posted to Tivoli OPC

```
UX21  Automation not allowed.
UX22  Automation control file error.
UX23  DB2 subsystem cannot be started, incorrect status.
UX24  DB2 subsystem is already started.
UX25  DB2 subsystem did not start.
UX26  Error response from AOCQRY.
```

Terminate Threads

Terminate Threads

Purpose

The terminate threads command will terminate all active threads for a DB2 subsystem. These include REMOTE, DB2CALL, BATCH, TSO, CICS/IMS connections and all remaining ('Other') threads.

TSO users will be issued with a message informing them that their thread is about to be terminated prior to actual thread termination.

The Terminate Threads function may also be invoked from the OPC Automation as a Non-subsystem Operation. For further information regarding invocation from OPC Automation, refer to *System Automation for OS/390 OPC Automation Programmer's Reference and Operator's Guide*.

Syntax

```
INGDB2 TERM subsystem
```

Parameters

subsystem

Name of DB2 subsystem for which all active threads are to be terminated.

Restrictions and Limitations

Terminate threads can only be used when:

- SA OS/390 is initialized
- The DB2 subsystem is defined to SA OS/390
- The status of the DB2 subsystem is 'UP'
- The OPC interface can only be used if OPC Automation is installed.

Usage

Use this command to terminate all active threads for a DB2 subsystem. It may be invoked by the INGDB2 command issued from the command line or via OPC Automation.

Input parameters are validated for accuracy and any errors found are logged and the process is terminated. The requested subsystem is then checked to determine if automation is enabled. Once these preliminary checks have been successfully completed the requested function is initiated.

Example(s)

To terminate threads for a DB2 subsystem called DB2P, enter the following from the command line:

```
INGDB2 TERM DB2P
```

Policy Entries

DB2 Control policy item entries can be used to control the length of time to Terminate Threads.

"Terminate Threads Delay" represents the delay between each iteration of the terminate threads request.

"Cycles" represents the maximum number of iterations of the terminate threads request automation is to attempt.

Terminate Threads

To invoke the terminate threads function via the OPC Automation interface, the following startup command is required to be coded against an "Automation Function" of UXxxxxxx.

```
INGRDTTH &EHKVAR1
```

Where UXxxxxxx must match the first token of the operation text as specified in the OPC plan.

IMS BMP threads can be handled separately by requesting this via the Connection Monitoring Policy Entries. Using the CONN Message Policy entry for the DB2 subsystem, create a coded entry as required by the CDEMATCH common routine.

Code 1	Code 2	Code 3	Value Returned
IMSCONID	IMSCTLJB	STOPBMP	YES/NO

where

Code 1

is the IMS connection ID

Code 2

is the IMS Control Region job name

Code 3

is a fixed request identifier

Value Returned

is Yes or No, to indicate if this IMS job's BMPs should be stopped using the IMS /STOP REG ABDUMP command or not.

Messages

```
AOF004I PROCESSING FAILED FOR db2cmd
AOF014I SPECIFIED PARAMETER cycle INVALID
AOF144I PARAMETER parameter_name INVALID
AOF146I PARAMETER MUST BE NUMERIC
AOF204I EXPECTED PARAMETERS MISSING OR INVALID FOR REQUEST INGRDTTH - text
AOF332I SUBSYSTEM subsystem COULD NOT BE LOCATED ON domain
AOF583I AUTOMATION FOR SUBSYSTEM subsystem (JOB jobname) IS SET OFF - AUTOMATION NOT ATTEMPTED
ING107E INDOUBT THREADS EXIST - SHUTDOWN OF subsystem WILL NOT PROCEED
ING108I NO THREADS LEFT IN subsystem
ING109E thread_number THREADS COULD NOT BE TERMINATED FROM subsystem.
ING110I TERMINATED THREADS FROM subsystem, CYCLE cycle.
ING112I YOUR TSO DB2 (subsystem) THREAD IS ABOUT TO BE TERMINATED BY AUTOMATION.
ING113I YOUR TSO DB2 (subsystem) THREAD HAS BEEN TERMINATED BY AUTOMATION.
ING114E jobname CANCELLED BY AUTOMATION DUE TO subsystem THREAD TERMINATION.
ING126I thread_type THREADS: nn.
ING127A THREADS FOUND AFTER LAST CYCLE OF DB2 (subsystem), FORCE SHUTDOWN.
```

Return Codes

8 Process failure – see accompanying message.
4 Automation not allowed.
0 Normal End.
-1 Command, instruction or nested command list encountered an error.
-5 Command list cancelled.

Error Codes Posted to Tivoli OPC

UX41 Automation not allowed.
UX43 Termination of connections unsuccessful.
UX44 Error response from DB2 command.

Start/Stop Tablespace

Start/Stop Tablespace

Purpose

For DB2 Tablespace Start, the necessary Tablespace start command will be issued.

For DB2 Tablespace Stop, certain active threads that use the Tablespace will be terminated. These include REMOTE, DB2CALL, BATCH and TSO. TSO users will be issued with a message informing them that their thread is about to be terminated prior to actual thread termination.

The Tablespace Start/Stop function may also be invoked from the OPC Automation as a Non-subsystem Operation. For further information regarding invocation from OPC Automation, refer to *System Automation for OS/390 OPC Automation Programmer's Reference and Operator's Guide*.

Syntax

```
INGDB2 TABLE subsystem,request_type,dbname,tsname
```

Parameters

subsystem

Name of DB2 subsystem

request type

'START' (start Tablespace)

'STOP' (stop Tablespace)

dbname

Database name

tsname Tablespace name to be started/stopped

Restrictions and Limitations

Tablespace Start/Stop can only be used when:

- SA OS/390 is initialized
- The DB2 subsystem is defined to SA OS/390.
- The OPC interface can only be used if OPC Automation is installed.

Usage

Use this command to start/stop a Tablespace for a DB2 subsystem. It may be invoked by the INGDB2 command handler or via OPC Automation.

Input parameters are validated for accuracy and any errors found are logged and the process is terminated. The requested subsystem is then checked to determine if automation is enabled. Once these preliminary checks have been successfully completed the requested function is initiated.

Example(s)

Example 1

To start a Tablespace where the DB2 subsystem is DB2P, the database name is DB2PDBN and the Tablespace name is DB2PTSN, then enter the following command on the command line:

```
INGDB2 TABLE DB2P,START,DB2PDBN,DB2PTSN
```

Example 2

To stop a Tablespace where the DB2 subsystem is DB2P, the database name is DB2PDBN and the Tablespace name is DB2PTSN, then enter the following command on the command line:

```
INGDB2 TABLE DB2P,STOP,DB2PDBN,DB2PTSN
```

Policy Entries

DB2 Control policy item entries can be used to control the length of time to Terminate Threads.

"STOP tablespace delay" represents the delay between each iteration of stop Tablespace attempt.

TSO logoff delay represents the delay before issuing the TSO logoff message to users of the Tablespace.

To invoke the start/stop Tablespace function via the OPC Automation interface, the following startup command is required to be coded against an "Automation Function" of UXxxxxxx.

```
INGRDSTS &EHKVAR1
```

Where UXxxxxxx must match the first token of the operation text as specified in the OPC plan.

Messages

```
AOF004I PROCESSING FAILED FOR db2cmd
AOF144I PARAMETER parameter_name INVALID
AOF204I EXPECTED PARAMETERS MISSING OR INVALID FOR REQUEST INGRDSTS - text
AOF332I SUBSYSTEM subsystem COULD NOT BE LOCATED ON domain
AOF583I AUTOMATION FOR SUBSYSTEM subsystem (JOB jobname) IS SET OFF - AUTOMATION NOT ATTEMPTED
ING109E thread_no THREADS COULD NOT BE TERMINATED FROM subsystem
ING129E jobname CANCELLED. TABLESPACE dbname.tsname(subsystem) NEEDED TO BE STOPPED.
ING130I TABLESPACE dbname.tsname(subsystem) IS TO BE STOPPED. PLEASE STOP USING IT.
ING131I YOU WERE CANCELLED BECAUSE TABLESPACE dbname.tsname(subsystem) IS TO BE STOPPED.
ING132I thread_no THREADS CANCELLED DUE TO STOP OF TABLESPACE dbname.tsname(subsystem)
```

Return Codes

```
8      Process failure – see accompanying message.
4      Automation not allowed.
0      Normal End.
-1     Command, instruction or nested command list encountered an error.
-5     Command list cancelled.
```

Error Codes Posted to Tivoli OPC

```
UXA1  Automation not allowed.
UXA2  Error response from DB2 command.
UXA4  Tablespace is still allocated.
```

Event-Driven Functions

This is a detailed description of each of the commands that can be invoked as the result of an event, either a NetView AT trap or a NetView Timer expiration.

Connection Monitoring

Connection Monitoring

Purpose

Connection monitoring is designed to, where possible, dynamically discover CICS and IMS connections to a DB2 subsystem. Once discovered the status of a connection can be maintained by tracking the relevant messages generated as the connection is affected by the operating environment.

When dynamic discovery is not feasible (due to the inability of automating the relevant messages) then the connection information can be read from ACF CONN entry-type entries, and the status checked by issuing the relevant MVS commands.

In all cases, should the connection be found in the "DOWN" status then the necessary restart command will be automatically issued.

Restrictions and Limitations

Connection Monitoring can only be used when:

- SA OS/390 is initialized.
- The DB2 connection is defined to SA OS/390.
- The CICS and IMS application is defined to SA OS/390.
- The CICS or IMS attachment facility is installed for the relevant subsystems.

Usage

This function is driven by a NetView Timer expiration in order to check the connections that are being monitored to be "UP" and, if necessary, issue a recovery command. This function can also be driven from the NetView AT trap in order to update the connection status.

Input parameters are validated for accuracy and any errors found are logged and the process is terminated. The requested subsystem is then checked to determine if automation is enabled. Once these preliminary checks have been successfully completed the requested function is initiated.

For the NetView AT event driven process the automation flag for the connection minor resource is **not** checked and, depending on the message that was trapped, the relevant CGlobal information for the particular connection will be updated.

If the process is driven as a result of a NetView Timer expiration then all known connections are checked for availability. The connections to check are identified from the ACF CONN entry-type entries, as well as the Cglobals built from the NetView AT driven process. The individual connection's automation flags are checked to see if recovery should be considered. For each connection, if automation is "ON" and the connection status is "DOWN" (all ACF CONN connection entries are assumed to be "DOWN" for this purpose) then the connection is checked for its current status. If the connection is confirmed to be "DOWN" then a recovery command will be issued. For ACF entry identified connections the recovery command is issued from the ACF, otherwise a command is built from discovered information and then issued.

Policy Entries

DB2 Control policy item entries can be used to control the length of time between Connection Monitoring cycles.

"Connection monitor delay" represents the delay between each NetView Timer expiration which will trigger the connection status checking cycle.

Connection Monitoring

Connection monitoring is initially invoked to run on a timer initiated by the DB2 UP or NetView restart messages.

You can control Connection Monitoring by using the CONN and CONN.*connid* minor resource automation flags, where *connid* represents the name of a connection that requires automation to be switched off.

The connection identification entries can be entered into your automation policy using the ISPF Customization Dialogs "MESSAGES" policy item for the DB2 subsystem. The Message ID should be "CONN" against which the "CODE" and "CMD" entries should be made:

Code 1	Code 2	Code 3	Value Returned
CICONID	CICS4A	CICS	CICS4A ENTRY

where

Code 1 is the connection ID (CICS applid)

Code 2 is the CICS jobname

Code 3 is the connection type (CICS or IMS)

Code 4 is the connection description.

and this could be the CMD entry required:

Pass or Selection	Automated Function	Command Text
CICONID		MVS F CICS4A,DSNC STRT Z

where

Pass or Selection
is the connection ID

Automated Function
is not used

Command Text
is the required recovery command to be issued.

AT Entries

The following messages will trigger connection monitoring to run from the NetView message automation table.

These messages require IMS Automation to be installed, as they are required to be added to the EVISPINM table and be available to IMS Automation AOI exit in the case of non-DBCTL regions, or pre-message set to "YES" for DBCTL regions. Please refer to Chapter 10, "Optional Additions to the PPI", in the *System Automation for OS/390 IMS Automation Programmer's Reference and Operator's Guide*.

```
IF MSGID='DSNM001I'  
THEN EXEC(CMD('INGRDCNM '));
```

```
IF MSGID='DSNM002I'  
THEN EXEC(CMD('INGRDCNM '));
```

```
IF MSGID='DSNM003I'  
THEN EXEC(CMD('INGRDCNM '));
```

Connection Monitoring

Messages

AOF004I PROCESSING FAILED FOR *db2cmd*
AOF144I PARAMETER *parameter_name* INVALID
AOF204I EXPECTED PARAMETERS MISSING OR INVALID FOR REQUEST INGRDCNM - *text*
AOF205A *time : command* COMMAND FAILED FOR *clist_name : interval* - WAIT TIME EXPIRED
AOF332I SUBSYSTEM *subsystem* COULD NOT BE LOCATED ON *domain*
AOF583I AUTOMATION FOR SUBSYSTEM *subsystem* (JOB *jobname*) IS SET OFF - AUTOMATION NOT ATTEMPTED
ING101A *subsystem* CONNECTION TO *conn_desc* (*conn_id*) DOWN. RECOVERY COMMAND ISSUED
ING102I *subsystem* CONNECTION TO *conn_desc* (*conn_id*) IS UP.

Return Codes

8 Process failure – see accompanying message.
4 Automation not allowed.
0 Normal End.
-1 Command, instruction or nested command list encountered an error.
-5 Command list cancelled.

Critical Event Monitoring

Purpose

Critical Event Monitoring is split into two distinct levels.

1. Specific Event Monitoring

This level of Critical Event Monitoring handles specific Critical Events that may occur during normal day to day running of DB2.

These include:

DSNB250E, DSNB311I, DSNB312I, DSNB320, DSNB321I,DSNB322I, DSNB323I, DSNB350I, DSNB351I	Recover Failed Dataspace (for data sharing only)
DSNB309I	Recover failed Group Buffer Pool
DSNJ002I	Switch active log datasets
DSNR004I	Does not contain INDOUBT=0
DSNP007I	Dataset could not be extended
DSNJ110E	Last active log data set is every 5% full
DSNJ111E	All active log data sets full
DSNJ115I	Archive data set could not be allocated
DSNT500I/501I	Generates alert

2. Non-Specific Event Monitoring

This level of Critical Event Monitoring handles non-specific Critical Events that may occur during normal day to day running of DB2. These include any DB2 messages that are entered into the NetView AT to invoke INGRDREC (the generic critical event handler).

This function will attempt to issue a recovery command, for the trapped DB2 message, from the ACF if its critical threshold has not been exceeded. If a command does not exist, then an alert is forwarded to SDF.

Required recovery commands are to be entered into the customization dialog under the MESSAGES/USER DATA policy item for entry type *Application* (APL), for any DB2 subsystem that requires DB2 Critical Event message recovery.

Restrictions and Limitations

Critical Event Monitoring can only be used when:

- SA OS/390 is initialized
- The DB2 subsystem is defined to SA OS/390.

Usage

For each of the Event Monitoring processes:

Input parameters are validated for accuracy and any errors found are logged and the process is terminated. The requested subsystem is then checked to determine if automation is enabled. Once these preliminary checks have been successfully completed the requested function is initiated.

DSNB250E, DSNB311I, DSNB312I, DSNB320I, DSNB321I, DSNB322I, DSNB323I, DSNB350I, DSNB351I – Recover Failed Dataspace

Critical Event Monitoring

Policy Entries

None.

AT Entries

```
IF MSGID='DSNB250E' |
MSGID='DSNB311I' |
MSGID='DSNB312I' |
MSGID='DSNB320I' |
MSGID='DSNB321I' |
MSGID='DSNB322I' |
MSGID='DSNB323I' |
MSGID='DSNB350I' |
MSGID='DSNB351I' |
THEN EXEC(CMD('INGRDSTS'));
```

Messages

None.

DSNB309I – Recover Failed Group Buffer Pool

This function will stop DB2 on receipt of the DSNB309I for Group Buffer Pool GBP0. This is triggered using an AT trap which will invoke INGRDTTH to perform the INGREQ STOP command. SA OS/390 will then attempt to start any other DB2 defined within the sysplex based on preference values.

Policy Entries

None.

AT Entries

```
IF MSGID = 'DSNB309I' &
TEXT = . 'POOL GBP0' . 'FAILURE' .
THEN EXEC(CMD('INGRDTTH ') ROUTE(ONE %AOFOPGSSOPER%));
```

Messages

None.

DSNV086E – Unrecoverable/Recoverable DB2 Abends

This function will identify specific DB2 abends as non-recoverable. This will cause the DB2 subsystem to "Break" DB2. SA OS/390 will then attempt to start any other DB2 defined within the sysplex based on preference values. Other DB2 abends will be recoverable.

Policy Entries

None.

AT Entries

```
*Non Recoverable
IF MSGID = 'DSNV086E' &
(TEXT = .'REASON=00F70600') *
(TEXT = .'REASON=00F70602') *
(TEXT = .'REASON=00E30105') *
(TEXT = .'REASON=00E30078')
THEN EXEC(CMD('INGRDTTH ') ROUTE(ONE %AOFOPGSSOPER%))
EXEC(CMD('TERMSG FINAL=YES,BREAK=YES')
ROUTE(ONE %AOFOPGSSOPER%));
```

* Recoverable

```
IF MSGID = 'DSNV086E'
  THEN EXEC(CMD(' INGRDTTH ') ROUTE(ONE %AOFOPGSSOPER%))
  EXEC(CMD(' TERMMSG FINAL=NO,ABEND=YES' )
  ROUTE(ONE %AOFOPGSSOPER%));
```

Messages

None.

DSNJ002I – Switch active log datasets

This function will issue a command when the critical threshold is reached for the DSNJ002I message occurrence for the log dataset specified by the "Active log dataset" DB2 Control policy item.

Policy Entries

The required command to be issued should be entered using the Messages policy item. Thresholds should be entered against the LOG minor resource for the DB2 subsystem in question.

AT Entries

```
IF MSGID='DSNJ002I'
  THEN EXEC(CMD(' INGRD002 '));
```

Messages

```
AOF583I  AUTOMATION FOR SUBSYSTEM subsystem(JOB jobname) IS SET OFF - AUTOMATION NOT ATTEMPTED
ING115A  DB2 SUBSYSTEM subsystem IS DOING EXCESSIVE LOGGING.
```

DSNR004I - Does not contain INDOUBT=0

This function will issue an alert when the DSNR004I is issued with an INDOUBT value greater than zero.

Policy Entries

None.

AT Entries

```
IF MSGID='DSNR004I'
  THEN EXEC(CMD(' INGRD004 '));
```

Messages

None.

DSNP007I – Dataset could not be extended

This function will first check to see if the message contains either of the reserved database IDs DSNDB01, or DSNDB06, as well as checking the list that can be provided by the ACF DATABASE CODE entry for this DB2 subsystem. If one of these database names is found in the message then this function will issue a command from the ACF only if the original message was for a BATCH job. An alert is issued for both BATCH and non BATCH jobs.

Policy Entries

The command to be issued should be entered using the Messages policy item. This policy item should also be used to describe the database names that should be used for matching against those contained within the message. The Message ID should be "DATABASE" against which the "CODE" entry should be made. For example:

Critical Event Monitoring

Code 1	Code 2	Code 3	Value Returned
DSNDB04			NULL
DSNDB07			NULL
DSQDBCTL			NULL
DSNGEDLC	00C90089		NULL

Where Code1, Code2, and Code3 entries can hold any combination of database name and return code character strings. These values are checked to see if they are contained within the DSNJ007I message text to decide if this function should proceed. The Value Returned is set to NULL as it is a requirement of the CODE entry to contain a non-blank value.

AT Entries

```
IF MSGID='DSNP007I'  
THEN EXEC(CMD('INGRD007 '));
```

Messages

```
AOF583I  AUTOMATION FOR SUBSYSTEM subsystem(JOB jobname) IS SET OFF - AUTOMATION NOT ATTEMPTED  
ING118E  EXTEND FAILED, DSN dsname IN SUBSYSTEM subsystem.  
ING119A  EXTEND FAILED, DSN dsname IN SUBSYSTEM subsystem.
```

DSNJ110E– Last active log dataset is % full

This function will issue a command from the ACF when the critical threshold is reached for the DSNJ110E message. This is determined by the percentage full figure in the message becoming equal to or greater than that entered in the “Value Returned” part of the ACF PRIMLOG_PCT entry.

Policy Entries

The command to be issued should be entered using the Messages policy item.

The DB2 Control policy item entry can be used to enter this percentage value. Here 90 represents the threshold value which will be compared with the “% full” value in the message. If the value in the message exceeds this threshold, automation proceeds to issue the ACF specified command.

AT Entries

```
IF MSGID='DSNJ110E'  
THEN EXEC(CMD('INGRD111 '));
```

Messages

```
AOF583I  AUTOMATION FOR SUBSYSTEM subsystem (JOB jobname)  
IS SET OFF - AUTOMATION NOT ATTEMPTED
```

DSNJ111E – All active log datasets full

This function will issue an alert each time that this message is received outside of the “Active log alert” time period. This will prevent a flood of logged messages. Also if the number of messages received within this elapsed period exceeds a “Threshold” limit then a command will be issued from the ACF. Both of these control parameters can be entered via the DB2 Control policy item for the DB2 subsystem.

Policy Entries

The command to be issued should be entered using the Messages policy item.

AT Entries

```
IF MSGID='DSNJ111E'
THEN EXEC(CMD('INGRD111 '));
```

Messages

AOF583I AUTOMATION FOR SUBSYSTEM *subsystem*(JOB *jobname*) IS SET OFF - AUTOMATION NOT ATTEMPTED
 ING116A DB2 SUBSYSTEM *subsystem* IS WAITING FOR LOG DATA SETS.

DSNJ115I – Archive dataset could not be allocated

This function will issue an alert, along with any specified recovery command, each time this message is received outside of the "Log off load interval" time setting. This will prevent a flood of logged messages getting processed. The control parameter can be entered via the DB2 Control policy item for the DB2 subsystem.

Policy Entries

The command to be issued should be entered using the Messages policy item.

AT Entries

```
IF MSGID='DSNJ115I'
THEN EXEC(CMD('INGRD115 '));
```

Messages

AOF583I AUTOMATION FOR SUBSYSTEM *subsystem*(JOB *jobname*) IS SET OFF - AUTOMATION NOT ATTEMPTED
 ING117A DB2 SUBSYSTEM *subsystem* COULD NOT ALLOCATE AN ARCHIVE DATA SET

DSNT500I/501I – Generate DSNT500I/501I alert

This function will first check to see if the message contains either of the reserved database IDs DSNDB01, or DSNDB06, as well as checking the list that can be provided by the ACF DATABASE CODE entry for this DB2 subsystem. If one of these database names is found in the message then this function will forward this message as an alert.

Policy Entries

This Message policy item should also be used to describe the database names that should be used for matching against those contained within the message. The Message ID should be "DATABASE" against which the "CODE" entry should be made. The Message ID should be "DATABASE" against which the "CODE" entry should be made. For example:

Code 1	Code 2	Code 3	Value Returned
DSNDB04			NULL
DSNDB07			NULL
DSQDBCTL			NULL
DSNGEDLC	00C90089		NULL

Where Code1, Code2, and Code3 entries can hold any combination of database name and return code character strings. These values are checked to see if they are contained within the DSNT500/501I message text to decide if this function should proceed. The Value Returned is set to NULL as it is a requirement of the CODE entry to contain a non-blank value.

AT Entries

```
IF (MSGID='DSNT500I' | MSGID='DSNT501I')
THEN EXEC(CMD('INGRD500 '));
```

Critical Event Monitoring

Messages

ING134I *msgtext*

DSNnnnnS – Generic alert

This function will attempt to issue either a reply or a command for the message DSNnnnnS, from the ACF if the critical threshold has not been exceeded for the message. If a reply and command does not exist, then an alert is forwarded to SDF.

System Environment

Policy Entries

The command or reply to be issued should be entered using the Messages policy item. The DB2 Thresholds policy item should be used to enter the thresholds using the message ID as the minor resource name.

AT Entries

```
IF MSGID='DSNnnnnS'  
THEN EXEC(CMD('INGRDREC EHKVAR1'));
```

Messages

```
AOF583I AUTOMATION FOR SUBSYSTEM subsystem(JOB jobname) IS SET OFF - AUTOMATION NOT ATTEMPTED  
AOF584I time : resname autotype IS SET ON - autotypeCOMMAND NOT FOUND FOR  
          restype resname - "text"  
AOF587I time : RECOVERY FOR restype resname CONTINUING - CRITICAL THRESHOLD action  
ING133E msgtext
```

Return Codes

- 8 Process failure – see accompanying message.
- 4 Automation not allowed.
- 0 Normal End.
- 1 Command, instruction or nested command list encountered an error.
- 5 Command list cancelled.

Glossary

This glossary includes terms and definitions from:

- The *IBM Dictionary of Computing* New York: McGraw-Hill, 1994.
- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies can be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018. Definitions are identified by the symbol (A) after the definition.
- The *Information Technology Vocabulary* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.

The following cross-references are used in this glossary:

Contrast with. This refers to a term that has an opposed or substantively different meaning.

Deprecated term for. This indicates that the term should not be used. It refers to a preferred term, which is defined in its proper place in the glossary.

See. This refers the reader to multiple-word terms in which this term appears.

See also. This refers the reader to terms that have a related, but not synonymous, meaning.

Synonym for. This indicates that the term has the same meaning as a preferred term, which is defined in the glossary.

Synonymous with. This is a backward reference from a defined term to all other terms that have the same meaning.

A

ACF. Automation control file.

ACF/NCP. Advanced Communications Function for the Network Control Program. See *Advanced Communications Function* and *Network Control Program*.

ACF/VTAM*. Advanced Communications Function for the Virtual Telecommunications Access Method. Synonym for *VTAM*. See *Advanced Communications Function* and *Virtual Telecommunications Access Method*.

ACO. Automated console operations.

active monitoring. In SA OS/390, the acquiring of resource status information by soliciting such information at regular, user-defined intervals. See also *passive monitoring*

adapter. Hardware card that enables a device, such as a workstation, to communicate with another device, such as a monitor, a printer, or some other I/O device.

adjacent hosts. Systems connected in a peer relationship using adjacent NetView sessions for purposes of monitoring and control.

adjacent NetView. In SA OS/390, the system defined as the communication path between two SA OS/390 systems that do not have a direct link. An adjacent NetView is used for message forwarding and as a communication link between two SA OS/390 systems. For example, the adjacent NetView is used when sending responses from a focal point system 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 the NetView program, 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 service value that determines the level at which SA OS/390 changes the associated icon on the NMC workstation to the alert color. SA OS/390 may also issue an alert. See *warning threshold*

AMC. (1) Automation Manager Configuration (2) The Auto Msg Classes entry type

APF. Authorized program facility.

APPC. Advanced program-to-program communications.

application. An OS/390 subsystem or job monitored by SA OS/390.

Application entry. A construct, created with the customization dialog, used to represent and contain policy for an application.

application group. A named set of applications. An application group is used to represent and contain common policy for the contained applications. Applications in an application group can be automated as an entity. Application groups can behave differently according to their defined nature (BASIC, MOVE, SERVER). An application group can also be used for monitoring purposes.

ApplicationGroup entry type. A construct, created with the customization dialog, 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 OS/390 running a routine (the ASCB checker) which searches the OS/390 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-bits 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 OS/390 facility that uses SDF and enables interaction with automation before SA OS/390 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 dialog, or dynamically.

authorized program facility (APF). A facility that permits identification of programs that are authorized to use restricted functions.

automated function. SA OS/390 automated functions are NetView automation operators, that are assigned to perform specific automation functions. However, SA OS/390 defines its own synonyms, or *automated function names*, for the NetView automation operators, and these function names are referred to in the sample policy databases provided by SA OS/390. For example, the automation operator AUTBASE corresponds to the SA OS/390 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. An OS/390 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 OS/390.

automatic restart management element name. In MVS 5.2 or later, OS/390 automatic restart management requires the specification of a unique sixteen character name for each address space which registers with it. All automatic restart management policy is defined in terms of the element name, including SA OS/390's interface with it.

automation. The automatic initiation of actions in response to detected conditions or events. SA OS/390 provides automation for OS/390 applications, OS/390 components, and remote systems that run OS/390. SA OS/390 also provides tools that can be used to develop additional automation.

automation agent. In SA OS/390, 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 control file (ACF). In SA OS/390, a file that contains system-level automation policy information. There is one master automation control file for each NetView system on which SA OS/390 is installed. The SA OS/390 customization dialog must be used to build the automation control files. They must not be edited manually.

automation flags. In SA OS/390, 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 OS/390 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 dialog.

automation manager. In SA OS/390, 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 stand-alone system.

Automation NetView. In SA OS/390 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 OS/390 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, OS/390 subsystems, OS/390 data sets, and OS/390 components.

automation policy settings. The automation policy information contained in the automation control file. This information is entered using the customization dialog. You can display or modify these settings using the customization dialog.

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 the NetView program.

automation status file. In SA OS/390, a file containing status information for each automated subsystem, component or data set. This information is used by SA OS/390 automation when taking action or when determining what action to take.

automation table. Also called NetView message automation table. See *NetView message automation table*

autotask. See *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*.

beaconing. The repeated transmission of a frame or messages (beacon) by a console or workstation upon detection of a line break or outage.

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 OS/390, channel path ID; the address of a channel.

CHPID port. A label that describes the system name, logical partitions, and channel paths.

channel-attached. (1) Attached directly by I/O channels to a host processor (for example, a channel-attached device). (2) Attached to a controlling unit by cables, rather than by telecommunication lines. Contrast with *link-attached*. Synonymous with *local*.

CI. Console integration.

CICS/VS. Customer Information Control System for Virtual Storage.

CLIST. Command list.

clone. A set of definitions for application instances which are derived from a basic application definition by substituting a number of different system-specific values into the basic definition.

clone ID. A generic means of handling system-specific values such as the MVS SYSCONE or the VTAM subarea number. Clone IDs can be substituted into application definitions and commands to customize a basic application definition for the system that it is to be instantiated on.

CNC. A channel path that transfers data between a host system image and an ESCON control unit. It can be point-to-point or switchable.

command. A request for the performance of an operation or the execution of a particular program.

command facility. The component of the NetView program 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 OS/390, REXX command lists that can be used for automation procedures.

command procedure. In the NetView program, 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 OS/390 subset of the CPC operations management commands.

common routine. One of several SA OS/390 programs that perform frequently used automation functions. Common routines can be used to create new automation procedures.

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*.

communications path. A set of communications facilities that SA OS/390 uses to connect a focal point system to a target system. There are two types: a NetView connection and a workstation.

communications task. An SA OS/390 task responsible for all communications with a number of workstations. Communications tasks receive commands from target control tasks and send inbound messages to the message monitoring task. There can be many communications tasks. Communications tasks are defined using the configuration dialogs.

configuration dialogs. A user interface for entering the information that defines the SA OS/390 configuration. In SA OS/390 the configuration dialogs are an ISPF application.

connectivity view. In SA OS/390, 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 OS/390, 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 which 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 OS/390 systems in a sysplex. See also *sysplex couple data set* and *XCF couple data set*

coupling facility. The hardware element that provides high-speed caching, list processing, and locking functions in a sysplex.

CP. Central processor.

CPC. Central processor complex.

CPC operations management commands. A set of commands and responses for controlling the operation of System/390 CPCs.

CPC subset. All or part of a CPC. It contains the minimum *resource* to support a single control program.

CPCB. Command processor control block; an I/O operations internal control block that contains information about the command being processed.

CPU. Central processing unit. Deprecated term for *processor*.

cross-system coupling facility (XCF). XCF is a component of OS/390 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 dialog. The customization dialog is an ISPF application. They are used to customize the enterprise policy, like for example the automated enterprise resources and the relationships between resources, or the automation policy for systems in the enterprise.

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.

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.

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). An OS/390 component that provides limited-function console communication during system recovery situations.

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.

DSIPARM. This file is a collection of members of NetView's DSIPARM data set. It determines how automation is applied to resources.

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.

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 OS/390 should stop trying to restart or recover an application, subsystem or component, or off-load a data set.

ESA. Enterprise Systems Architecture.

event. (1) In the NetView program, 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 OS/390

monitoring highlights exception conditions and allows an SA OS/390 enterprise to be managed by exception.

extended recovery facility (XRF). A facility that minimizes the effect of failures in OS/390, 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*

file manager commands. A set of SA OS/390 commands that read data from or write data to the automation control file. These commands are useful in the development of automation that uses SA OS/390 facilities.

focal point. In the NetView program, the focal point domain is the central host domain. It is the central control point for any management services element containing control of the network management data.

focal point system. (1) A system that can administer, manage, or control one or more target systems. There are a number of different focal point systems associated with IBM automation products. (2) **SA OS/390 automation focal point system.** The SA OS/390 automation focal point system is an SA OS/390 NetView system that collects status information from other SA OS/390 NetViews within your enterprise. It is supported by SA OS/390, which uses NetView NNT/OST sessions to forward information to it. (3) **SA OS/390 configuration focal point system.** The SA OS/390 configuration focal point is the system on which SA OS/390 is installed. The SA OS/390 policy databases exist on this system and are built into automation control file fragments, and processor control file on this system (if this is being done using SA OS/390). (4) **SA OS/390 workstation focal-system.** The SA OS/390 workstation focal point must be the same as the NMC focal point system. SA OS/390 code is installed on both the workstation and the NMC focal point system. The SA OS/390 workstation focal point system collects status information about systems and applications within your enterprise. It is supported by SA OS/390. (5) **NMC focal point system.** The NMC focal point system is a NetView system with an attached workstation server and LAN which gathers information about the state of the network. This focal point system uses RODM to store the data it collects in the data model. The information stored in RODM can be accessed from any LAN-connected workstation with NetView Management Console installed. (6) **NPDA focal point system.** This is a NetView system which collects all the NPDA alerts that are generated within your enterprise. It is supported by NetView. If you have SA OS/390 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*. (7) **Status focal point system.** In NetView, the system to which STATMON, VTAM and NLDM send status information on network resources. If you have a NMC focal point, it must be on the same system as the Status focal point. (8) **SA OS/390 focal point system.** This is a NetView system that has SA OS/390 host code installed, and network connections to a number of SA OS/390 workstations. The SA OS/390 focal point system receives messages from the systems and operator consoles of the machines 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. (9)

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 the NetView program, 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 OS/390 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 OS/390, a set of self-contained automation routines that can be called from the NetView message automation table, or from user-written automation procedures.

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 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 OS/390, a function which monitors the validity of the status forwarding path between remote systems and the NMC focal point system, and monitors the availability of remote OS/390 systems, to ensure that status information displayed on the SA OS/390 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 OS/390 publications refer to target systems or focal-point systems instead of hosts.

host (primary processor). The processor at which you enter a command (also known as the *issuing processor*)

HWMCA. Hardware Management Console Application. Application for the graphic hardware management console that monitors and controls a central processor complex. It is attached to a target processor (a system 390 microprocessor cluster) as a dedicated system console. This microprocessor uses OCF to process commands.

I

images. A grouping of processors and I/O devices that you define. You can define a single-image mode which allows a multiprocessor system to function as one central processor image.

IMS/VS. Information Management System/Virtual Storage.

inbound. In SA OS/390, messages sent to the focal-point system from the PS/2 computer 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 OS/390-provided automation that issues the correct OS/390 start command for each subsystem when SA OS/390 is initialized. The automation ensures that subsystems are started in the order specified in the automation control file and that prerequisite applications are functional.

input/output support processor (IOSP). The hardware unit that provides I/O support functions for the primary support processor and maintenance support functions for the processor controller.

Interactive System Productivity Facility (ISPF). An IBM licensed program that serves as a full-screen editor and dialog manager. Used for writing application programs, it provides a means of generating standard screen panels and interactive dialogs between the application programmer and the terminal user.

interested operator list. The list of operators who are to receive messages from a specific target system.

internal token. A *logical token* (LTOK); name by which the I/O resource or object is known; stored in IODF.

Inter-System Control Facility (ISCF). A program offering that provides remote operations support for 3090, 4381, and 308x processors. ISCF is a predecessor to SA OS/390. It runs as a command processor on the NetView program and works in conjunction with the ISCF/PC program.

Inter-System Control Facility/PC (ISCF/PC). A program offering that runs as an application on an IBM personal computer and works in conjunction with ISCF.

IOCDs. I/O configuration data set. The data set that describes the I/O configuration.

I/O Ops. See *I/O operations*

IOSP. Input/Output Support Processor.

I/O operations. The part of SA OS/390 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).

I/O resource number. Combination of channel path identifier (CHPID), device number, etc. See internal token.

IPL. Initial program load.

ISA. Industry Standard Architecture.

ISCF. Inter-System Control Facility.

ISCF/PC. Inter-System Control Facility/Personal Computer.

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 OS/390 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 OS/390 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 OS/390, link connection is the physical medium of transmission.

link-attached. Describes devices that are physically connected by a telecommunication line. Contrast with *channel-attached*.

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 which 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 () mode*.

logical switch number (LSN). Assigned with the switch parameter of the CHPID macro of the IOCP

logical token (LTOK). Resource number of an object in the IODF

logical unit (LU). In SNA, a port through which an end user accesses the SNA network and the functions provided by system services control points (SSCPs). An LU can support at least two sessions — one with an SSCP and one with another LU — and may be capable of supporting many sessions with other LUs. See also *physical unit (PU)* and *system services control point (SSCP)*.

logical unit (LU) 6.2. A type of logical unit that supports general communications between programs in a distributed processing environment. LU 6.2 is characterized by (a) a peer relationship between session

partners, (b) efficient use of a session for multiple transactions, (c) comprehensive end-to-end error processing, and (d) a generic application program interface (API) consisting of structured verbs that are mapped into a product implementation. Synonym for advanced program-to-program communications (APPC).

logically partitioned (LPAR) mode. A central processor mode that enables an operator to allocate system processor hardware resources among several logical partitions. Contrast with *basic mode*.

LOGR. The sysplex logger.

LP. Logical partition.

LPAR. Logically partitioned (mode).

LU. Logical unit.

LU-LU session. In SNA, a session between two logical units (LUs) in an SNA network. It provides communication between two end users, or between an end user and an LU services component.

LU 6.2. Logical unit 6.2.

LU 6.2 session. A session initiated by VTAM on behalf of an LU 6.2 application program, or a session initiated by a remote LU in which the application program specifies that VTAM is to control the session by using the APPCCMD macro.

M

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.

NetView message automation table. A table against which the NetView program compares incoming messages. A match with an entry triggers the specified response. SA OS/390 entries in the NetView automation table trigger an SA OS/390 response to target system conditions.

message class. A number that SA OS/390 associates with a message to control routing of the message. During automated operations, the classes associated with each message issued by SA OS/390 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 OS/390 process of sending messages generated at an SA OS/390 target system to the SA OS/390 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). An OS/390 table that screens all messages sent to the OS/390 console. The MPF compares these messages with a customer-defined list of messages on which to automate, suppress from the OS/390 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 OS/390 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.

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 OS/390 images in a multisystem environment.

multisystem environment. An environment in which two or more OS/390 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 OS/390.

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. An OS/390 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.

MVS/ESA SP. Multiple Virtual Storage/Enterprise Systems Architecture System Product.

MVS/XA. Multiple Virtual Storage for Extended 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 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 OS/390 activities.

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 *operator station task*

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.

NetView-NetView task session. A session between two NetView programs that runs under an NetView-NetView task. In SA OS/390, NetView-NetView task sessions are used for communication between focal point and remote systems.

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 OS/390 the NetView that performs network management functions, such as managing the configuration of a network. In SA OS/390 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.

NMC focal point system. See *focal point system*

NMC workstation. The NMC workstation is the primary way to dynamically monitor SA OS/390 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.

NIP. Nucleus initialization program.

NNT. NetView-NetView task.

notification message. An SA OS/390 message sent to a human notification operator to provide information about significant automation actions. Notification messages are defined using the customization dialog.

notification operator. A NetView console operator who is authorized to receive SA OS/390 notification messages. Authorization is made through the customization dialog.

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

OCA. In SA OS/390, operator console A, the active operator console for a target system. Contrast with *OCB*.

OCB. In SA OS/390, 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. Contrast with *screen-oriented processor*.

OPC/A. Operations Planning and Control/Advanced.

OPC/ESA. Operations Planning and Control/Enterprise Systems Architecture.

operating system (OS). Software that controls the execution of programs and that may provide services such as resource allocation, scheduling, input/output control, and data management. Although operating systems are predominantly software, partial hardware implementations are possible. (T)

OS/2. A software operating system for personal computers (workstations). OS/2 supports sophisticated business applications and gives you multiple simultaneous host sessions, DOS sessions, Windows sessions, and OS/2 sessions with a graphic interface that lets you select resources by clicking on an icon with a mouse.

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 OS/390, 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 OS/390, a console that displays output from and sends input to the operating system (OS/390, VM, VSE, or TPF). Also called *operating system console*. In the SA OS/390 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.

OS/390 component. A part of OS/390 that performs a specific OS/390 function. In SA OS/390, component refers to entities that are managed by SA OS/390 automation.

OS/390 subsystem. Software products that augment the OS/390 operating system. JES and TSO/E are examples of OS/390 subsystems. SA OS/390 includes automation for some OS/390 subsystems.

OS/390 system. An OS/390 image together with its associated hardware, which collectively are often referred to simply as a system, or OS/390 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.

OS/2. The IBM Operating System/2 program.

OS/2 program. The IBM Operating System/2* program. SA OS/390 uses the intersystem communication and emulation services of the Communication Manager.

outbound. In SA OS/390, messages or commands from the focal-point system to the PS/2 computer or 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 OS/390 panels you can see status, type commands on a command line using a keyboard, configure your system, and passthru to other consoles. See also *help panel*. (3) In computer graphics, a display image that defines the locations and characteristics of display fields on a display surface. Contrast with *screen*.

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

parameter. (1) A variable that is given a constant value for a specified application and that may denote the application. (2) An item in a menu for which the user specifies a value or for which the system provides a value when the menu is interpreted. (3) Data passed to a program or procedure by a user or another program, namely as an operand in a language statement, as an item in a menu, or as a shared data structure.

partition. (1) A fixed-size division of storage. (2) In VSE, a division of the virtual address area that is available for program processing. (3) On an IBM Personal Computer fixed disk, one of four possible storage areas of variable size; one can be accessed by DOS, and each of the others may be assigned to another operating system.

partitionable CPC. A CPC that can be divided into 2 independent CPCs. See also *physical partition*, *single-image mode*, *MP*, *side*.

partitioned data set (PDS). A data set in direct access storage that is divided into partitions, called *members*, each of which can contain a program, part of a program, or data.

passive monitoring. In SA OS/390, the receiving of unsolicited messages from OS/390 systems and their resources. These messages can prompt updates to resource status displays. See also *active monitoring*

path. Communication link (either NetView or PS/2) between a processor and console

PCE. Processor controller. Also known as the "support processor" or "service processor" in some processor families.

PDS. Partitioned data set.

Personal System/2* (PS/2) computer. A family of microcomputers manufactured by IBM Corporation.

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 OS/390 enterprise. See *SA OS/390 policy*

policy database. The database where the automation policy is recorded.

POR. Power-on reset.

port. (1) System hardware to which the I/O devices are attached. (2) On an ESCON switch, a port is an addressable connection. The switch routes data through the ports to the channel or control unit. Each port has a name that can be entered into a switch matrix, and you can use commands to change the switch configuration. (3) An access point (for example, a logical unit) for data entry or exit. (4) A functional unit of a node through which data can enter or leave a data network. (5) In data communication, that part of a data processor that is dedicated to a single data channel for the purpose of receiving data from or transmitting data to one or more external, remote devices. (6) power-on reset (POR) (7) A function that re-initializes all the hardware in a CPC and loads the internal code that enables the CPC to load and run an operating system.

PP. Physically partitioned (configuration).

PPT. Primary POI task.

primary host. The base program at which you enter a command for processing.

primary POI task (PPT). The NetView subtask that processes all unsolicited messages received from the VTAM program operator interface (POI) and delivers them to the controlling operator or to the command processor. The PPT also processes the initial command specified to execute when NetView is initialized and timer request commands scheduled to execute under the PPT.

primary system. A system is a primary system for an application if the application is normally meant to be running there. SA OS/390 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 OS/390 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.

processor operations control file. Named by your system programmer, this file contains configuration and customization information. The programmer records the name of this control file in the processor operations file generation panel ISQDPG01.

processor operations workstation. This 3270-type display echoes commands run by automation and relays commands from the focal point system to the target processor, and messages from the target processor to the focal point system. The display can also be used as operator consoles. The workstation attaches to a processor through the processor controller.

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 (.

ProcOps. See *Processor Operations*

product automation. Automation integrated into the base of SA OS/390 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 *Application* representing a processor operations target system.

PR/SM. Processor Resource/Systems Manager.

PS/2 computer. In SA OS/390, an IBM PS/2 computer that runs the OS/2 program and that contains a communication adapter. The PS/2 computer, which is connected to one or more target systems, contains the distributed feature code of SA OS/390. An SA OS/390 workstation may be either an IBM PS/2 computer or an IBM PS/55 computer, unless otherwise stated.

PS/2 LU name. In SA OS/390, the focal-point SNA logical unit name of the workstation.

PS/2 name. In SA OS/390, the logical name of the workstation.

PS/2 ports. On a PS/2 workstation, a port is a cable connector on a 3270 or ASCII logic card that attaches it to a target system. The port is associated with the screen handler for the PS/2 and identified by a port letter.

PU. Physical unit.

R

remote system. A system that receives resource status information from an SA OS/390 focal point system. An SA OS/390 remote system is defined as part of the same SA OS/390 enterprise as the SA OS/390 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 OS/390, any OS/390 application, OS/390 component, job, device, or target system capable of being monitored or automated through SA OS/390.

Resource Access Control Facility (RACF). A program that can provide data security for all your resources. RACF protects data from accidental or deliberate unauthorized disclosure, modification, or destruction.

resource group. A physically partitionable portion of a processor. Also known as a *side*.

Resource Monitoring Facility (RMF). 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 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 OS/390-provided automation that monitors subsystems to ensure that they are running. If a subsystem fails, SA OS/390 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 OS/390 automation focal point system. See *focal point system*

SA OS/390 customization dialog. An ISPF application through which the SA OS/390 policy administrator defines policy for individual OS/390 systems and builds automation control data.

SA OS/390 customization focal point system. See *focal point system*

SA OS/390 enterprise. The group of systems and resources defined in the customization dialog under one enterprise name. An SA OS/390 enterprise consists of connected OS/390 systems running SA OS/390.

SA OS/390 focal point system. See *focal point system*.

SA OS/390 policy. The description of the systems and resources that make up an SA OS/390 enterprise, together with their monitoring and automation definitions.

SA OS/390 policy administrator. The member of the operations staff who is responsible for defining SA OS/390 policy.

SA OS/390 satellite. If you are running two NetViews on an OS/390 system to split the automation and networking functions of NetView, it is common to route alerts to the Networking NetView. For SA OS/390 to process alerts properly on the Networking NetView, you must install a subset of SA OS/390 code, called an *SA OS/390 satellite* on the Networking NetView.

SCA. In SA OS/390, system console A, the active system console for a target hardware. Contrast with *SCB*.

SCB. In SA OS/390, system console B, the backup system console for a target hardware. Contrast with *SCA*.

screen. Deprecated term for display panel.

screen-based processor. See *screen-oriented processor*.

screen handler. In SA OS/390, 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.

screen-oriented processor. A central processor complex that interacts with human operators through line-mode display consoles. For System/390 and System/370 processors, the display consoles use either an IBM 3270 data stream or an ASCII data stream. Contrast with *OCF-based processor*.

SDF. Status display facility.

SDLCL. Synchronous data link control.

SDFS. 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 OS/390 does not start the application on its secondary systems.

server. A server is a workstation, that shares resources, which include directories, printers, serial devices, and computing powers.

service 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 OS/390 policy setting that determines when to notify the operator of deteriorating service for a resource. See *alert threshold* and *warning threshold*

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.

session. In SNA, a logical connection between two network addressable units (NAUs) that can be activated, tailored to provide various protocols, and deactivated, as requested. Each session is uniquely identified in a transmission header by a pair of network addresses identifying the origin and destination NAUs of any transmissions exchanged during the session.

session monitor. The component of the NetView program that collects and correlates session-related data and provides online access to this information. The successor to NLDM.

shutdown automation. SA OS/390-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.

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 OS/390 message that directly responds to a command. Contrast with *unsolicited message*.

SRPI. Server-Requester Programming Interface.

SSCP. System services control point.

SSI. Subsystem interface.

start automation. SA OS/390-provided automation that manages and completes the startup process for subsystems. During this process, SA OS/390 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 OS/390 that displays status of resources such as applications, gateways, and write-to-operator messages (WTOs) 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 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 OS/390 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 OS/390 enterprise definition and is used for monitoring purposes.

SubGroup entry. A construct, created with the customization dialog, 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 OS/390, an OS/390 application or subsystem defined to SA OS/390.

subsystem interface. The OS/390 interface over which all messages sent to the OS/390 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 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 non-switched links. The configuration of the link connection may be point-to-point, multi-point, 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.

SysOps. See *System Operations*

sysplex. A set of OS/390 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, OS/390 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 which 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 OS/390 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 OS/390 generic name for the IBM Sysplex Timer (9037).

system. In SA OS/390, system means a focal point system (OS/390) or a target system (MVS, VM, VSE, TPF or CF).

System Automation for OS/390. The full name for SA OS/390.

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 OS/390, the hardware system console for processor controllers or service processors of processors connected using SA OS/390. In the SA OS/390

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 OS/390 you can select SDSF from a pull-down menu to see the resources' status, view the OS/390 system log, see WTOR messages, and see active jobs on the system.

System entry type. A construct, created with the customization dialog, 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 OS/390 system.

system operations. The part of SA OS/390 that monitors and controls system operations applications and subsystems such as NetView, SDSF, JES, RMF, TSO, RODM, ACF/VTAM, CICS, IMS, and OPC.

system services control point (SSCP). In SNA, the focal point within an SNA network for managing the configuration, coordinating network operator and problem determination requests, and providing directory support and other session services for end users of the network. Multiple SSCPs, cooperating as peers, can divide the network into domains of control, with each SSCP having a hierarchical control relationship to the physical units and logical units within its domain.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks.

System/390 microprocessor cluster. A configuration that consists of central processor complexes (CPCs) and may have one or more integrated coupling facilities.

T

TAF. Terminal access facility.

target. A processor or system monitored and controlled by a focal point system.

target control task. In SA OS/390, target control tasks process commands and send data to target systems and workstations through communications tasks. A target control task (a NetView autotask) is assigned to a target system when the target system is initialized.

target hardware. In SA OS/390, the physical hardware on which a target system runs. It can be a single-image or physically partitioned processor. Contrast with *target system*.

target system. (1) In a distributed system environment, a system that is monitored and controlled by the focal-point system. Multiple target systems can be controlled by a single focal-point system. (2) In SA OS/390, a computer system attached to the focal-point system for monitoring and control. The definition of a target system includes how remote sessions are established, what hardware is used, and what operating system is used.

task. (1) A basic unit of work to be accomplished by a computer. (2) In the NetView environment, an operator station task (logged-on operator), automation operator (autotask), application task, or user task. A NetView task performs work in the NetView environment. All SA OS/390 tasks are NetView tasks. See also *communications task*, *message monitor task*, and *target control task*.

telecommunication line. Any physical medium, such as a wire or microwave beam, that is used to transmit data.

terminal access facility (TAF). (1) A NetView function that allows you to log onto multiple applications either on your system or other systems. You can define TAF sessions in the SA OS/390 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 OS/390 automation performs a predefined action. See *alert threshold*, *warning threshold*, and *error threshold*.

time of day (TOD). Typically refers to the time-of-day clock.

Time Sharing Option (TSO). An optional configuration of the operating system that provides conversational time sharing from remote stations. It is an interactive service on OS/390, MVS/ESA, and MVS/XA.

Time-Sharing Option/Extended (TSO/E). An option of OS/390 that provides conversational time-sharing 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.

TPF. Transaction processing facility.

transaction processing facility (TPF). A high-availability, high-performance system, designed to support real-time, transaction-driven applications (such as an airline reservation system). The specialized architecture of TPF is intended to optimize system efficiency, reliability, and responsiveness for data communication and database processing.

TPF provides real-time inquiry and update to a large, centralized database, where message length is relatively short in both directions, and response time is generally less than three seconds. Formerly known as the Airline Control Program/Transaction Processing Facility (ACP/TPF).

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 which have been defined to SA OS/390. 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. On top of requests, triggers, in combination with events, are used to control the starting and stopping of applications in a single system or a Parallel Sysplex. Triggers act as inhibitors for the requested action.

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 OS/390 customization panels.

TSO/E. TSO Extensions.

U

UCB. The unit control block; an MVS/ESA data area that represents a device and which is used for allocating devices and controlling I/O operations.

unsolicited message. An SA OS/390 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.

V

Virtual Machine/System Product (VM/SP). An IBM licensed program. It is an operating system that manages the resources of a real processor to provide virtual machines to end users. As a time-sharing system control program, it consists of the virtual machine control program (CP), the conversational monitor system (CMS), the group control system (GCS), and the interactive problem control system (IPCS).

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/SP. Virtual Machine/System Product.

VM/SP HPO. Virtual Machine/System Product High Performance Option.

VSE. Virtual Storage Extended.

VTAM. Virtual Telecommunications Access Method.

W

warning threshold. An application service value that determines the level at which SA OS/390 changes the associated icon on the NMC workstation to the warning color. See *alert threshold*

workstation. In SA OS/390 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 OS/390 operator console. This request is made by an application and is handled by the WTO processor, which is part of the OS/390 supervisor program.

write-to-operator-with-reply (WTOR). A request to send a message to an operator at the OS/390 operator console which 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 OS/390 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 *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

3090/ACA/ISCF. 3090 Automated Console Application for Inter-System Control Facility.

3090/Automated Console Application/ISCF (3090/ACA/ISCF). A program offering that runs as a NetView application in conjunction with ISCF. It provides some built-in automation and an enhanced user interface. 3090/ACA/ISCF is limited to 3090 OS/390 targets.

3270 connection card. A 3270 emulator card installed in a workstation used to connect the workstation to the target system. Four 3270 connection cards can be installed in a workstation.

37x5. A 3705, 3725, or 3745 communication controller.

3x74. A 3174 or 3274 subsystem control unit. A control unit to which 3270-family display stations are attached.

Index

Special Characters

&AOCCLONE. 105, 146

Numerics

308x

CP states 131
target system basic
characteristics 118

3090

CP states 128
target system basic
characteristics 118

3270

connection card 222
connection cards 213, 221

3270 port description 222

4381

CP states 131
target system basic
characteristics 122

9021

CP states 128
target system basic
characteristics 118

9121

CP states 128
target system basic
characteristics 118

9672

target system basic
characteristics 118, 120

9674

target system basic
characteristics 121

A

ABENDS (JES3) 185

ACF 11, 247

ACFFQRY 81

ACORESTART 164

Action Message Retention Facility 10

actions on policy databases 23

Adjacent Domain policy definition 271

ADJACENT NETVIEW policy item 271

adjacent paths 334

alternate adjacent policy definition 271

alternate configuration support 361

alternate configurations 362

alternate data sets 97

AMC 11

AMC file 290, 298

AMRF 10

AMRF buffer 259

AOCCLONE 105, 144, 146

AOCCLONE clone IDs 102

AOFADMON 112

AOFAJMON 112

AOFAPMON 112

AOFARCAT 265

AOFATMON 112

AOFPCSM 112

AOFMINORCHECK 47

AOFOPF sample 265

AOFUXMON 112

AOFWRKxx 264

AOFWRKxx operators 269

APG automation name 342

APPC automation 10

APPC/MVS

automation 22

APPLGROUP INFO policy item 136, 137

application (entry type) 3

application class 143

Application Defaults 5

Application Defaults policy item 106

APPLICATION DEFAULTS policy
item 106

Application Defaults policy object 262

application definition file 302

application group 3

nature 134

nature of 56

Sysplex Application Group 55

System Application Group 55

application group resource name 56

APPLICATION GROUPS policy item 95,
105

APPLICATION INFO policy item 148

application instance 143

application monitor routine 111, 151, 155

Application policy object 141

creating a new 141

policy items 147

application resource name 54

application resources

automatically linking 60

generating 58

application type 143

ApplicationGroup policy object 132

creating a new 133

policy items 135

APPLICATIONS policy item 136, 138

ARM

MOVE group for 340

ARM element name 339

ASCB chaining 151

with application automation 156

ASCH automation 10, 22

ASCII operator console 215

ASCII Port

definition 225

ASCII port description 225

ATTEND TIME policy item 251

ATTN code 227

AUTBASE 264

AUTCON 265

AUTGSS 265

AUTHB 265

AUTJES 265

AUTLOG 264

AUTMON 265

AUTMSG 264

AUTNET1, AUTNET2 264

AUTO MSG CLASS policy item 106
description 106

Auto Msg Classes 5

Auto Msg Classes policy object 269

Auto Operators 5

Auto Operators policy item 107

AUTO OPERATORS policy item 106

Auto Operators policy object 263

AUTO1 264

AUTO2 264

automated function 166, 172, 175

Automatic IPL 123

Automatic Restart Manager 339

defining element names 144

element name 339

MOVE group for 340

using clone values 144

automation

goal driven 31

Automation automation flag 43

automation concepts 31

automation configuration 13, 299

ACF 298

AMC file 298

automation control file 298

automation manager configuration
file 298

automation console policy item 105

AUTOMATION CONSOLE policy
item 105, 109

automation control file 11, 298, 299

building 292

distributing 298

migrating 23

automation flag

Automation 43

Initstart 44

Recovery 43

Restart 44

Shutdown 44

Start 44

types 43

automation flags 42

automation manager flags 46

extended 150

global automation flag 46

hold flag 46

minor resource flags 47

resource status 46

setting 45

setting application defaults 263

automation flags policy definition 148

AUTOMATION FLAGS policy item 148,
261, 262, 263

automation info 148

AUTOMATION INFO policy item 148,
150

- automation manager configuration
 - file 11, 290, 298, 299
 - building 292
- automation manager flags 46
- automation manager global automation flag 46
- automation name 56, 135
- automation networks 327
 - definition 270
- automation operators 263
- automation operators policy definition 264
- automation policy
 - building 11
 - distributing 11
- AUTOMATION policy item 105
- automation procedure 10
- automation programmer xv
- automation resource name 135
- automation setup policy definition 105
- AUTOMATION SETUP policy item 105, 111
- automation status file 391
- Automation Timeout policy item 106
- AUTOMATION TIMEOUT policy item 106
- Automation Timers policy item 105
- AUTOMATION TIMERS policy item 105
- autotask 5
- AUTREC 265
- AUTRPC 265
- AUTSHUT 265
- AUTSYS 264
- AUTWRKxx 264
- AUTXCF 265
- AUTXCF2 265

B

- BASEOPER 264
- BASIC 56, 134
- batch build 298
 - data sets 357
- batch build processing
 - defining data sets 356
- batch migration 324
- baud rate 226
- behaviour of application groups
 - ACTIVE 135
 - PASSIVE 135
- BUILD 293
 - BUILD command 290
 - building NetView and VTAM information 301
 - building processor control file 291
 - building PS/2 profile information 303
 - building system operations control files 292
 - building the automation control file 292, 293
 - building the automation manager configuration file 292
 - building the processor control file 290
 - description 289, 308

- BUILD (*continued*)
 - on the *Policy Data Base Selection* panel 23
- BUILD automation policy 11
- BUILD error messages 296
- Build Functions Menu 290
- build processing
 - with BUILD command 293
- BUILDF command 293

C

- cancel input 17
- Card
 - operator console 215
 - system console 213, 214
- CDS 96
- central storage
 - 308x Processor 131
 - 4381 processor 131
 - 4381 Processor 132
 - 9021, 9121, and 3090 processors 129
 - screen-oriented processor 212
- chaining 160
- channel path ID (CHPID) 217
- CHANNEL PATH IDS policy item 211, 216
- CHPID
 - target hardware 217
- CHPID assignments 216
- CICS automation 230
- CICS CONNECTION policy item 186, 233
- CICS CONTROL policy item 186, 187
- CICS link definitions 230, 232
 - policy items 233
- CICS link monitor periods 230, 234
- CICS monitoring periods 234
- CICS state/action table 231
- CICS state/action tables 230, 231
- CICS subsystems
 - policy items 185
- CLISTS policy item 276
- clone ID
 - assigning 105
 - Automatic Restart Manager 339
 - substitution in job names 105
- clone IDs
 - assigning 102
- clone values 144
- clone variables 146
- cloned element names 145
- cloning the job name 144
- CMD processing 165
- code processing 167
- Command Handler 371
- command interval 223
- command prefix 153
- command processing 165
- command requests 372
 - maintenance start 373
 - Start/Stop tablespace 378
 - terminate threads 376
- command routing mechanism 89
- command shutdown processing
 - for INIT and FINAL 175
 - for NORM, IMMED and FORCE 175

- commands
 - communications path 228
- commands and requests 31
- common routine 10
- communications path 4
 - description 228
- Communications Path
 - policy object 218
- communications path commands 228
- Communications Path policy object
 - creating a new 218
 - policy items 218
- communications task 5
 - commands 282
 - control parameters 280
- Communications Task policy object 279
 - creating a new 279
- concurrent user access 17, 349
- condition
 - relationship condition 161
 - shutdown 206
 - shutdown condition 48
 - startup 206
 - startup condition 48
 - unset condition 202
- condition (for relationships) 36
- CONDITION policy item
 - for a trigger 206
 - for an event 202
- configuration
 - of automation 13, 299
- configuration focal point system 13, 299
- configuration support, alternate 361
- connection monitoring 365, 380
 - CICS 366
 - IMS 366
- considerations for automation 10
- console
 - interval 223
 - model 224
 - type 222
- console areas definition 258
- console definition 257
- console ID policy definition 110
- console name 110
- console size definition 258
- Control File
 - allocate 291
 - message log 292
 - name 91, 291
- control file and control file log 291
- CONTROL PARAMETERS policy item 280
- controlled PS/2 List 282
- CONTROLLED PS/2S policy item 280, 282
- controller details
 - OPC 242
- CONTROLLER DETAILS policy item 107
- conversion
 - of initial policy database 341
 - selective policy database conversion 348
- conversion function 341
- Convert function
 - APG automation name 342

- converted data 342
- COPY policy item 27
- copying policy 28
- country code 223
- couple data set 96
- CP LPAR States panel 129
- CP states for
 - 308x 131
 - 3090, 9021, 9121 128
 - 4381 131
- CP STATES policy item 211
- CPCB (command processor control block) 393
- creating a policy database 18
- critical event monitoring 383
- Critical Event Monitoring 365
- critical events 365
- critical threshold 39, 40
- CRM 89
- customization dialog 1
 - cancel input 17
 - changing 351
 - features 16
 - help 351
 - introduction 2
 - invocation 14
 - primary menu 14
 - starting 14
 - using 13, 14, 16
- Customization Dialog Primary Menu 14

D

- data length 226
- DB2 automation 229
- DB2 Automation 365
- DB2 Automation installation 366
- DB2 CONTROL policy item 190
- DB2 subsystems
 - policy items 190
- DB2 System defaults 229, 230
- DB2 System Defaults
 - policy items 230
- DB2 SYSTEM DEFAULTS policy item 107, 230
- DEFAULTS sample database 20
- dependency relationships 36
- dependent resource 36, 137, 149, 157
- DISPACF 81
- DISPASGN command 229, 285
- DISPUNAP command 229, 284
- distribute automation policy 11
- distributing
 - system operations control files 298
 - the automation control file 298
- DISTRPS2 command 228, 282
- drain, JES2 182
- DSIOPF data set 264
- DSIPARM
 - description 394
- DSIPARM (for gateways) 333
- dual NetView 109

E

- E-T DATA policy item 286
- element name 339
 - in Automatic Restart Manager 144
- element names
 - in Automatic Restart Manager 339
- end of message code 227
- enter code 227
- enterprise 2
- enterprise definition 13, 88
- Enterprise policy object 88, 201
 - policy items 88
- entry name creation 314
- entry type 2, 25
 - short description 2
- environment definition, system
 - defaults 262
- ENVIRONMENT policy item 262
- error messages
 - BUILD 296
- error thresholds 39
- event 47
- event check time interval 281
- event-driven functions 365, 379
 - connection monitoring 380
 - critical event monitoring 383
- events 3
 - scenario 78
- events and conditions 48
- Events policy object
 - creating a new 201
 - policy items 202
- exit names policy definition 113
- expanded storage 132, 212
- extended automation flags 150
- extended recovery facilities 235
- external shutdown 156
- external startup 156
- EXTERNALLY 159
- EXTSTART status 339

F

- FALLBACK 179
- FALLBACK status
 - support 179
 - using 179
- fast path 16
- favored LPAR 130
- feature keywords
 - cross reference with customization dialog 345
- FINAL shutdown 174
- focal point
 - name 91
- focal point NetView domains policy
 - definition 271
- focal point services 270
- focal point system definition 328
- FORCE shutdown 174
- FORWARD policy item 271
- frequent threshold 40
- FULL SESSIONS policy item 272
- functional names for auto operators 264

G

- GATAOF01 265
- gateway
 - inbound 329
 - outbound 329, 330
- gateway monitor time policy
 - definition 111
- gateway policy definition 272
- GATEWAY policy item 272
- gateway session 330
 - DSIPARM definitions 333
 - securing 332
- gateway session definition 329
- GATOPER 265
- GENERATED RESOURCES policy item 137, 140, 150
- generating PS/2 profiles 305
- generic routine 10
- global automation flag 46
- goal driven automation 31
- graphic workstation description 399
- graphical interface 9
- GROUP INFO policy item 94
- Group policy object 92
 - creating a new 92
- grouping
 - scenario 32
 - support 31
- grouping resources 9
- groups 2
- groups and resources 31
- GSSOPER 265
 - SHUTDOWN 269
 - STARTUP 269

H

- HASPARENT 159
- HASPARENT relationship 37
- HBOPER 265
- heartbeat
 - explanation 103
 - heartbeat interval 101
 - missing heartbeat delay 101
- heartbeat interval 101
- help information 351
- hold flag 46
- HWMCA
 - description 395

I

- Image profile name 121
- Image Profile Name 119
- IMMED shutdown 173
- IMS automation 235
- IMS CONTROL policy item 192
- IMS ENVIRON policy item 192, 194
- IMS extended recovery facilities 235
- IMS RESOURCE NAMES policy item 239
- IMS STATE/ACTION policy item 236
- IMS state/action table 236
- IMS STATUS FILE policy item 238
- IMS subsystems
 - policy items 191

- IMS XRF resources 235, 238
- IMS XRF RESOURCES policy item 107
- IMS XRF status files 235, 237
- inbound retry count 223
- Includes 5
- INCLUDES policy item 108
- Includes policy object 286
- infrequent threshold 40
- ING.CUSTOM.AOFTABL 289, 356
- INGAUTO 46
- INGDLG 14
- INGEIO 396
- INGERMFBA automation 22
- INGEVENT 47
- INGREQ 31
- INGSEND PARMS policy item 89
- INGTHRES 39
- INGXSG 98
- INITSEL 14
- Initstart automation flag 44
- interested operator list 127
- invoking the customization dialog 14
- IOCDs 118
- IPL
 - address 123
 - MVS target system 123
 - parameter 123
 - screen-oriented target system 122
 - VM target system 125
- ISPF temporary data sets 357
- ISQDPN01 panel 301
- ISQMTSYS 113
- IXCARM macro invocations 339
- IXCMIAPU 339

J

- JES
 - spool recovery 181
 - spool status 180
 - spool usage 181
- JES subsystem 179
- JES2
 - automation 179
 - procedures 10, 22
 - recovery parameters 183
 - spool recovery 179
 - automation 10
 - establishing 182
 - from full condition 182
- JES2 DRAIN 182
- JES2 spool recovery 180
- JES2 SPOOLFULL 182
- JES2 SPOOLSHORT 182
- JES2 subsystems
 - policy items 181
- JES3
 - ABENDS 185
 - automation 22
 - procedures 10
 - MONITOR 185
 - SPOOL
 - automated startup and shutdown 10
 - control 184
 - SPOOLFULL 185
 - SPOOLSHORT 185

- JES3 subsystems
 - policy items 184
- JESOPER 265
- job name
 - defined 144
 - substituting 105
- job type
 - description 151
 - MVS 151
 - NONMVS 151
 - policy definition 151
 - TRANSIENT 151

K

- keywords for features 345

L

- LAN passthru function 292
- line command functions 370
 - Command Handler 371
- line mode functions 365
- line mode invocation 365
- link definitions (CICS) 230, 232
- link monitor periods (CICS) 230, 234
- LINK TO CLASS policy item 147
- LINK TO INSTANCES policy item 148
- LLA automation 22
- Load profile Name for support element 9674 122
- Load Profile Name for support element 9672 in (Mode 119
 - OCF-based processor in basic mode 121
- LOCATE 24
- LOGMODE 281
- LOGOPER 264
- LOGREC
 - automation policy definition 10
 - thresholds policy 253

M

- maintenance start 365, 373
- MAKEAVAILABLE 159
- MAKEUNAVAILABLE 159
- MCSFLAGS policy definition 110
- MEMBER OF policy item 150
 - for application groups 137, 141
 - for applications 177
- memory
 - target hardware (308x/9121/9021/3090) 132
 - target processor (308x/9121/9021/3090) 212
- MERGE 24
- message
 - forwarding path, defining 327
 - monitor task 91
 - table policy definition 113
- message class 114
- message classes 115, 116, 269
- message IDs
 - reserved 164
- message prefix 153
- message processing 163
 - CMD processing 165
 - code processing 167
 - reply processing 166
 - scenario 82
 - user defined data 168
 - with user-defined data 82
- messages
 - policy definition 149
 - response 123, 125
 - routing 128
- MESSAGES/USER DATA policy item 149, 163, 254
- MIGRATE command
 - Policy Data Base Selection 23
- migration
 - automation control file 309
 - entry name creation 314
 - include files 317
 - job for batch migration 324
 - product automation definition 317
 - selective migration 311
- migration functions 309
- minor resource flags 47
 - as subsidiary to major resource 254
 - for applications 149
- MINOR RESOURCE FLAGS policy item 149, 254, 259
- minor resources 259
 - AOFMINORCHECK 47
- missing heartbeat 101
 - processing 103
- missing heartbeat delay 101
- missing heartbeat processing 103
- MONITOR (JES3) 185
- monitoring
 - tape mounts 250
- Monitoring, Sysplex Timer 96
- MONITORING PERIOD policy item 233, 235
- monitoring periods for CICS 234
- MONOPER 265
- MOVE 56, 134
- MOVE group 67
- MOVE group for ARM 340
- MOVED 179
- MOVED status
 - Automatic Restart Manager 339
 - automation 339
 - using 179
- MSGOPER 264
- MULTISYS sample database 21
- MVS Component 5
- MVS Component policy item
 - to choose sets to be part of system object 106
- MVS COMPONENT policy item 106
- MVS Component policy object
 - dump data sets 10
 - using 252
- MVS console policy definition 268
- MVS job 151
- MVS messages, responding to 254
- MVS Target System IPL Information panel 123
- MVSCOMP defaults 5
- MVSCOMP Defaults policy item 106

MVSCOMP DEFAULTS policy item 106
 MVSCOMP Defaults policy object 260
 MVSCONSi 265
 MVSDUMP
 MVS Component policy object 252
 thresholds policy 253

N

naming restrictions 293
 nature
 BASIC 56
 MOVE 56
 SERVER 56
 nature of an application group 56
 nature of application groups
 BASIC 134
 MOVE 134
 SERVER 134
 NBUF policy definition 255
 NETOPER 264
 NETV_SA automation 22
 NETV_SSI automation 22
 NetView
 adjacent paths 334
 building information 301
 domain name 109
 nomination before BUILD 11
 operator definition file 302
 operator profile 302
 Operators 268
 policy item 105
 specifying domain name 109
 task definition file 302
 NetView and VTAM information 301
 NetView communications path
 definition panel 227
 NetView connection 227
 NetView message automation table
 updating 84
 NETVIEW policy item 105, 108
 Network 5
 Network policy item 106
 NETWORK policy item 106
 Network policy object 270
 networks automation
 definition process 327
 NMC focal point system 394
 NNT sessions 5
 and minor resources 105
 definition 274
 user 336
 NNT SESSIONS policy item 105
 NNT Sessions policy object 273
 NONMVS job 151
 NORM shutdown 173
 NORMCONS policy definition 257
 notification forwarding 328
 notification operator 114
 notification operators 107
 Notify Operators policy item 107
 NOTIFY OPERATORS policy item 107,
 113
 NVC DEFINITION policy item 219
 NVC PATH NAME policy item 211

O

OP CONS CONNECTIONS policy
 item 211, 215
 OPC automation 240
 OPC CONTROL policy item 200
 OPC controller details 240, 242
 policy items 242
 OPC special resources 240, 244
 OPC subsystems
 policy items 199
 OPC system details 240
 OPC SYSTEM DETAILS policy item 107
 OPC workstation domains 240, 245
 OPCA DOMAINID policy item 245
 OPCA PCS policy item 242, 243
 OPCA ENVIRON policy item 241
 operator xv
 definition file 302
 profile 302
 operator console
 interested operator 128
 Operator console
 assignment 215
 Operator Console
 assignment 215
 operator console connection 215
 Operator Console Connections
 panel 215
 OPERATORS policy item 264
 OS/390 job name defined 144
 outbound gateway 330
 output data set 295
 overview 1

P

panelid 6
 parity 226
 PASSTHRU idle time limit 281
 password mask 273
 password protection, SA OS/390 332
 PC
 communications path 219
 control port 220
 focal point port 220
 LU name 282
 name 282
 Reference Diskette 213, 214
 system console connection 213
 Policy Data Base Selection panel 18
 policy database 13
 adding to index 18
 available actions 23
 converting 310
 creating 18
 creating a new 17
 reports 358
 samples 20
 sending to remote nodes 361
 using existing 22
 policy database conversion 341
 policy item 6
 COPY 27
 WHERE USED 28
 policy items
 create or modify 353

policy items (*continued*)
 general policy items 27
 policy object 2, 6, 24, 25
 create a new 25
 policy objects
 Application 141
 Application Defaults 262
 ApplicationGroup 132
 Auto Msg Classes 269
 Auto Operators 263
 Communications Path 218
 Communications Task 279
 copying 28
 copying definitions 28
 deleting 26
 Enterprise 88, 201
 Group 92
 Includes 286
 MVS Component 252
 MVSCOMP Defaults 260
 Network 270
 NNT Sessions 273
 Processor 208
 Product Automation 229
 Resident CLISTS 275
 Service Periods 203
 Status Details 276
 SubGroup 98
 Sysplex Application Group 132, 135
 System 99
 System Application Group 135
 System Defaults 261
 Tape Attendance 250
 Timeout Settings 248
 Timers 246
 Triggers 205
 User E-T Pairs 286
 PolicyDB 18, 309
 poll
 interval 281
 timer 223
 port configurations 220
 PORT CONFIGURATIONS policy
 item 219
 port description
 3270 222
 ASCII 225
 Port letter 222
 POSTSTART 170
 PPIOPER 265
 preference value
 default 134
 preference values 139
 preferences 72
 PREPAVAILABLE 159
 PREPUNAVAILABLE 159
 primary data sets 97
 primary JES policy definition 111
 primary memory 131, 132, 212
 Primary Menu 14
 primary system association 178
 priority of requests 205
 processor control file 11, 290
 LAN passthru function 292
 PROCESSOR INFO policy item 211
 processor operations
 CP States 131

- processor operations (*continued*)
 - dedicated personal computer 402
 - PS/2 communications path 219
- PROCESSOR_OPERATIONS
 - automation 22
- Processor Operations CP States 131
- processor operations PC 208
- processor operations PC description 219
- Processor Operations policy item 108
- PROCESSOR_OPERATIONS policy
 - item 108, 117
- Processor Operations Port Configurations
 - panel 220
- processor operations PS/2 208
- Processor Operations PS/2
 - Distribution 282
- Processor Operations Target Definition
 - panel 117, 118
- PROCESSOR OPS INFO policy item 89, 91
- processor policy item 108
- PROCESSOR policy item 108
- Processor policy object 208
 - creating a new 209
 - policy items 210
- processors 3
- procname 102
- ProcOps PC 208
- product automation 4, 402
 - CICS 230
 - DB2 229
 - IMS 235
- Product Automation
 - policy object 229
- PRODUCTS sample database 21
- profile
 - concept 303
 - contents 304
 - generating 305
 - sample processor operations 303
- profile information before BUILD 11
- propagation of requests 37
- proxy resource 113
- PS/2 208
 - communications path 219
- PS/2 CHARACTERISTICS policy
 - item 219
- PS/2 command wait limit 281
- PS/2 computer
 - customizing for standalone function 306
- PS/2 poll response limit 281
- PS/2 profile 303
 - building profile information 303
 - contents 304
 - generating 305
- PS/2 profile information 11

R

- RACF 332
- RECOPER 265
- recovering WTOBUF 255
- Recovery automation flag 43
- Reference Diskette 213, 214
- relationship condition 161
- relationship type 158

- relationships 157
 - policy item for applications 148
- relationships concept 36
- RELATIONSHIPS policy item 137, 148, 157
- replicable applications 105
- replication 146
- reply processing 166
- reply shutdown processing 176
- REPORT 24
- reports
 - about ApplicationGroups 358
 - about Applications 358
 - about generated resources 358
- request 31
 - calculation of schedules 36
 - INGREQ 31
 - priority 31
- request propagation 37
- reserved message IDs 164
- Reset profile name for support element
 - 9672 in (Mode 119
 - 9674 121
 - OCF-based processor in basic mode 120
- Resident CLISTS 5
- RESIDENT CLISTS policy item
 - with minor resources 106
- RESIDENT CLISTS policy item 106
- Resident CLISTS policy object 275
- resource
 - dependent 36, 137, 149, 157
 - supporting 36, 137, 149, 157
- resource states
 - automation flags 46
 - description 46
 - list 46
 - SDF 47
- RESOURCE THRESHOLDS policy
 - item 186, 190, 192, 230
 - CICS 189
- resources
 - explanation 52
 - grouping 9
 - scenario 33, 51
- resources and groups 31
- RESOURCES policy item 136, 139
- Restart automation flag 44
- restart option policy definition 155
- Retry count 281
- RMF 10
- roll mode definition 257
- roll time definition 257
- route codes policy definition 110
- RPCOPER 265

S

- SA OS/390
 - enterprise
 - components 87
 - password protection 332
 - SA OS/390 topology manager 300
- SAF ENVIRONMENT policy item 272
- sample policy databases 20
 - DEFAULTS 20
 - MULTISYS 21

- sample policy databases (*continued*)
 - PRODUCTS 21
 - SYSPLEX 21
- schedule
 - automatic calculation 36
 - using 77
- schedule override 77
- screen-oriented processor policy
 - items 211
- Screens policy item 107
- SCREENS policy item 107
- SDF
 - automation control file entry 277
 - highlight 278
 - priority policy definition 278
 - root name policy definition 113
- SDF DETAILS policy item 277
- Secondary OCF Name 120, 122
- secondary system association 178
- select migration objects 311
- SEND COMMAND OPERS policy
 - item 88, 89
- sending policy databases 361
- SERVER 56
- SERVER group 74
- service period 36, 136, 204
 - linking to application 148
 - using 77
- SERVICE PERIOD policy item 136, 148
- service periods 3
- Service Periods policy object 203
- service window 36, 204
- SERVICE WINDOW policy item 203
- Session start time limit 281
- SESSIONS policy item 274
- setting automation flags 45
- settings menu 352
- Settings option 351
- shut delay policy definition 156
- SHUTDOWN
 - GSSOPER 269
- Shutdown automation flag 44
- shutdown condition 47, 48, 49, 206, 208
- SHUTDOWN condition for a trigger 207
- SHUTDOWN policy item 149, 173
- shutdown processing
 - command 175
 - reply 176
- SHUTOPER 265
- single NetView 109
- SMF data sets 10
- SMFDUMP
 - MVS Component policy object 252
 - thresholds policy 253
- SPECIAL RESOURCE policy item
 - OPC 244
- SPECIAL RESOURCES policy item 243
- spool recovery 181
- spool status 180
- spool usage 181
- SPOOLFULL 180, 182
- SPOOLFULL (JES3) 185
- SPOOLSHORT 180, 182
- SPOOLSHORT (JES3) 185
- standalone function, customizing PS/2
 - computer 306
- standard application set 22

- standard Groups 93
 - policy items 93
- start
 - start on IPL 154
 - start on recycle 154
 - start timeout policy definition 155
- Start automation flag 44
- start/stop tablespace 365
- Start/Stop tablespace 378
- starting the customization dialog 14
- STARTUP 170
 - GSSOPER 269
- startup condition 47, 48, 49, 206
- STARTUP condition for a trigger 207
- startup condition list 208
- startup phases
 - POSTSTART 170
 - PRESTART 170
 - STARTUP 170
- STARTUP policy item 149, 169
- state/action table for CICS 231
- STATE ACTION TABLE policy item 186, 192
- STATE/ACTION TABLE policy item 231
- state/action tables (CICS) 230
- state/action tables (IMS) 235
- state/action tables for CICS 231
- Status Detail 5
- Status Details
 - display facility 276
 - policy object 276
- status file 391
- status forwarding path 103
- stop bit 226
- SUBGROUP INFO policy item 99
- SubGroup policy object 98
 - creating a new 98
 - policy items 99
- subgroups 2
- SUBGROUPS policy item 94
- subsystem
 - shutdown 149
 - startup 148, 149
- subsystem name 142
- subtype 144
- supporting resource 36, 137, 149, 157
- symbolic values in job names 146
- SYS CONS CONNECTIONS policy item 211, 213
- Sysclone ID 105
- SYSLOG
 - data sets 10
 - MVS Component policy object 252
- SYSLOG thresholds policy definition 253
- SYSNAME naming restrictions 293
- sysname policy item 113
- SYSOPER 264
- SysOps NetView 109
- SysOps NetView Network Name 109
- sysplex
 - creating 52, 53
- Sysplex Application Group 55, 132, 134
 - generating Sysplex Application Group resources 57

- Sysplex Application Group policy object 132
- sysplex group 92
- sysplex Groups 94
 - policy items 94
- Sysplex Policy Definition panel 95
- SYSPLEX policy item 95
- SYSPLEX sample database 21
- Sysplex Timer Monitoring 96
- System Application Group 55, 134
 - generating System Application Group resources 63
- System Application Groups 135
- system association policy definition 150
- SYSTEM ASSOCIATION policy item 150, 177
- system categories 178
- system console
 - active 213
 - assignment 127
 - backup 213
 - definition 213
 - interested operator 128
- System Console Connections 213
- System Defaults 5
- System Defaults policy item and minor resources 106
- SYSTEM DEFAULTS policy item 106
- System Defaults policy object 261
- SYSTEM INFO policy item 105
- system monitor time policy definition 111
- system operations control files 290, 298
 - building 292
 - distributing 298
- System policy object 99
 - creating a new 99
 - policy items 104
- system programmer xv
- systems, definitions 2
- SYSTEMS policy item
 - for Groups 94
 - for SubGroups 99

T

- TAF 270
- TAF full-screen sessions 335
- TAF full-screen sessions policy definition 272
- tape attendance 4
- Tape Attendance policy item with minor resources 106
- TAPE ATTENDANCE policy item 106
- Tape Attendance policy object 250
- tape mount, monitoring 250
- tape policy 250
- target
 - (mode 217
 - Automatic IPL 123
 - central storage 129
 - connections 213
 - control tasks 91
 - focal point name 91
 - image profile name 119, 121
 - IOCDS name 118
 - IPL address 123

- target (*continued*)
 - IPL parameter 123
 - load profile name 119
 - Load profile name 121, 122
 - MVS IPL information 124
 - operating system 118
 - reset profile name 119, 121
 - Reset profile name 120
 - secondary OCF name 120, 122
 - select components 117
 - time zones 127
- Target
 - connections 215
- target definition
 - 9021, 9121, 3090, 308x 118
 - 9672 processor 118
 - for 9674 processor 121
 - OCF-based processors in (mode 118
 - screen-oriented processor 117
- Target Definition panel
 - 308x processor 118
 - 3090 processor 118
 - 4381 processor 122
 - 9021 processor 118
 - 9121 processor 118
- target hardware
 - central storage 131, 132, 212
 - CHPID 216
 - CP states 131
 - expanded storage 132, 212
 - primary memory 131, 132, 212
 - vector processor 213
- Target hardware
 - 3270 connection card 127
 - processor operations CP states 211
- target system
 - time zones 127
- target system basic characteristics
 - 308x 118
 - 3090 118
 - 4381 122
 - 9021 118
 - 9121 118
 - 9672 118, 120
 - 9674 121
- Target System Connections panel 126
- task
 - definition file 302
 - message monitor 91
 - target control 91
- term delay policy definition 156
- Terminal Access Facility 270, 335
- terminate threads 365, 376
- TGWARN 180
- threshold
 - critical 39, 40
 - for applications 41
 - for MVS Components 41
 - for OS/390 components 39
 - frequent 40
 - infrequent 40
 - MVSCOMP 39
- THRESHOLD policy items 254
- thresholds
 - setting up 39
- THRESHOLDS policy item 149, 261, 262, 263

- time zone 127
- time zones 127
- TIMEAUTO 46
- timeout code 227
- timeout settings 4
- Timeout Settings policy object 248
- TIMEOUT VALUES policy item 248
- timeout variable 248
- timers 4
- TIMERS policy item 246
- Timers policy object 246
- To Domain policy definition 271
- TPF
 - ASCII connection 226
 - communication parameters 226
- TRANSIENT job 151
- transient rerun 152
- TRANSIENT1 automation 22
- TRANSIENT2 automation 22
- trigger 47, 48
 - link to an application group 136
 - linking to application 148
 - restriction 50
- TRIGGER policy item 136, 148
- trigger policy object 3
- triggers
 - scenario 78
- triggers and conditions 48
- Triggers policy object 205
- TSCF configuration 324
 - merge 324
- TSO/E automation 10, 22

U

- UCB (unit control block) 407
- unset condition 201, 202
- updating the NetView message
 - automation table 84
- user defined data processing 168
- User E-T Pairs 6
 - description 106
 - policy object 286
 - scenario 80
- USER E-T PAIRS policy item 105
- user groups
 - automation programmer xv
 - operator xv
 - system programmer xv
- user NNT sessions 336
- user-written functions 360

V

- vector processor 213
- VIEW 24
- VLF automation 22
- VM Target System IPL Information
 - panel 125
- VTAM
 - application definition file 302
 - automation 22
 - building information 301
- VTAM automation 10
- VTAM information 301
- VTAMLST 272

W

- Where Used panel 29
- WHERE USED policy item 28
- WLM resource name 145
- workstation domains
 - OPC 245
- WORKSTATION DOMAINS policy
 - item 108
- WTOBUF
 - automation policy definition 10
 - recovery 252
 - routines 254
- WTOBUF RECOVERY policy item 253, 255
- WWV receiver 226

X

- XCF group name
 - INGXSG, default 98
 - INGXSGxy 98
- XCFOPER 265
- XCFOPER2 265
- XOFF code 227
- XON code 227
- XRF 235
- XRF STATUS FILES policy item 107

Readers' Comments — We'd Like to Hear from You

System Automation for OS/390
Defining Automation Policy
Version 2 Release 1

Publication No. SC33-7039-02

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

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

Address

Company or Organization

Phone No.



Fold and Tape

Please do not staple

Fold and Tape



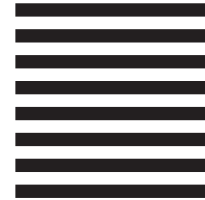
NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Deutschland Entwicklung GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany



Fold and Tape

Please do not staple

Fold and Tape



Program Number: 5645-006



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC33-7039-02



Spine information:



System Automation for OS/390 Defining Automation Policy

Version 2 Release 1

SC33-7039-02