System Automation for z/OS
Version 3 Release 4

Defining Automation Policy

IBM
Before using this information and the product it supports, read the information in Appendix C, “Notices,” on page 373.
## Contents

**Figures** ........................................ vii  
**Tables** ........................................... xi  
**Accessibility** ................................. xiii  
- Using assistive technologies ........ xiii  
- Keyboard navigation of the user interface .... xiii  
- z/OS information ............................... xiii  

**How to send your comments to IBM**  ........................................... xv

**About This Book** ................................. xvii  
- Who Should Use This Book ................. xvii  
- Notes on Terminology ....................... xvii  
- Where to Find More Information ............ xvii  
- The System Automation for z/OS Library.  xvii  
- Related Product Information .............. xvii  
- Using LookAt to look up message explanations xviii  

**Chapter 1. Introducing the Customization Dialog** ................................. 1

- How to Define Automation Policy: An Outline ........ 1  
- Entry Type Introduction .......................... 2  
- Policy Objects and Policy Items ............... 5  
- Considerations for the SA z/OS Graphical Interface 6  
- Grouping Resources Logically .................... 7  
- Considerations for Automation .................... 7  
- Assigning System Automation Symbols (AOCCLONE) .... 8  
- Using System Symbols and System Automation Symbols .... 9  
- Nested Classes .................................. 11  
- Class Hierarchy .................................. 12  
- Inheritance and Defaulting ...................... 13  

**Chapter 2. Using the Customization Dialog** ................................. 15

- How to Start the Customization Dialog ........ 16  
- Getting Help ................................... 17  
- How to Navigate in the Customization Dialog ... 17  
- How to Use an Entry Type as a Fast Path ........ 18  
- How to Use a Policy Item as a Fast Path ........ 19  
- Concurrent User Access ......................... 19  
- How to Cancel Input on a Panel ................ 19  
- How to Create New Policy Objects ............ 19  
- How to Rename a Policy Object ............... 21  
- How to Delete Policy Objects ................... 21  
- How to Delete a Single Policy Object ........... 22  

- How to Delete Several Policy Objects in One Go (Bulk Deletion) .......... 22  
- How to Find Data in a Policy Item ............. 23  
- Command and User Data Policies ............... 25  
- Common Policy Items .............................. 26  
- DESCRIPTION Policy Item ....................... 26  
- COPY Policy Item ................................ 27  
- WHERE USED Policy Item ......................... 28  
- Select ... ....................................... 29  

**Chapter 3. Scenarios on How to Use the Customization Dialog** ................................. 33

- How to Work with Resources .................... 33  
- A Grouping Scenario .............................. 33  
- Grouping the Resources......................... 35  
- How to Use Schedules (Service Periods) .......... 59  
- Schedule Override ................................ 59  
- Early Start ..................................... 60  
- How to Use Triggers and Events ................... 60  
- Scenario Environment and Behavior ............... 60  
- A Possible Solution .............................. 61  
- How to Use User E-T Pairs ....................... 67  
- Step 1: Specifying User Data in the Customization Dialog .... 68  
- Step 2: Access the User Data .................... 68  
- How to Implement Message Processing with User-Defined Data ........... 69  
- Step 1: Specify the User-Defined Data ........... 69  
- Step 2: Specify the Commands .................... 70  
- Step 3: Update the NetView Automation Table .... 71  
- Step 4: Write the REXX Script .................... 73  

**Chapter 4. Entry Type Reference** ................................. 75

- Defining Automation for TWS Components ........ 76  
- Defining Automation for CICS Components .......... 77  
- Enterprise Entry Type ............................ 77  
- Policy Items for an Enterprise ................... 77  
- SEND COMMAND OPERS Policy Item ............. 79  
- INGSEND PARMS Policy Item ..................... 80  
- PROCOPS FOCAL POINTS Policy Item ............. 81  
- AUTO MSG CLASSES Policy Item .................. 82  
- SYMBOL DESCRIPTION Policy Item ............... 83  
- Group Entry Type ................................. 83  
- Creating a New Group ......................... 84  
- Policy Items for Standard Groups ............... 85  
- Policy Items for Syplex Groups .................. 86  
- GROUP INFO Policy Item ......................... 88  
- SYSPLEX Policy Item .............................. 89  
- NMC DEFINITIONS Policy Item .................... 91  
- LOCAL PAGE DATA SET Policy Item ............. 92  
- JOB DEFINITIONS Policy Item .................... 93  
- JOB/ASID DEFINITIONS Policy Item ............... 94  
- COMMAND DEFINITIONS Policy Item ............. 95  
- COMMAND FLOODING Policy Item ............... 97  
- RESOURCE DEFINITIONS Policy Item ............. 97  

© Copyright IBM Corp. 1996, 2012
## Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Entry Type Selection Panel</td>
</tr>
<tr>
<td>2.</td>
<td>Entry Name Selection Panel for Groups</td>
</tr>
<tr>
<td>3.</td>
<td>Policy Selection Panel for Sysplex Groups</td>
</tr>
<tr>
<td>4.</td>
<td>Nested Class Links</td>
</tr>
<tr>
<td>5.</td>
<td>Inheritance from Classes and Defaulting</td>
</tr>
<tr>
<td>6.</td>
<td>Confirmation Panel for Bulk Deletion with WHERE USED Links</td>
</tr>
<tr>
<td>7.</td>
<td>Sample Description Policy Item Panel</td>
</tr>
<tr>
<td>8.</td>
<td>Applications for ApplicationGroup Panel for System Application Groups</td>
</tr>
<tr>
<td>9.</td>
<td>Scenario: Current Configuration with Sysplex Application Group CICSAOR</td>
</tr>
<tr>
<td>10.</td>
<td>Scenario: Current Configuration with Applications CICS1AR1, CPSMCMS, CPSMWUI</td>
</tr>
<tr>
<td>11.</td>
<td>Scenario: Resources for the CICSAOR Sysplex Application Group</td>
</tr>
<tr>
<td>12.</td>
<td>Scenario: Current Configuration with Application Group DB2</td>
</tr>
<tr>
<td>13.</td>
<td>Scenario: Select System Panel for DB2 Application group</td>
</tr>
<tr>
<td>14.</td>
<td>Scenario: Select Resources Panel for DB2 Application Group</td>
</tr>
<tr>
<td>15.</td>
<td>Scenario: Grouping Resources</td>
</tr>
<tr>
<td>16.</td>
<td>Scenario: Resources for the CICS Sysplex Application Group</td>
</tr>
<tr>
<td>17.</td>
<td>Scenario: Current Configuration with Applications Linked to DB2</td>
</tr>
<tr>
<td>18.</td>
<td>Scenario: Select Resources Panel for Application Group NET</td>
</tr>
<tr>
<td>19.</td>
<td>Scenario: Select Resources Panel for DWHAPG with valid APG Resources</td>
</tr>
<tr>
<td>20.</td>
<td>Scenario: Setting Preferences with the Select Resources Panel for DWHAPG</td>
</tr>
<tr>
<td>21.</td>
<td>Scenario: Define New Entry Input Panel for WBSRVAPG with Converted Preferences</td>
</tr>
<tr>
<td>22.</td>
<td>Scenario: Define New Entry Input Panel for WBSRVAPG</td>
</tr>
<tr>
<td>23.</td>
<td>Policy Objects in the Solution</td>
</tr>
<tr>
<td>24.</td>
<td>Generated Resources in the Solution</td>
</tr>
<tr>
<td>25.</td>
<td>UET Entry-Type Selection Panel</td>
</tr>
<tr>
<td>26.</td>
<td>UET Keyword-Data Specification Panel</td>
</tr>
<tr>
<td>27.</td>
<td>Scenario: Using the Message Processing Panel</td>
</tr>
<tr>
<td>28.</td>
<td>Scenario: User Data Processing Panel</td>
</tr>
<tr>
<td>29.</td>
<td>Scenario: Command Processing Panel</td>
</tr>
<tr>
<td>30.</td>
<td>Scenario: Message Automation Overview Panel with Default Values</td>
</tr>
<tr>
<td>31.</td>
<td>Scenario: Editor with User Values for Automation Table Entry</td>
</tr>
<tr>
<td>32.</td>
<td>Scenario: Message Automation Overview with User Values</td>
</tr>
<tr>
<td>33.</td>
<td>Entry Type Selection Panel for Enterprises</td>
</tr>
<tr>
<td>34.</td>
<td>Operator Profile for Sending Commands Panel</td>
</tr>
<tr>
<td>35.</td>
<td>INGSEND Command Parms Panel</td>
</tr>
<tr>
<td>36.</td>
<td>Processor Operations Focal Point Information Panel</td>
</tr>
<tr>
<td>37.</td>
<td>AUTO MSG CLASSES Panel</td>
</tr>
<tr>
<td>38.</td>
<td>System Automation Symbols (AOCCLOX) Description Panel</td>
</tr>
<tr>
<td>39.</td>
<td>Define New Entry Panel for a Sysplex Group</td>
</tr>
<tr>
<td>40.</td>
<td>Policy Selection Panel for Standard Groups</td>
</tr>
<tr>
<td>41.</td>
<td>Policy Selection Panel for Sysplex Groups</td>
</tr>
<tr>
<td>42.</td>
<td>Group Information Panel for Standard Groups</td>
</tr>
<tr>
<td>43.</td>
<td>Sysplex Policy Definition Panel for Sysplex Groups</td>
</tr>
<tr>
<td>44.</td>
<td>NMC Definitions Panel for Sysplex Groups</td>
</tr>
<tr>
<td>45.</td>
<td>Local Page Data Set Recovery Panel</td>
</tr>
<tr>
<td>46.</td>
<td>Local Page Data Set Recovery Job Definition Panel</td>
</tr>
<tr>
<td>47.</td>
<td>&quot;Hung&quot; Command Definitions Panel</td>
</tr>
<tr>
<td>48.</td>
<td>Command Flooding Definition Panel</td>
</tr>
<tr>
<td>49.</td>
<td>Long Running ENQ Job/ASID Definitions Panel</td>
</tr>
<tr>
<td>50.</td>
<td>&quot;Hung&quot; Command Definitions Panel</td>
</tr>
<tr>
<td>51.</td>
<td>Command Flooding Definition Panel</td>
</tr>
<tr>
<td>52.</td>
<td>Long Running ENQ Resource Definition Panel</td>
</tr>
<tr>
<td>53.</td>
<td>Recovery Options Panel</td>
</tr>
<tr>
<td>54.</td>
<td>Define New Entry Panel for a Subgroup</td>
</tr>
<tr>
<td>55.</td>
<td>Policy Selection Panel for Subgroups</td>
</tr>
<tr>
<td>56.</td>
<td>Define New Entry Panel for a System</td>
</tr>
<tr>
<td>57.</td>
<td>Policy Selection Panel for Systems</td>
</tr>
<tr>
<td>58.</td>
<td>System Information Panel</td>
</tr>
<tr>
<td>59.</td>
<td>Automation Symbols Panel</td>
</tr>
<tr>
<td>60.</td>
<td>Environment Definition: CONSOLE (Route Codes) Panel</td>
</tr>
<tr>
<td>Page</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>87.</td>
<td>SPOOLFULL Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>88.</td>
<td>SPOOLSHORT Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>89.</td>
<td>SPOOLFULL Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>90.</td>
<td>SPOOLSHORT Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>91.</td>
<td>SPOOLFULL Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>92.</td>
<td>SPOOLSHORT Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>93.</td>
<td>SPOOLFULL Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>94.</td>
<td>SPOOLSHORT Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>95.</td>
<td>SPOOLFULL Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>96.</td>
<td>SPOOLSHORT Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>97.</td>
<td>SPOOLFULL Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>98.</td>
<td>SPOOLSHORT Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>99.</td>
<td>SPOOLFULL Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>100.</td>
<td>SPOOLSHORT Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>101.</td>
<td>SPOOLFULL Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>102.</td>
<td>SPOOLSHORT Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>103.</td>
<td>SPOOLFULL Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>104.</td>
<td>SPOOLSHORT Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>105.</td>
<td>SPOOLFULL Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>106.</td>
<td>SPOOLSHORT Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>107.</td>
<td>SPOOLFULL Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>108.</td>
<td>SPOOLSHORT Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>109.</td>
<td>SPOOLFULL Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>110.</td>
<td>SPOOLSHORT Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>111.</td>
<td>SPOOLFULL Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>112.</td>
<td>SPOOLSHORT Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>113.</td>
<td>SPOOLFULL Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>114.</td>
<td>SPOOLSHORT Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>115.</td>
<td>SPOOLFULL Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>116.</td>
<td>SPOOLSHORT Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>117.</td>
<td>SPOOLFULL Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>118.</td>
<td>SPOOLSHORT Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>119.</td>
<td>SPOOLFULL Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>120.</td>
<td>SPOOLSHORT Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>121.</td>
<td>SPOOLFULL Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>122.</td>
<td>SPOOLSHORT Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>123.</td>
<td>SPOOLFULL Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>124.</td>
<td>SPOOLSHORT Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>125.</td>
<td>SPOOLFULL Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>126.</td>
<td>SPOOLSHORT Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>127.</td>
<td>SPOOLFULL Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>128.</td>
<td>SPOOLSHORT Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>129.</td>
<td>SPOOLFULL Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>130.</td>
<td>SPOOLSHORT Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>131.</td>
<td>SPOOLFULL Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>132.</td>
<td>SPOOLSHORT Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>133.</td>
<td>SPOOLFULL Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>134.</td>
<td>SPOOLSHORT Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>135.</td>
<td>SPOOLFULL Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>136.</td>
<td>SPOOLSHORT Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>137.</td>
<td>SPOOLFULL Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>138.</td>
<td>SPOOLSHORT Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>139.</td>
<td>SPOOLFULL Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>140.</td>
<td>SPOOLSHORT Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>141.</td>
<td>SPOOLFULL Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>142.</td>
<td>SPOOLSHORT Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>143.</td>
<td>SPOOLFULL Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>144.</td>
<td>SPOOLSHORT Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>145.</td>
<td>SPOOLFULL Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>146.</td>
<td>SPOOLSHORT Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>147.</td>
<td>SPOOLFULL Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>148.</td>
<td>SPOOLSHORT Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>149.</td>
<td>SPOOLFULL Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>150.</td>
<td>SPOOLSHORT Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>151.</td>
<td>SPOOLFULL Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>152.</td>
<td>SPOOLSHORT Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>153.</td>
<td>SPOOLFULL Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>154.</td>
<td>SPOOLSHORT Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>155.</td>
<td>SPOOLFULL Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>156.</td>
<td>SPOOLSHORT Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>157.</td>
<td>SPOOLFULL Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>158.</td>
<td>SPOOLSHORT Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>159.</td>
<td>SPOOLFULL Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>160.</td>
<td>SPOOLSHORT Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>161.</td>
<td>SPOOLFULL Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>162.</td>
<td>SPOOLSHORT Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>163.</td>
<td>SPOOLFULL Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>164.</td>
<td>SPOOLSHORT Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>165.</td>
<td>SPOOLFULL Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>166.</td>
<td>SPOOLSHORT Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>167.</td>
<td>SPOOLFULL Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>168.</td>
<td>SPOOLSHORT Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>169.</td>
<td>SPOOLFULL Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>170.</td>
<td>SPOOLSHORT Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>171.</td>
<td>SPOOLFULL Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>172.</td>
<td>SPOOLSHORT Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>173.</td>
<td>SPOOLFULL Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>174.</td>
<td>SPOOLSHORT Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>175.</td>
<td>SPOOLFULL Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>176.</td>
<td>SPOOLSHORT Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>177.</td>
<td>SPOOLFULL Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>178.</td>
<td>SPOOLSHORT Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>179.</td>
<td>SPOOLFULL Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>180.</td>
<td>SPOOLSHORT Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>181.</td>
<td>SPOOLFULL Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>182.</td>
<td>SPOOLSHORT Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>183.</td>
<td>SPOOLFULL Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>184.</td>
<td>SPOOLSHORT Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>185.</td>
<td>SPOOLFULL Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>186.</td>
<td>SPOOLSHORT Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>187.</td>
<td>SPOOLFULL Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>188.</td>
<td>SPOOLSHORT Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>189.</td>
<td>SPOOLFULL Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>190.</td>
<td>SPOOLSHORT Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>191.</td>
<td>SPOOLFULL Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>192.</td>
<td>SPOOLSHORT Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>193.</td>
<td>SPOOLFULL Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>194.</td>
<td>SPOOLSHORT Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>195.</td>
<td>SPOOLFULL Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>196.</td>
<td>SPOOLSHORT Processing Panel for JES2 Subsystems</td>
</tr>
<tr>
<td>197.</td>
<td>SPOOLFULL Processing Panel for JES3 Subsystems</td>
</tr>
<tr>
<td>198.</td>
<td>SPOOLSHORT Processing Panel for JES3 Subsystems</td>
</tr>
</tbody>
</table>
Tables

1. System Automation for z/OS Library xvii
2. System Symbols and System Automation Symbols 10
3. System Symbols 11
4. System Automation Symbols 11
5. Entry Details for Applications in the Scenario 39
6. Preference Value Scenarios 136
7. SA z/OS Message Classes 278
8. Policy Selection Filter Attributes 366
Accessibility

Publications for this product are offered in Adobe Portable Document Format (PDF) and should be compliant with accessibility standards. If you experience difficulties when using PDF files, you may view the information through the z/OS Internet Library website or the z/OS Information Center. If you continue to experience problems, send an email to mhvrdfs@us.ibm.com or write to:

IBM® Corporation
Attention: MHVRCFS Reader Comments
Department H6MA, Building 707
2455 South Road
Poughkeepsie, NY 12601-5400
U.S.A.

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

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

Using assistive technologies

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

Keyboard navigation of the user interface

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

z/OS information

z/OS information is accessible using screen readers with the BookServer or Library Server versions of z/OS books in the Internet library at:

http://www.ibm.com/systems/z/os/zos/bkserv/
How to send your comments to IBM

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

Use one of the following methods to send us your comments:
1. Send an email to s390id@de.ibm.com
4. Mail the comments to the following address:
   IBM Deutschland Research & Development GmbH
   Department 3248
   Schoenaicher Str. 220
   D-71032 Boeblingen
   Federal Republic of Germany
5. Fax the comments to us as follows:
   From Germany: 07031-16-3456
   From all other countries: +(49)-7031-16-3456

Include the following information:
• Your name and address
• Your email address
• Your telephone or fax number
• The publication title and order number:
  IBM Tivoli System Automation for z/OS V3R4.0 Defining Automation Policy
  SC34-2646-00
• The topic and page number related to your comment
• The text of your comment.

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

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

If you have a technical problem

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

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

**Who Should Use This Book**

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

**Notes on Terminology**

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

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

**Resources, Applications, and Subsystems**

These three terms are used in the SA z/OS library to designate automated entities. The correct term to be used depends on the context. See the chapter “Introduction” in IBM Tivoli System Automation for z/OS Operator's Commands for the definition of these terms.

**Where to Find More Information**

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

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

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

<table>
<thead>
<tr>
<th>Title</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM Tivoli System Automation for z/OS Planning and Installation</td>
<td>SC34-2645</td>
</tr>
<tr>
<td>IBM Tivoli System Automation for z/OS Customizing and Programming</td>
<td>SC34-2644</td>
</tr>
<tr>
<td>IBM Tivoli System Automation for z/OS Defining Automation Policy</td>
<td>SC34-2646</td>
</tr>
<tr>
<td>IBM Tivoli System Automation for z/OS User's Guide</td>
<td>SC34-2647</td>
</tr>
<tr>
<td>IBM Tivoli System Automation for z/OS Messages and Codes</td>
<td>SC34-2648</td>
</tr>
<tr>
<td>IBM Tivoli System Automation for z/OS Operator's Commands</td>
<td>SC34-2649</td>
</tr>
<tr>
<td>IBM Tivoli System Automation for z/OS Programmer's Reference</td>
<td>SC34-2650</td>
</tr>
</tbody>
</table>
Table 1. System Automation for z/OS Library (continued)

<table>
<thead>
<tr>
<th>Title</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM Tivoli System Automation for z/OS Product Automation Programmer’s</td>
<td>SC34-2643</td>
</tr>
<tr>
<td>Reference and Operator’s Guide</td>
<td></td>
</tr>
<tr>
<td>IBM Tivoli System Automation for z/OS TWS Automation Programmer’s</td>
<td>SC34-2651</td>
</tr>
<tr>
<td>Reference and Operator’s Guide</td>
<td></td>
</tr>
<tr>
<td>IBM Tivoli System Automation for z/OS End-to-End Automation Adapter</td>
<td>SC34-2652</td>
</tr>
<tr>
<td>IBM Tivoli System Automation for z/OS Monitoring Agent Configuration and User’s Guide</td>
<td>SC34-2653</td>
</tr>
</tbody>
</table>

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

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

---

**Related Product Information**

You can find books in related product libraries that may be useful for support of the SA z/OS base program by visiting the z/OS Internet Library at [http://www.ibm.com/systems/z/os/zos/bkserv](http://www.ibm.com/systems/z/os/zos/bkserv)

**Using LookAt to look up message explanations**

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

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

- The Internet. You can access IBM message explanations directly from the LookAt Website at [www.ibm.com/systems/z/os/zos/bkserv/lookat/index.html](http://www.ibm.com/systems/z/os/zos/bkserv/lookat/index.html)
- Your z/OS TSO/E host system. You can install code on your z/OS or z/OS.e systems to access IBM message explanations using LookAt from a TSO/E command line (for example: TSO/E prompt, ISPF, or z/OS UNIX System Services).
- Your Microsoft Windows workstation. You can install LookAt directly from the [z/OS Collection](SK3T-4269) or the [z/OS and Software Products DVD Collection](SK3T-4271) and use it from the resulting Windows graphical user interface (GUI). The command prompt (also known as the DOS > command line) version can still be used from the directory in which you install the Windows version of LookAt.
- Your wireless handheld device. You can use the LookAt Mobile Edition from [www.ibm.com/systems/z/os/zos/bkserv/lookat/lookatm.html](http://www.ibm.com/systems/z/os/zos/bkserv/lookat/lookatm.html) with a handheld device that has wireless access and an Internet browser (for example: Internet Explorer for Pocket PCs, Blazer or Eudora for Palm OS, or Opera for Linux handheld devices).
You can obtain code to install LookAt on your host system or Microsoft Windows workstation from:

- A CD-ROM in the z/OS Collection (SK3T-4269).
- The z/OS and Software Products DVD Collection (SK3T-4271).
- The LookAt Website (click Download and then select the platform, release, collection, and location that suit your needs). More information is available in the LOOKAT.ME files available during the download process.

**Summary of Changes for SC34-2646-00**

This document contains information previously presented in System Automation for z/OS V3R3.0 Defining Automation Policy, SC34-2572-02.

The following sections provide details of significant or major changes in this edition. All changes are indicated by a vertical line to the left of the change.

**New Information**

The following information is new in this edition:

**Status Determination field for the Application Group Policy Item**

A Status Determination field is added to the Application Group Policy Item to specify how the compound status of an application group is determined. See “APPLGROUP INFO Policy Item” on page 122.

**Templates for Text File Update of the Customization Dialog**

An option to create templates for text file update of policy objects in the Customization Dialog is introduced in “Creating New Policy Objects Using Text Files” on page 351.

**Command and User Data Policies**


**zEnterprise Ensembles entry type**

The zEnterprise Ensembles entry type (ENS) that enables you to use zEnterprise z/BX Compute power is added. See “zEnterprise Ensembles Entry Type” on page 241.

**Runtokens**

The Runtoken field for identifying a runmode is added in “APPLICATION INFO Policy Item” on page 152 and “MONITOR INFO Policy Item” on page 231.

**INGROMON**

The INGROMON monitoring routine is added for the Monitor Routine entry in “APPLICATION INFO Policy Item” on page 152.

**Job Log Monitor Interval**

The Job Log Monitor Interval field is added to the “APPLICATION INFO Policy Item” on page 152.

**Connection Monitor Interval**

The Connection Monitor Interval field is added in “PROCESSOR INFO Policy Item” on page 222 for Internal Processor data.

**IP Stack**

IP Stack field information has been added to the screen illustration and descriptions in “USS CONTROL Policy Item” on page 210.
System and Application Specific Overrides

Keywords *BEGINAPL, *ENDAPL, *BEGINMVC, *ENDMVC, *BEGINSYS and *ENDSYS have been introduced to mark an AT override section for selected application instances, classes or MVS component entries, see “System and Application Specific Overrides” on page 329.

Enabling/Disabling NetView Automation Table or Message Revision Table syntax checking

A syntax check is now available for overrides specified in the NetView Automation Table or Message Revision Table entries. See “Enabling and Disabling AT / MRT Override Syntax Checking” on page 362.

Changed Information

The following information has been changed in this edition:

Application Types and Subtypes

New application types (TCPIP, INFOSPHERE, and LIFELINE) are added for defining a new application and where otherwise appropriate. See “Application Entry Type” on page 140.

&*JOBNAME. variable

A new variable &*JOBNAME. replaces the &SUBSJOB variable when “Converting from SA z/OS 3.2” on page 355.

Automation Flags Policy Item

Panel layout has changed and now shows inherited data. See “AUTOMATION FLAGS Policy Item” on page 160.

Minor Resources Policy Item

Panel layout has changed and now shows inherited data. See “MINOR RESOURCES Policy Item” on page 193.

Moved Information

- Instructions for handling line commands within a Command Processing Panel are moved from the Command Processing section in MESSAGES/USER DATA Policy Item to “Command and User Data Policies” on page 25 in Chapter 2, “Using the Customization Dialog,” on page 15.
- The Operators Policy Item and the Structure of AOFWRKxx Operators sections are moved to the IBM Tivoli System Automation for z/OS User’s Guide.
- The MESSAGE AUTOMATION and MPF DEFINITIONS policy items are removed from the Enterprise Entry Type section and placed in the Messages Entry Type section. See “Messages Entry Type” on page 253.

Deleted Information

The following information has been deleted:

- The following definitions have been retired:
  - entry type ISA (IMS State Action Tables)
  - entry type TMO (Timeout Settings)
- The following policy items have been retired (because of the retired entry types):
  - policy STATE ACTION TABLE within entry type APL (Application)
  - policy AUTOMATION TIMEOUT within entry type SYS (Systems)

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

This document contains terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.
Chapter 1. Introducing the Customization Dialog

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

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

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

Before you begin to use the customization dialog to define your enterprise automation policy, you need to determine how you want SA z/OS to work within your enterprise. This is part of the process of planning to use SA z/OS. There are many considerations for planning to use SA z/OS that you should be aware of. For more information, see the planning instructions given in IBM Tivoli System Automation for z/OS Planning and Installation.

Note:
Do not use the customization dialogs on a Kanji terminal because 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.

How to Define Automation Policy: An Outline

Automation policy is needed if you want to use system operations or processor operations automation functions. The process of defining automation policy comprises the following steps:

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

2. Use the BUILD function of the customization dialog to produce the required output, that is, the configuration files: automation control file (ACF), automation manager configuration file (AMC), NetView automation table (AT), the message revision table (MRT), and the MPFLSTSA file.

3. Distribute the generated output to each system in the SA z/OS enterprise where SA z/OS system operations functions are to run. In general, this is all of
Entry Type Introduction

All resources in your enterprise, such as systems or applications, or other objects that you use to automate your enterprise data processing, such as components of your z/OS operating system, are called entry types. A complete list of SA z/OS entry types is given on the Entry Type Selection panel (see Figure 1 on page 5), which is the first panel you see after you have decided which enterprise or policy database you are going to work with.

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

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

The letters and numbers 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 on 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 is given in Chapter 3, “Scenarios on How to Use the Customization Dialog,” on page 33.

1/ENT Enterprise

An Enterprise is the top level of your SA z/OS structure. It comprises enterprise wide data for the complete policy database.

2/GRP Group

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

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

3/SBG Subgroup

Like groups, subgroups are collections of systems. You define a subgroup by selecting the member systems and assigning a name to the subgroup. Subgroups can be collected into groups. They offer a finer level of structuring groups.
4/SYS System
A system is an image of an operating system running on a host processor monitored by SA z/OS. The system may be MVS, VM, VSE, LINUX, or CF (coupling facility). A system definition includes for example:
- Links to other objects associated with the system, for example, links to application groups
- Application defaults
- System defaults
- System automation symbols

5/APG ApplicationGroup
An application group is a set of applications that you assign a name to. 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 or more sysplexes.

6/APL Application
An application is a z/OS subsystem or job that runs on a system in the SA z/OS enterprise. SA z/OS monitors applications through the messages they issue to the system, MVS commands, and SA z/OS-provided services. An application can belong to more than one application group. An application definition includes:
- Subsystem name and the name of the procedure that runs the job
- Warning and alert thresholds
- Start and Stop specifications
- Individual automation flags

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

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

9/TRG Triggers
In combination with events, triggers are used to control the starting and stopping of resources (APL, APG, or MTR).

10/PRO Processor
A processor is the physical or physically-partitioned hardware that a system runs on. A "logical" processor, called the ProcOps Service Machine (PSM) monitors VM guest systems.

All processors of the IBM mainframe family are supported, which includes all IBM System z9®, IBM System z10®, IBM eServer® zSeries®, IBM S/390®, and IBM z196 Enterprise.

11/MTR Monitor Resource
The Monitor Resource entry type enables you to monitor the performance and health of an application. That is, a separate status informs you about the application's health. This health status can be used by the automation manager to make decisions and, if necessary, trigger automation for the subject application.
**Entry Type Introduction**

12/zEnterprise Ensembles
The zEnterprise Ensembles entry type (ENS) enables you to use zEnterprise z/BX Compute power.

20/PRD Product Automation
With this entry type, you can define automation policy for TWS and CICS®.

21/MSG Messages
This entry type displays a list of all the messages that are defined in the policy database together with the messages from the MVC entry, +SA_PREDEFINED_MSGS.

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.

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 you want to monitor tape mounts on this system. If you do want to monitor tape mounts, you must enter warning and alert thresholds (in minutes and seconds) indicating how long a wait is allowed in mounting a tape before the response is considered unsatisfactory.

33/MVC MVS Component
This entry type allows you to apply automation to various z/OS data sets, resources and facilities.

34/MDF MVSCOMP Defaults
Sets defaults for z/OS component (entry type MVS Component) automation.

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

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

37/AOP Automation Operators
This entry type 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 to act on.

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

39/NTW Network
Contains definitions for Gateway, SDF focal points, OMEGAMON sessions, TAF sessions, authentication, and SOAP servers.
Entry Type Introduction

40/XDF Sysplex Defaults
This entry type specifies default values for sysplex application groups.

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

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

99/UET User E-T Pairs
User-defined entry/type data for the automation control file. Used for user-defined extensions to automation policy.

Policy Objects and Policy Items
A policy object is represented by an entry name for a particular entry type, for example, an application or timer.

Each policy object has a number of policy items associated with it. To define your policy, you enter data for the policy items of each policy object in the SA z/OS enterprise. For example, if you choose option 2 on the Entry Type Selection panel [Figure 1] to specify Group policy, Figure 2 on page 6 is displayed.

<table>
<thead>
<tr>
<th>MENU</th>
<th>OPTIONS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFGETYP</td>
<td>Entry Type Selection</td>
<td></td>
</tr>
<tr>
<td>Option =&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 ENT</td>
<td>Enterprise</td>
<td>30 TMR</td>
</tr>
<tr>
<td>2 GRP</td>
<td>Groups</td>
<td>32 TPA</td>
</tr>
<tr>
<td>3 SBG</td>
<td>SubGroups</td>
<td>33 MVC</td>
</tr>
<tr>
<td>4 SYS</td>
<td>Systems</td>
<td>34 MDF</td>
</tr>
<tr>
<td>5 APG</td>
<td>ApplicationGroups</td>
<td>34 SDF</td>
</tr>
<tr>
<td>6 APL</td>
<td>Applications</td>
<td>36 ADF</td>
</tr>
<tr>
<td>7 EVT</td>
<td>Events</td>
<td>37 AOP</td>
</tr>
<tr>
<td>8 SVP</td>
<td>Service Periods</td>
<td>38 NFY</td>
</tr>
<tr>
<td>9 TRG</td>
<td>Triggers</td>
<td>39 NTW</td>
</tr>
<tr>
<td>10 PRO</td>
<td>Processors</td>
<td>40 XDF</td>
</tr>
<tr>
<td>11 MTR</td>
<td>Monitor Resources</td>
<td>41 RES</td>
</tr>
<tr>
<td>12 ENS</td>
<td>zEnterprise Ensembles</td>
<td>42 SCR</td>
</tr>
<tr>
<td>20 PRD</td>
<td>Product Automation</td>
<td>99 UET</td>
</tr>
<tr>
<td>21 MSG</td>
<td>Messages</td>
<td>(*)</td>
</tr>
</tbody>
</table>

(+) Multi-User-Capable

Figure 1. Entry Type Selection Panel
A label for each panel (for example, AOFGETYP, AOFGENAM) is shown to the left of the panel title. This is the panel ID. You can toggle its display by entering the command PANELID at the command line.

Figure 2 on page 6 shows the Group policy objects SYSPLEX1 and SYSPLEX2.
Entry Type Introduction

If you select a group, the Policy Selection panel for this group is displayed, as shown in Figure 3. This panel gives you a list of the policy items for a group. You can enter S (for SELECT) in the Action field next to any policy item to display the associated policy input panel. Figure 3 shows the Policy Selection panel for groups, listing the policy items for the selected policy object SYSPLEX1.

Considerations for the SA z/OS Graphical Interface

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

The objects visible on the NMC workstation are based on the resources that are defined for NMC using the INFORMLIST specification in the customization dialog and therefore known to SA z/OS. For more details, see for example “APPLICATION INFO Policy Item” on page 152.
Considerations for the SA z/OS Graphical Interface

The automation manager component of SA z/OS monitors the resources and their status. SA z/OS permanently stores information about the resources and their status into the focal point RODM from where it is retrieved by the NMC workstation.

Grouping Resources Logically

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

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

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

Considerations for Automation

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

SA z/OS provides automation procedures that are internally invoked according to the NetView automation table and that manage, for example, the following:
- TCP/IP
- HTTP Server
- Job Entry Subsystem/2 (JES2)
- Job Entry Subsystem/3 (JES3)
- Time Sharing Option/Extensions (TSO/E)
- APPC
- ASCH
- Resource Measurement Facility™ (RMF™)
- Virtual Telecommunication Access Method (VTAM®)
- NetView
- Transient jobs
- USS processes
Considerations for Automation

- Non-z/OS resources
- Any other applications

SA z/OS manages and integrates with:
- CICS
- IMS
- DB2
- TWS
- SAP
- Websphere
- TBSM
- GDPS and others.

SA z/OS provides automation procedures that enable recovery of the following z/OS components and data sets:
- SYSLOG data sets
- LOGREC data sets
- System Management Facility (SMF) data sets
- Write-to-operator (WTO) buffers
- JES spools
- z/OS dump data sets
- Action Message Retention Facility (AMRF) buffers
- Sysplex resources, for example, coupling facility

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

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

The SA z/OS policy administrator uses the customization dialog to set the way that enterprise resources are to be handled by SA z/OS when predefined conditions are met. In addition to invoking provided automation procedures, SA z/OS can be used to trigger user-written automation procedures.

---

Note:
In order to avoid problems with automation procedures and customization dialogs, do not use the following terms for entry names or application subsystem names: RC, RETURN, or SIGL.

---

Assigning System Automation Symbols (AOCCLONE)

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

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

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

Using System Symbols and System Automation Symbols
You can specify system symbols as well as system automation symbols (AOCCCLONE variables) in the fields in the customization dialog that are given in Table 2 on page 10.
### Assigning System Automation Symbols (AOCCLONE)

**Table 2. System Symbols and System Automation Symbols**

<table>
<thead>
<tr>
<th>Entry Type</th>
<th>Policy Item</th>
<th>Field Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>APL</td>
<td>APPLICATION INFO</td>
<td>ARM element name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Command prefix</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JCL procedure name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Job name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Startup parameter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WLM resource name</td>
</tr>
<tr>
<td></td>
<td>APPLICATION SYMBOLS</td>
<td>SYMBOL1–SYMBOL9</td>
</tr>
<tr>
<td></td>
<td>DB2® CONTROL</td>
<td>DB2 Active Log Dataset Name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DB2 Subsystem ID</td>
</tr>
<tr>
<td></td>
<td>IMS® CONTROL</td>
<td>IMS Control Region Name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IMS Subsystem ID</td>
</tr>
<tr>
<td></td>
<td>JES2/JES3 SPOOLFULL</td>
<td>Command text</td>
</tr>
<tr>
<td></td>
<td>JES2/JES3 SPOOLSHORT</td>
<td>Command text</td>
</tr>
<tr>
<td></td>
<td>JES2 DRAIN</td>
<td>Force drain lines</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Force drain VTAM interface</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Force drain spool offloaders</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Force drain printers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Force drain readers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Force drain punches</td>
</tr>
<tr>
<td></td>
<td>MESSAGES/USER DATA</td>
<td>Message ID</td>
</tr>
<tr>
<td></td>
<td>MESSAGES/USER DATA (type CMD)</td>
<td>Command text</td>
</tr>
<tr>
<td></td>
<td>MESSAGES/USER DATA (type CODE)</td>
<td>Code1/2/3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Value Returned</td>
</tr>
<tr>
<td></td>
<td>MESSAGES/USER DATA (type REPLY)</td>
<td>Reply text</td>
</tr>
<tr>
<td></td>
<td>MESSAGES/USER DATA (type USER)</td>
<td>Data</td>
</tr>
<tr>
<td></td>
<td>MINOR RESOURCES</td>
<td>Minor resource names</td>
</tr>
<tr>
<td></td>
<td>OPC CONTROL</td>
<td>OPC Subsystem ID</td>
</tr>
<tr>
<td></td>
<td>STARTUP</td>
<td>Command text</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Startup parameter</td>
</tr>
<tr>
<td></td>
<td>SHUTDOWN</td>
<td>Command text</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reply text</td>
</tr>
<tr>
<td></td>
<td>USS CONTROL</td>
<td>Command/Path</td>
</tr>
<tr>
<td></td>
<td></td>
<td>File Name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Filter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Monitoring Command</td>
</tr>
<tr>
<td></td>
<td></td>
<td>User ID</td>
</tr>
<tr>
<td></td>
<td>ENS</td>
<td>TCPIP stack</td>
</tr>
<tr>
<td></td>
<td>GRP</td>
<td>Temporary Data Set HLQ</td>
</tr>
<tr>
<td></td>
<td>MVC</td>
<td>Command text</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Message ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reply text</td>
</tr>
<tr>
<td></td>
<td>MESSAGES/USER DATA (type CODE)</td>
<td>Code1/2/3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Value Returned</td>
</tr>
<tr>
<td></td>
<td>MESSAGES/USER DATA (type USER)</td>
<td>Data</td>
</tr>
<tr>
<td></td>
<td>MINOR RESOURCES</td>
<td>Minor resource names</td>
</tr>
</tbody>
</table>
Assigning System Automation Symbols (AOCCLONE)

Table 2. System Symbols and System Automation Symbols (continued)

<table>
<thead>
<tr>
<th>Entry Type</th>
<th>Policy Item</th>
<th>Field Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTR</td>
<td>MONITOR INFO</td>
<td>Activate command(^1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Deactivate command(^1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Monitor command(^1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Monitored Object(^1)</td>
</tr>
<tr>
<td>NTW</td>
<td>SOAP SERVER</td>
<td>Host Name(^1)</td>
</tr>
<tr>
<td>SYS</td>
<td>AUTOMATION CONSOLE</td>
<td>Console Name</td>
</tr>
<tr>
<td></td>
<td>SYSTEM INFO</td>
<td>UNIX Installation Path(^1)</td>
</tr>
<tr>
<td>TMR</td>
<td>TIMERS</td>
<td>Command text(^1)</td>
</tr>
<tr>
<td>UET</td>
<td>E-T DATA</td>
<td>Data(^1)</td>
</tr>
</tbody>
</table>

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

Table 3 lists the fields in the customization dialog where you can only specify system symbols.

Table 3. System Symbols

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

Table 4 lists the field in the customization dialog where you can only specify system automation symbols (AOCCLONE variables).

Table 4. System Automation Symbols

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Policy Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supporting Resource Name</td>
<td>Policy RELATIONSHIPS of entry type APL and APG</td>
</tr>
</tbody>
</table>

Nested Classes

Nested class support allows you to link one application class to another, thus enabling the specification of a class hierarchy. Links from a class can be made in two directions:
- **Downward** to multiple classes or instances, where the policy that is defined in the upper-level class is inherited by the classes or instances
- **Upward** to a single class, so that the lower-level application inherits the policy that is defined in the upper-level class

These classes can carry data on various levels, be nested, and an application can inherit data from a class chain. This can be particularly useful for policy databases with lots of applications of the same type, for example, large IMS installations with lots of IMS subsystems. You might define a top-level class with basic definitions.
Nested Classes

for all IMS applications, and nested classes with more specific definitions for the
various IMS subtypes, so that applications can inherit policy values from the
top-level class and specific nested classes.

Class policies are inherited down to the instance level where any changes to the
policy on that level override the inherited policy. Inherited policy data is displayed
in a different color that you can change in the customization dialog Settings Menu
(see “Changing Your Customization Dialog Environment” on page 361). Inherited
data is displayed for linked instances for the following policies:

- “APPLICATION INFO Policy Item” on page 152
- “AUTOMATION FLAGS Policy Item” on page 160
- “RELATIONSHIPS Policy Item” on page 164
- “MESSAGES/USER DATA Policy Item” on page 165
- “STARTUP Policy Item” on page 181
- “SHUTDOWN Policy Item” on page 185
- “THRESHOLDS Policy Item” on page 192
- “MINOR RESOURCES Policy Item” on page 193
- “DB2 CONTROL Policy Item” on page 205
- “IMS CONTROL Policy Item” on page 207
- “OPC CONTROL Policy Item” on page 209
- “USS CONTROL Policy Item” on page 210

Note that you can overwrite inherited data for an instance with the same value for
the following policy items:
- APPLICATION INFO
- THRESHOLDS
- DB2 CONTROL
- IMS CONTROL
- OPC CONTROL
- USS CONTROL

Example:
If an application instance is linked to a class with a job type of TRANSIENT
and you change the Transient Rerun value, you also need to specify the Job
Type TRANSIENT on the instance even though it has already been inherited
from the class.

Class Hierarchy

Classes can be linked to each other and to instances according to the following
rules:
- An application class can have one or more downward links to other classes or
  instances
- There can be only one upward link from an application class to any particular
class
- There is no limit for the number of supported class levels

Figure 4 on page 13 shows example links that are allowed or forbidden between
classes and instances.
Inheritance and Defaulting

Figure 5 shows how inheritance from application classes relates to defaulting from the defaults policy items.

The data that is defined in Defaults policies and for application classes is handled differently. If class data is modified and built, a configuration refresh updates the applications that are linked to the modified classes. However, if defaults are modified and built, a configuration refresh updates only the global variables for the modified entries but not for the affected objects, for example, applications.
Class/Instance Link Considerations
You should be aware of the following when making links between classes and instances:

- Only applications with either the same type or no type specification can be in a link chain.
- Type-specific policies are only available if the type is specified explicitly rather than inherited.

For example, the IMS CONTROL policy item is available for an application of type IMS. It is not available, however, for an application that does not have a type specification but is linked to a class of type IMS.

Thus, if an application without a type is linked to a class with a type specification, you should be aware that you can only specify type-specific data for the class, but it will be inherited by the instance.

Furthermore, if you need to specify type-specific data at the application level, you must specify the type for the application. For example, some of the fields for applications of type USS allow mixed case. An application without a type specification does not allow mixed-case data, but it could inherit such data from a class of type USS.
Chapter 2. Using the Customization Dialog

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

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

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

Your first step in defining automation policy will be to create a new policy database, using one of the sample databases as a model. SA z/OS provides sample policy databases containing sample enterprise definitions. These are described in "Using SA z/OS Sample Policies" on page 303.

To do this, you should create an empty policy database using the *EMPTY sample database, and then use the IMPORT function to import policy definitions from either the other sample databases that are supplied with SA z/OS, or from your own policy databases. For more details about importing policy databases, see "Importing Policy Database Data" on page 337. You should note that this is now the preferred method for creating new policy databases.

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

You select entry types from the list shown on the Entry Type Selection panel [Figure 1 on page 5] and certain objects from these entry types to specify the way enterprise resources are to be handled by SA z/OS when predefined conditions are met. You can link automation policies to individual systems as required to meet your automation needs.

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

To start the customization dialog:

1. Log on to TSO/E using a logon procedure that makes the customization dialog available to you. See IBM Tivoli System Automation for z/OS Planning and Installation for instructions on how to create an appropriate logon procedure.

2. Select the customization dialog from the ISPF menu that it has been installed on. SA z/OS displays the Customization Dialog Primary Menu as shown in Figure 6.

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

- Type the name of your desired policy database into the Current Policy Data Base field (if not filled in from a previous invocation) and enter option 1 (you can also enter Open) at the Option line. This leads you directly to the Entry Type Selection panel shown in Figure 1 on page 5.

- Type ? in the Current Policy Data Base field and press Enter to get a list of available policy databases in the Policy Database Selection panel as shown in Figure 234 on page 299. Select the desired policy database from this panel. (You can also reach the Policy Database Selection panel by entering option 4, or Policies, at the Option line.)

If no policy databases are displayed on the Policy Database Selection panel when you start the customization dialog, you must either create a policy database (see “Creating a New Policy Database” on page 299) or add an existing policy database to the list.

If other users want to view the policy database that you have just created, they must add it to their policy database list on the Policy Database Selection panel using the ADD command.

You can also select the following options:

0 Settings

This option displays the Customization Dialog Settings Menu, see “Changing Your Customization Dialog Environment” on page 361.
How to Start the Customization Dialog

2 Build
This option displays the Build Functions Menu, see Chapter 7, “Building and Distributing Configuration Files,” on page 317.

3 Report
This option displays the Report Selection Menu, see Chapter 6, “Creating Reports about Policy Databases,” on page 309.

4 Policies
This option displays the Policy Database Selection panel with the list of existing policy databases, see “Creating a New Policy Database” on page 299.

5 Data Management
This option displays the Data Management Menu panel, see “Start Panel for Data Management” on page 337.

U User
This option invokes the User-written Functions Selection Menu. This menu allows you to add installation-specific functions to the customization dialog. For more information see “User-written Functions” on page 369.

X Exit
Use this option to exit the customization dialog

Getting Help

Online help is available for the customization dialog. You can display online help by entering the HELP command (or press the PF key associated with the HELP command) while on a particular dialog panel.

When selecting HELP from the action bar, the pull-down lists the following selections:

Automation Help
This help provides online information about important functions of the customization dialog. It combines the help panels of the functional panels within the customization dialog. This combination allows you to view the help panels sequentially, or you can choose selected topics from lists that are displayed on many of the help pages.

Changes for this Release
This panel provides a list of the changes and enhancements that have been added to the customization dialog for the different releases of version 2.

How to Navigate in the Customization Dialog

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

- For panels that display a list that you are required to select one or more entries from 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
How to Navigate in the Customization Dialog

Enter the number corresponding to the action you want to take. For example, place an S in the Action column next to “entry name” and select 1 (OPEN) from the Action menu.

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

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

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

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

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

- On data entry panels, the END (PF3) key verifies input, saves the data, and exits the current panel.
- Pressing the Enter key verifies input and redisplay the current panel.
- In certain cases a long message is displayed in a pop-up window on the panel. If this pop-up window covers required input areas you can move it. For more information see the section “Movable Pop-Ups” in z/OS ISPF Dialog Developer’s Guide and Reference.

How to Use an Entry Type as a Fast Path

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

- 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, enter:
  
  APL

- If you want to start with defining a new application group, for example, enter:
  
  APG; NEW application group_name

- If you want to reach a certain entry type and you already know the name of the object that you want to edit, you can enter a fast path from every panel within the customization dialog. For example, to reach the APPC application, enter:
  
  =APL; S APPC

To reach the application group CICS_APG, enter:

=APG; S CICS_APG

To reach the Entry Name Selection panel for Applications, enter:
How to Navigate in the Customization Dialog

How to Use a Policy Item as a Fast Path

The Entry Name Selection panel has a special feature that allows you to jump directly to a particular policy item for an entry (skipping the Policy Selection panel for that entry). If you enter an abbreviation for a policy item in the Action column of an entry, SA z/OS immediately takes you to the policy item that best matches what you have typed. For example, if you are in entry type APL, entering I takes you to the Application Information policy item, and entering LC takes you to the Link Instance to Class policy item.

There are no predefined abbreviations for policy items; SA z/OS finds the best match for any input, for example, the beginning of policies or of words, and significant letters. This also ensures that abbreviations work for user-defined policy items.

Note the following special cases:
- Characters reserved for actions will not be matched with policy items. For example, $ is reserved for action “select”. Thus it will not be matched with policy items (such as “SERVICE PERIOD”).
- If there is no best match, you will be taken to the Policy Selection panel, with an error message.

Concurrent User Access

Because 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 several important entry types. These are marked with * on the Entry Type Selection Panel. For other entry types, only one user at a time can access a policy database. See also “Concurrent Customization by Multiple Users,” on page 357 for further information.

How to Cancel Input on a Panel

To cancel your input on the current panel, type CANCEL or C at the Command prompt. CANCEL can be used to discard data entered on the panel, whether you have pressed the Enter key. However, once the message POLICY SAVED is displayed when leaving a panel, the data entered has been saved and cannot be canceled. To delete input after this message is displayed, you must go back to that panel and type over existing values.

How to Create New Policy Objects

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

1. From the Entry Type Selection panel, you select the desired policy object's entry type. For example, if you want to create a new Application object, enter option 6 or APL at the command line.
2. SA z/OS displays the Entry Name Selection panel for the selected entry type. You can either select action NEW from the COMMANDS pull-down or enter the command new entry_type_name at the panel's command line. The entry type name is optional for this command.
How to Create New Policy Objects

3. SA z/OS displays the Define New Entry panel (an example for defining a new Application is shown in Figure 7) for the selected entry type, where you now specify the required information. See Chapter 4, “Entry Type Reference,” on page 75 for more information about how to create new policy objects.

![Figure 7. Define New Entry Panel for an Application](image)

4. If you specify an entry name and press Enter, the subsystem name and job name are defaulted to the entry name, if possible. This is also true for all other entry types where the entry name can be used as a default for further qualifications.

Throughout Chapter 4, “Entry Type Reference,” on page 75, if there is no extra information given about creating a particular policy object, it is created using a generic panel, as shown in Figure 8.

![Figure 8. Sample Generic Define New Entry Panel](image)

For more details about the description fields, see “DESCRIPTION Policy Item” on page 26.
How to Rename a Policy Object

The process of renaming an existing policy object is the same for all types of policy object. From the Entry Name Selection panel, type `r` or `rename` in the **Action** field in front of the entry you want to rename, as shown in **Figure 9**.

![Figure 9. Entry Name Selection Panel: Rename Action](image)

The sample screen shows a list of applications with command `r` in the action field of entry name `ASCH`. Pressing the Enter key will display a rename pop-up panel where you can enter the new entry name, as shown in **Figure 10**.

![Figure 10. Entry Name Rename Pop-up Panel](image)

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

You can leave the rename pop-up panel using END or CANCEL without renaming the policy object.

**Note:** To modify the Enterprise name, edit the policy database and then simply overtype the name, see “Specifying Settings for Policy Databases” on page 304.

How to Delete Policy Objects

There are two ways to delete policy objects; you can:
- Delete a single policy object
- Delete any number of policy objects in one go (bulk deletion)
Note that if a class is selected for deletion and there are objects linked to it that inherit data, the confirmation panel is displayed. If a class is deleted that has only upward links, no confirmation panel is shown. These links are removed without further notification because there is no object left that loses data.

How to Delete a Single Policy Object

To delete a single policy object use action d against the policy object that you want to delete. This displays a confirmation panel; the example shown in Figure 11 lists any links that exist to the policy object.

How to Delete Several Policy Objects in One Go (Bulk Deletion)

To delete several policy objects in one go, type d in the action column against each of the policy objects that you want to delete and press Enter. This displays the confirmation panel shown in Figure 12 on page 23. Note that only policy objects of the same entry type can be deleted in one go.
How to Delete Policy Objects

You can use the Show confirmation for field to set how confirmation of the bulk deletion is handled:

**ALL**
- Only the policy object that is named on the confirmation panel will be deleted. The confirmation panel will then be redisplayed with the name and links of the next policy object that you marked with d.
- This will continue until you change the value in the Show confirmation for field or all the policy objects that you marked with d have been deleted.

**LINKED**
- The policy object that is named on the confirmation panel will be deleted. SA z/OS then processes the rest of the policy objects that you marked with d in the following way:
  - Any policy object that is not involved in links will be deleted without further confirmation.
  - As soon as a policy object with links is encountered the confirmation panel is redisplayed with that policy object’s name and links on it.

**NONE**
- All the policy objects that you marked with d will be deleted at once, without further confirmation.

Note that when a policy object with links is deleted, the links are treated exactly as in “How to Delete a Single Policy Object” on page 22.

How to Find Data in a Policy Item

You can use the FIND/RFIND command for the following policy items:

- **AUTOMATION FLAGS** of entry types APL, ADF, SDF, and MDF (on EXIT and DISABLE TIME panel)
- **MINOR RESOURCES** of entry types APL and MVC
- **MESSAGES/USER DATA** of entry types APL, APG, MTR, and MVC
- **STARTUP** of entry types APL
- **SHUTDOWN** of entry types APL
- **HEALTHSTATE** of entry type MTR
- **OPCA DOMAINID** of entry type ODM
How to Find Data in a Policy Item

- E-T DATA of entry type UET

  The FIND command is used to find and display the next occurrence of a character string in the data that is being edited. You can abbreviate it to F.

  For every FIND command, you must specify the string of characters that is to be located. In most cases, you simply enter FIND, or its abbreviation F, followed by the character string. For example, to find the character string `demo`, enter the following:

  `find demo`

  The cursor is placed at the beginning of the string. Automatic scrolling is performed, if necessary, to bring the string into view. You can use left or right scrolling (by default, the PF10 and PF11 keys) to bring other fields into view.

  Note:

  Only the data that is visible is searched through for the following:
  - The Message ID column on the Message Processing panel for entry types APL, APG and MTR
  - The Keyword column on the UET Keyword-Data Specification panel

  You should therefore consider expanding the column width so that all data is visible before performing a search.

  You can limit the columns that are searched through by the FIND command by specifying a column name or a non-ambiguous abbreviation for it. The string is found if it is completely contained within the designated column. For example, to find the character string `demo` in the Data column on the UET Keyword-Data Specification panel, enter:

  `find demo data`

  To use the same string that was used in the previous FIND command, you can replace it with a * (asterisk). For example, to find the character string `demo` again in the Data column, enter:

  `find * data`

  If you want to find data strings that contain spaces or special characters, you can enclose the string in quotation marks. For example, to find the expression `A = B`, enter:

  `find 'a = b'`

  Normally the FIND command searches for the next occurrence of the specified character string, starting at the top of the displayed page or at the cursor position (if the cursor is within the data portion of the display). You can control the scope of the search by entering the optional parameter ALL. For example, to perform a search for all of the occurrences of the character string `demo`, starting at the beginning of the first line of data and continuing to the bottom, enter:

  `find demo all`

  If the string is found one or more times, the cursor is placed at the beginning of the first occurrence and the string is highlighted. You can use the RESET command to remove the highlighting.
How to Find Data in a Policy Item

A message is displayed indicating the number of times that the string was found. If you press the RFIND PF key (the default is PF5 or PF17), the second occurrence is found, then the third, etc. until the bottom of the data is reached.

If the string is not found within the data, a message is displayed indicating that the characters cannot be found.

The RFIND (repeat find) command can be used to repeat the last FIND command that was entered. Because this command is sensitive to the current position of the cursor, it is normally entered via a PF key. By pressing the RFIND PF key repeatedly, you can find successive occurrences of the specified character string. After the first occurrence of a character string has been found using the FIND command, the RFIND key will find the next occurrence. After the last occurrence of a character string has been found, you can use the RFIND key to continue the search, wrapping around from the bottom line to the top line of the data.

If you enter the optional ALL parameter or a column name as part of the FIND command, you can specify them in any order. Although normally not required, you can avoid ambiguity with the optional parameters by using quotation marks or apostrophes to enclose the character string that is to be found. For example, the following command is ambiguous:

```
find all k
```

It could mean either scan for the string all in the Keyword column, or scan for all occurrences of the string k. In such cases the message Put string in quotes is displayed. You should reenter the command with the search string in quotation marks.

If the FIND command cannot be interpreted, the message Parameter not recognized is displayed. Reenter the command with appropriate parameters.

If you exit a panel, the search string is not retained, so you need to enter a new FIND command if you return to the panel.

Command and User Data Policies

There are policies which show data in a tabular form and have input fields which are longer than the standard screen width of 80 characters. A typical example is the 'Command Processing' panel in the MESSAGES/USER DATA policy for Applications (see Figure 105 on page 167). Any number of commands can be defined with each command specification in one line. The command input field has a length of 159 characters, so to see it in full length it is necessary to scroll to the right.

However, when editing long commands scrolling right and left may be cumbersome and a separate panel with all input fields of one data line on one screen can be invoked with S (select) or E (edit) line commands. Here is an example for one line of the 'Command Processing' panel in the MESSAGES/USER DATA policy:
Command and User Data Policies

This is available for all policies with "long" input fields:
- MESSAGES/USER DATA for entry types APL, APG, MTR, and MVC
- STARTUP for entry type APL
- SHUTDOWN for entry type APL
- HEALTHSTATE for entry type MTR
- E-T DATA for entry type UET
- OPCA DOMAINID for entry type ODM.

Furthermore, these policies have a **Cmd** column which allows you to insert, delete and move lines.

You can use the following line commands in the **Cmd** field to manipulate the entries on this panel:
- **I**nn
  - Insert one or **nn**, if you specify a number, new lines after the selected line
- **R**nn
  - Repeat the selected line one or **nn** times if you specify a number
- **D**nn
  - Delete the selected line, and the following **nn** lines if you specify a number
- **M**nn
  - Move the selected line and the following **nn** lines if you specify a number
- **A**
  - Move lines after the selected line
- **B**
  - Move lines before the selected line.

For these policies, you can use the FIND/RFIND command on this panel to locate data. See "How to Find Data in a Policy Item" on page 23 for more details.

**Common Policy Items**

The following policy items are found on many policy definitions:
- **DESCRIPTION**
- **COPY**
- **WHERE USED**

**DESCRIPTION Policy Item**

[Figure 13 on page 27](#) shows an example of a Description policy item panel (in this case for the SYSTEM entry type)
Both types of description (short and long) can be displayed in the customization dialog, depending on the VIEW option that you choose.

**Short Description**

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

**Long Description**

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

### COPY Policy Item

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

**Copy Process**

The following outlines the process of using an object's COPY policy item when 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) at the command line to create a new object. SA z/OS displays the Define New Entry panel for the policy object that you want to create.
3. Enter information for the new object as prompted by its Define New Entry panel. Exit the panel (using the PF3 key or END command).
4. The Policy Selection panel for the new object is displayed. Select its COPY policy item.
5. A panel listing all the objects that are currently defined with the same entry type is displayed. Select the policy object that you want to copy policy information from. Take note of the following:
   - 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 84).
   - For entry type SYS, copying is only possible for those entries that have the same operating system.
   - For entry type APL, copying is only possible for those entries that have the same application type.
Common Policy Items

- 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 Entry Type" on page 116).

6. When the copy is complete, you return to the Policy Selection panel for the new object. A Copy complete message in the top right corner shows that the copy was successful.

If you copy groups, processors, or zEnterprise ensembles, you may receive the message Partial copy complete, which means that it has not been possible to copy all entries. The help function gives details about the data that was not copied.

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

Information that is Copied
The following indicates what information is or is not copied by the COPY policy item:

1. All source object data is copied to the target object for all entry types with two exceptions:
   - Any data that has to be entered during the creation process on the associated Define New Entry panel for the target object is not copied from the source object.
   - The short and extended description on the Define New Entry panel is copied from the source object whenever this is not already specified for the target object.

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

3. None of the links to other objects that are higher up in the hierarchy of objects are copied from the source to the target. For example, if you copy an application that is linked to an application group, the copied target application is not linked to that application group.

4. If you copy a class or instance with “upward” links (that is, links to a class), these links are copied. However, if you copy a class that has “downward” links, they are not copied.

5. Links to members of sysplex groups (that is, links to subgroups, systems, or Sysplex Application Groups) are not copied. This is to avoid conflicts such as having the same system defined in more than one sysplex.

6. Links from systems to processors are not copied to avoid conflicts such as having the same system defined for more than one processor.

7. Resource links of application groups are not copied.

WHERE USED Policy Item

The WHERE USED policy item is effectively a container for references to existing policy objects. The container may be empty. To put a reference to a policy object in the container, select that policy object from within WHERE USED. SA z/OS automatically ensures that each reference in WHERE USED is matched by a reverse reference in an appropriate policy item of the other policy object (details further on in this section). Thus, WHERE USED actually contains one direction of mutual references. These mutual references are called “WHERE USED links” or “links”, for short.
What Policy Objects Can Be Linked?

Whether it is possible to have a link between two policy objects depends on their entry types. The entry types that SA z/OS supports are arranged in the hierarchy as shown in Figure 14 on page 30. The figure shows the various counterparts of WHERE USED. By following the arrows backwards, you can tell what information you will find in WHERE USED. That is, the entry type where a given policy object may be used.

Key to the figure:
Gray boxes in Figure 14 on page 30 represent entry types, and arrows represent policy items. When an arrow is attached to a white box inside a gray box (rather than to the gray box itself), the policy item that the arrow represents is only present on policy objects with the specific property named in the white box. An arrow named z from an entry type named X to an entry type named Y means:

• Every policy object in entry type X has a policy item named z. This policy item contains zero, one or more references to policy objects in entry type Y.
• Conversely, every policy object in entry type Y has the WHERE USED policy item. Any references in its WHERE USED policy item are to policy objects in entry type X.

With a few exceptions, the name of an arrow matches the name of the entry type that it points to (that is: the one that is lower in the hierarchy).
Common Policy Items

Figure 14. Hierarchy of Entry Types
How to Create, View or Remove Links

Because links are *mutual* references, there are two alternative approaches to create, view or remove them.

**Bottom-Up Approach**: 

**Step 1.** In the lower policy object (regarding the hierarchy in Figure 14 on page 30), open the WHERE USED policy item. This displays a list of all the policy objects where the current one *may* be used. Those that have a status of SELECTED are the ones where the current one *is* actually used at present.

**Step 2.** To create a link, apply action “s” to the policy object in question.

**Step 3.** To remove a link, apply action “m” to the policy object in question.

**Top-Down Approach:**

**Step 1.** In the upper object (regarding the hierarchy in Figure 14 on page 30), open the policy item that takes you to the intended lower type, as shown in Figure 14 on page 30. This displays a list of all the policy objects that the current one *may* use. Those that have a status of SELECTED are the ones that the current one *does* actually use at present.

**Step 2.** To create a link, apply action “s” to the policy object in question.

**Step 3.** To remove a link, apply action “m” to the policy object in question.

Select ...

For almost every entry type there are policies to link 'child' entry types. These link panels are just the reverse of the WHERE USED policies of the child entry types. Here is an example for Application Groups with the policies to link Applications (APL), Trigger (TRG) and Service Periods (SVP).

<table>
<thead>
<tr>
<th>ACTION</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>Policy Selection Row 1 to 13 of 14</td>
</tr>
<tr>
<td>SCROLL</td>
<td>PAGE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Entry Type</th>
<th>PolicyDB Name</th>
<th>Enterprise Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ApplicationGroup</td>
<td>USER_PDB</td>
<td>USER_ENTERPRISE</td>
</tr>
<tr>
<td>Base_APPL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Action</th>
<th>Policy Name</th>
<th>Policy Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>Select applications for system APG</td>
</tr>
<tr>
<td>APPLICATIONS</td>
<td>TRIGGER</td>
<td>Select trigger</td>
</tr>
<tr>
<td>SERVICE PERIOD</td>
<td></td>
<td>Select service period</td>
</tr>
</tbody>
</table>

*Figure 15. Policy Selection Panel*

When you select one of these policy items, a list of entry names is shown as in Figure 16 on page 32.
Entries which are already linked are shown with Status ‘SELECTED’.

With action SELECT or ADD (or abbreviations S and A) the entries can be selected while with action REMOVE (or M) any selection can be reverted. It is also possible to make a single selection without scrolling via the command by entering the command SELECT or REMOVE with an entry name as parameter.

Usually multiple entries can be selected, but there are a few policies where only one linked entry is allowed, for example, the policy SERVICE PERIOD. For those policies always the last selection is taken. That means if an entry is already selected, and then another selection via the Action column is specified, then the indication ‘SELECTED’ for the already selected entry is removed automatically. And if multiple selections are specified in the Action column then the bottom selection is used.

Initially the displayed list is sorted alphabetically, but you can use the SORT command to get all selected entries to the top. The SORT command has the column name as parameter, so in the example above the command ‘SORT STATUS’ would move all selected applications to the top of the screen. The column can be abbreviated, so ‘SORT S’ would also move all selected applications to the top, while ‘SORT A’ would show the applications again sorted alphabetically by name.

The SORT command is also available on the WHERE USED panels. On some of those panels, entries from different types are shown, for example, WHERE USED for Trigger shows Applications and ApplicationGroups. On these panels the command ‘SORT T’ would show the names sorted by Type and for each type sorted by name.
Chapter 3. 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 59
- “How to Use Triggers and Events” on page 60
- “How to Use User E-T Pairs” on page 67
- “How to Implement Message Processing with User-Defined Data” on page 69

How to Work with Resources

This section contains a scenario on how to group resources for applications in a complex configuration.

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, see “Grouping the Resources” on page 35.

Grouping applications and nesting application groups is a process that can be logically divided into multiple steps:
- “Step 1: Identify Resources”
- “Step 2: Group the Resources” on page 34
- “Step 3: Nest the Groups” on page 35
- “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 that represent a data warehouse and a Web server:
- The data warehouse requires the following resources:
  - VTAM
  - TCP/IP
  - DB2 (consisting of components DB21MSTR, DB21IRLM)
  - CICS (consisting of components CIC5MA5IN, CICS1A1R1, CPSMCMAS, CPSM5UI, CICS1TOR, CICS1FOR)
  - DWH
- The Web server requires the following resources:
  - VTAM
  - TCP/IP
  - DB2 (consisting of components DB21MSTR, DB21IRLM, DB21DIST,
    DB21DBM1)
How to Work with Resources

- WBSRV

**Step 2: Group the Resources**

Some of the required resources that were identified in the previous step are candidates for aggregation:

- Resources that are made of multiple components. In our example, these are:
  - CICS1AR1, CPSMCMAS, and CPSMWUI, which could be aggregated into a group called CICSAOR, so that they can be automated as a unit.
  - The group CICS, which will contain resources CICSMAIN, CICS1TOR, CICS1FOR and the aggregated CICSAOR group.
  - The group DB2, which will contain resources DB21MSTR, DB21IRLM, DB21DIST, and DB21DBM1.

- 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. We will aggregate them into a group called NET.
  - DB21MSTR, DB21IRLM, DB21DIST, and DB21DBM1, which are used by the data warehouse and the Web server. We will aggregate them into a group called DB2.

- Finally, we implement the data warehouse as the group DWHAPG and the Web server as the group WBSRVAPG to allow nesting the other groups into them as required later (see “Step 3: Nest the Groups” on page 35).

*Figure 17* shows a possible aggregation of all components into groups.

*Figure 17. Application Groups*

In *Figure 17*, oval labels represent application groups and rectangular labels represent applications. For example, application group CICS contains applications CICSMAIN, CICS1TOR, CICS1FOR and the nested application group CICSAOR.
**Step 3: Nest the Groups**

Figure 18 shows how the application groups are nested in this environment. From an operator's point of view, a start request against WBSRVAPG should start all imbedded components in a reasonable order. The same is true for requests against all other groups such as, for example, NET and DB2.

**Step 4: Match Resources to Systems**

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

In our example, a possible decision could be:

- The Web server group WBSRVAPG should run on SYSTEM1.
- The data warehouse group DWHAPG should run on SYSTEM2 and (if SYSTEM2 is not available) on SYSTEM3.
- The DB2 group, containing applications DB21MSTR, DB21IRLM, DB21DIST, and DB21DBM1, should run on SYSTEM2 and on SYSTEM3.
- The NET group, containing applications VTAM AND TCPIP, should run in the entire sysplex.
- The CICS group, containing applications CICSMAIN, CICS1FOR, CICS1TOR and the application group CICSAOR should run on SYSTEM2 and on SYSTEM3.

See “How to Work with Resources” on page 33 for detailed information on how to implement this scenario using the customization dialog.

**Grouping the Resources**

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

- “Step 1: Creating the Policy Objects that Make Up the Sysplex”
- “Step 2: Creating the Applications” on page 38
- “Step 3: Creating the Application Groups” on page 39

**Step 1: Creating 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.
Step 1: Creating the Policy Objects that Make Up the Sysplex

Creating a sysplex consists of:
- Creating the systems
- Creating the sysplex group
- Linking the systems to the sysplex group

Step 1a: Creating 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 102 to see some sample customization dialog panels 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 currently displays an empty list.
2. Enter NEW SYSTEM1 at the command line.
   The Define New Entry panel for systems is displayed. The Entry name field is already filled with SYSTEM1. The default operating system is MVS.
3. Press Enter to have the Image/System name field default to the same as the Entry name field. This also enters default values in the Heartbeat Interval and Missing Heartbeat Delay fields. Press PF3. The Policy Selection panel for systems is displayed, showing a message indicating that the system has been 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 1b: Creating the Sysplex: You may want to read the information in “Creating a New Group” on page 84 to see some sample customization dialog panels before you continue with the scenario.

1. On the Entry Type Selection panel, enter option 2 or GRP to go to the Entry Name Selection panel for groups, which displays an empty list.
2. Enter NEW SYSPLEX1 at the command line.
   The Define New Entry panel for groups is displayed; the Entry name field is already filled with SYSPLEX1.

3. The Group Type field defaults to SYSPLEX which is correct here because you want to create a sysplex group.
   Press PF3 to exit this panel. The Policy Selection panel for Groups is displayed, showing a message that the group has been created.

4. Consider specifying default definitions for all your sysplex resources (that is, sysplex application groups) using the Sysplex Defaults entry type (XDF) and linking it to your sysplex group, as follows:
   a. Return to the Entry Type Selection panel and select the Sysplex Defaults entry type by entering XDF or 40. This displays the Entry Name Selection panel for Sysplex Defaults.
   b. To create a new entry called, for example, XDF1, enter new xdf1 at the command line. This displays the Define New Entry panel with XDF1 in the Entry name field.
      Press PF3 to exit this panel. The Policy Selection panel for Sysplex Defaults is displayed, showing a message that the entry has been created.
   c. Select the RESOURCE INFO policy item to display the Sysplex Resource Information Defaults panel and specify the needed data. Press PF3 to save the changes and return to the Policy Selection panel.
   d. Select the WHERE USED policy item to display the Where Used panel. Select the SYSPLEX1 sysplex by typing an S in the Action column and pressing Enter. Press PF3 to save the changes and return to the Policy Selection panel.

5. On the Policy Selection panel for the SYSPLEX Group entry type (which you can reach via the Entry Type Selection panel), select the SYSTEMS policy item. The Systems for Group panel is displayed showing the three systems defined in “Step 1a: Creating the Systems” on page 36.

6. Select all three systems with an S and press PF3 until you return to the Entry Type Selection panel.
   Now your environment contains a sysplex with three z/OS systems.
**Step 1: Creating the Policy Objects that Make Up the Sysplex**

---

**Background: Ways to Build a Sysplex**

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

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

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

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 several systems, but systems can only be linked to one sysplex group (either directly or indirectly).

The following figure shows 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: Creating 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 140 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** at the command line.
   - The Define New Entry panel for applications is displayed and the **Entry name** field is already filled with **VTAM**.
3. Press Enter to have the **Subsystem Name** and **Job Name** fields filled with their default value. These fields are mandatory. Both of them default to the
Step 2: Creating the Applications

Application’s Entry name (see "Subsystem Name" on page 141 and "Job Name" on page 144 for more information on this defaulting mechanism). So, in this case, they default to VTAM.

Background: Application Resource Name

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

4. Press PF3 to leave this panel. The Policy Selection panel is displayed, showing a message that the application has been 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 using the values from Table 5.

Table 5. Entry Details for Applications in the Scenario

<table>
<thead>
<tr>
<th>Entry Name</th>
<th>Application Type</th>
<th>Subtype</th>
</tr>
</thead>
<tbody>
<tr>
<td>WBSRV (Web Server)</td>
<td>blank</td>
<td>–</td>
</tr>
<tr>
<td>DWH (Data Warehouse)</td>
<td>blank</td>
<td>–</td>
</tr>
<tr>
<td>CICSMAIN</td>
<td>CICS</td>
<td>TOR</td>
</tr>
<tr>
<td>CICSITOR</td>
<td>CICS</td>
<td>TOR</td>
</tr>
<tr>
<td>CICS1FOR</td>
<td>CICS</td>
<td>FOR</td>
</tr>
<tr>
<td>CICS1AR1</td>
<td>CICS</td>
<td>AOR</td>
</tr>
<tr>
<td>CPSMCMAS</td>
<td>CICS</td>
<td>CPSM</td>
</tr>
<tr>
<td>CPSMWUI</td>
<td>CICS</td>
<td>WUI</td>
</tr>
<tr>
<td>DB21MSTR</td>
<td>DB2</td>
<td>MSTR</td>
</tr>
<tr>
<td>DB21IRLM</td>
<td>DB2</td>
<td>IRLM</td>
</tr>
<tr>
<td>DB21DIST</td>
<td>DB2</td>
<td>DIST</td>
</tr>
<tr>
<td>DB21DBM1</td>
<td>DB2</td>
<td>DBM1</td>
</tr>
<tr>
<td>TCPIP</td>
<td>blank</td>
<td>–</td>
</tr>
</tbody>
</table>

Now all subsystems (applications) are defined to your environment.

Step 3: Creating the Application Groups

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

According to the example from "A Grouping Scenario" on page 33, the groups should run on the following systems:
- WBSRVAPG should run on SYSTEM1.
- DWHAPG should run on SYSTEM2 and alternatively on SYSTEM3.
- DB2 should run on SYSTEM2 and SYSTEM3.
- NET should run in the entire sysplex.
- CICS should run on SYSTEM2 and on SYSTEM3.
Step 3: Creating the Application Groups

---

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 can be located on multiple systems in one or more sysplexes.
- System Application Groups are located on single systems.

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

---

Step 3a: Create Sysplex Application Group CICSAOR: You start with the creation of the sysplex application group CICSAOR because CICSAOR contains only applications that have already been defined (CICS1AR1, CPSMCMAS, CPSMWUI).

You may want to read the information in ["Creating a New ApplicationGroup" on page 117](#) to see some sample customization dialog panels 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` at the command line. The Define New Entry panel for application groups is displayed. The **Entry name** field has already been filled with CICSAOR.

3. Check that the **Application Group Type** field contains SYSPLEX and the **Nature** field contains BASIC.
Step 3: Creating the Application Groups

---

**Background: Nature of an Application Group**

In SA z/OS application groups can have one of three natures, depending on how many components must be available:

**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 their primary component fail 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 their components fail 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 a MOVE group, for example, may not be appropriate for a SERVER group).

In this scenario, the application group consists of three applications (CICS1AR1, CPSMCMAS, and CPSMWUI) that should all be available. You should therefore make it a BASIC Group.

4. Type CICSA0R in the Automation Name field.

---

**Background: Application Group Resource Name**

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

If you do not specify an Automation Name, no resource is generated for the application group. Nevertheless, the application resources for all applications that are linked to this application group are created and become available for automation.

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

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

7. Select the sysplex group and press PF3. You briefly see the Command Progress Display pop-up window, showing that the resource has been created, before it disappears automatically.

You can see the generated resources in the GENERATED RESOURCES policy item of the application group. Return to the Policy Selection panel for CICSA0R.
Step 3: Creating the Application Groups

---

**Background: Generating Sysplex Application Group Resources**

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

Only one resource is created (rather than one resource per system) and the resource name does not have a system identifier.

CICSAOR has an Automation Name; therefore a resource for this application group is generated.

The current configuration with application group CICSAOR is shown in Figure 19.

---

**Figure 19. Scenario: Current Configuration with Sysplex Application Group CICSAOR**

8. On the Policy Selection panel for CICSAOR select the APPLICATIONS policy item. This displays the Applications for ApplicationGroup panel with all of the applications.

9. Select CICS1ARI, CPSMCMAS, and CPSMWUI and press PF3 (END) to return to the Policy Selection panel for CICSAOR.
**Background: Generating Application Resources**

Application resources are generated when applications are linked to systems. This is the case for CICS1AR1, CPSMCMAS, and CPSMWUI, which are now linked to CICSAOR. There are also linked indirectly via sysplex SYSPLEX1 to SYSTEM1, SYSTEM2, and SYSTEM3.

One resource is generated for each linked system.

Again, you see the Command Progress Display pop-up window telling you that the resources are being created. For example, for CICS1AR1, the following resources are generated:

- CICS1AR1/APL/SYSTEM1
- CICS1AR1/APL/SYSTEM2
- CICS1AR1/APL/SYSTEM3

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

**Note:** These application resources are deleted if one of the following happens:
- The application is deleted.
- The application group CICSAOR is deleted.
- Other parts of the link chain APL-APG-GRP-SYS are deleted or delinked (that is, if SYSPLEX1 or any of the systems SYSTEM1, SYSTEM2, or SYSTEM3, or the links between them are deleted).

The current links are shown in Figure 20.

![Figure 20. Scenario: Current Configuration with Applications CICS1AR1, CPSMCMAS, CPSMWUI](image)

10. On the Policy Selection panel for CICSAOR select the RESOURCES policy item. This displays the Select Resources panel with 9 application resources selected, as shown in Figure 21 on page 44.
Step 3: Creating the Application Groups

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

When you press Enter, the Number of selected Resources field is now 6.

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

13. Select the APPLGROUP INFO policy item to display the Application Group Information panel for CICSAOR.

14. Press PF8 to scroll to the bottom of the input field section and change the Automatically link Application-Resources field to NO.

You need to do this because the application group definitions are now complete. Further modifications within the sysplex should not affect the completed resource selections on the Select Resources panel.

**Background: Automatically link Application-Resources Field**

You should set this input field to NO if the definition of an application group resource (with its linked applications, its nature and its behavior and so on) is complete. This prevents any possible changes in the automation policy, such as 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. That is, the number of generated application resources or linked application resources, and possibly the behavior of the application group resource will change. For example, if you link another system to SYSPLEX1, this produces three more resources in the CICSAOR application group, 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 completed application group definition.
Step 3: Creating the Application Groups

15. Press PF3 until you return to the Entry Name Selection panel.
   The CICSAOR application group is now complete. If a start request is issued
   against CICSAOR, start requests for CICS1AR1, CPSMCMAS, and CPSMWUI
   are also automatically issued on SYSTEM2 and SYSTEM3.

Step 3b: Create System Application Group DB2: The DB2 application group runs
on two specific systems, so you define it as a system application group.

1. Enter NEW DB2 at the command line on the Entry Name Selection panel for
   application groups. This displays the Define New Entry panel for application
   groups and the Entry name field already filled with DB2.

2. Enter the required values on this panel:
   a. In this scenario DB2 runs permanently on only two of the three systems of
      SYSPLEX1 (SYSTEM2 and SYSTEM3), so type SYSTEM in the Application
      Group Type field.
   b. You want DB2 be started with all of its components but without specific
      backup considerations, so leave the Nature field as BASIC.
   c. To generate resources you need an automation name, so type DB2 in the
      Automation Name field.

Press PF3. The Policy Selection panel is displayed indicating that the
application group has been created.

Note: A resource is not created for the APG at this point because the
application group is not yet linked to any systems.

3. Select the WHERE USED policy item. This displays the Where Used panel
   with the three systems that you created in “Step 1a: Creating the Systems” on
   page 36.

4. DB2 should only run on SYSTEM2 and SYSTEM3. Select SYSTEM2 and
   SYSTEM3 and press PF3 (END). You will see the Command Progress Display
   pop-up window telling you that the following 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 a link to a system.
One resource is generated for each linked system.

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 22 on page 46.
Step 3: Creating the Application Groups

5. On the Policy Selection panel for DB2, select the APPLICATIONS policy item to link the required applications. This displays the Applications for ApplicationGroup panel with all 12 applications.

6. Select DB21MSTR, DB21IRLM, DB21DIST, and DB21DBM1, and press PF3 (END) to return to the Policy Selection panel.

   Now DB21MSTR, DB21IRLM, DB21DIST, and DB21DBM1 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.

   One resource is generated for each system link of each application. The Command Progress Display pop-up window indicates that the following resources are generated:
   - DB21IRLM/APL/SYSTEM2
   - DB21IRLM/APL/SYSTEM3
   - DB21DBM1/APL/SYSTEM2
   - DB21DBM1/APL/SYSTEM3
   - DB21DIST/APL/SYSTEM2
   - DB21DIST/APL/SYSTEM3
   - DB21MSTR/APL/SYSTEM2
   - DB21MSTR/APL/SYSTEM3

   The current configuration is shown in Figure 23.

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

   This displays the Select System panel, showing the number of resources that have been selected for each of the two systems. You can compare the resulting panel on your system with Figure 24 on page 47.

---

**Figure 22. Scenario: Current Configuration with Application Group DB2**

**Figure 23. Scenario: Current Configuration with Applications Linked to DB2**
8. A System Application Group can be linked to several systems. This generates application resources for those systems. So to work with the resource links, you first have to specify which system you want to get the resources for. This is done on the Select System panel by selecting one of the systems.

On the Select System panel, select either SYSTEM2 or SYSTEM3 and press Enter. This displays the Select Resources panel with, for example, the two APL Resources for SYSTEM2. You can compare the resulting panel on your system with Figure 25.

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

9. On the Select Resources panel, press PF3 (END) or CANCEL. This returns you to the Select System panel, where you can also view the Select Resources panel for the APL resources in SYSTEM3 to check the created links for SYSTEM3.

10. Press PF3 (END) twice to return to the Policy Selection panel for the DB2 application group.

11. Select the APPLGROUP INFO policy item to display the Application Group Information panel. Press PF8 to scroll down the input field section and change the Automatically link Application-Resources into APG field to NO.

The DB2 application group is now complete.

12. Press PF3 to return to the Entry Name Selection panel for application groups.

The application group DB2 is now complete. If a start request is issued against DB2 on SYSTEM2, start requests to the application resources DB21MSTR,
Step 3: Creating the Application Groups

DB21IRLM, DB21DIST, and DB21DBM1 are also automatically issued for SYSTEM2. A start request against application group DB2 on SYSTEM3 does the same for SYSTEM3.

Step 3c: Create System Application Group CICS: You create the CICS application group with a nature of BASIC and with links to three applications (CICSMAIN, CICSITOR, CICS1FOR) that you defined in “Step 2: Creating the Applications” on page 38.

1. To create a System Application Group called CICS, repeat steps 2–8 of “Step 3b: Create System Application Group DB2” on page 43 accordingly. In the equivalent of step 8, select the applications CICSMAIN, CICSITOR, and CICS1FOR on the Applications for ApplicationGroup panel.

2. On the Policy Selection panel for application group CICS, select the RESOURCES policy item. The Select System panel is displayed as shown in Figure 26.

    Select SYSTEM2. A panel is displayed similar to the one shown in Figure 27.

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

4. Press PF3 to return to the Entry Name Selection panel for application groups.
Step 3: Creating the Application Groups

The CICS application group is now complete. If a start request is issued against
CICS, start requests are also issued for the application resources CICS1FOR,
CICSMAIN, and CICS1TOR (on SYSTEM2 and SYSTEM3).

The current configuration is shown in Figure 28.

![Figure 28. Scenario: Current Configuration with Applications Linked to DB2](image)

Step 3d: 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 29.

![Figure 29. Scenario: Select Resources Panel for Application Group NET](image)

Step 3e: Create MOVE Group DWHAPG: The Data Warehouse Group DWHAPG
should run on one system (SYSTEM2) and, if something goes wrong on SYSTEM2,
on SYSTEM3. This application group is created with the nature MOVE.
Step 3: Creating the Application Groups

Background: What Are Move Groups?
All previous application groups were of nature BASIC, which means that all of their components are equivalent and all of them need to be available before the application group is considered to be available.

A MOVE Group, in contrast, is designed for backup solutions. A MOVE Group can have many components, but exactly one is expected 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, but additional preparation is needed.

1. From the Entry Type Selection panel, enter option 5 or APG. The Entry Name Selection panel is displayed with four application groups: CICS, CICSAOR, DB2, and NET.
2. Enter NEW DWHAPG at the command line. The Define New Entry panel is displayed with the Name field already filled with DWHAPG.
3. Enter the required values on this panel:
   a. The application group DWHAPG should run alternatively on two of the three systems, so leave the Application Group Type field as SYSPLEX.
   b. You want to start the components of DWHAPG exactly once, but with backups. Change Nature to MOVE.
   c. To generate resources you need an automation name, so type DWHAPG in the Automation Name field.
4. Press PF3 to exit the panel. The Policy Selection panel for Groups is displayed showing a message that the application group has been created.
5. Select the WHERE USED policy item to display the Where Used panel with sysplex group SYSPLEX1.
6. Select the sysplex group and press PF3.
   The Command Progress Display panel informs you that the DWHAPG/APG resource has been created.
7. Select the RESOURCES policy item. The Select Resources panel shows an empty list of resources. The reason is that you did not link any applications to the application group and so no application resources were generated.
   Note that the Availability Target is 1 (exactly 1 resource is expected to be up in a MOVE Group). You cannot change this value.
8. Change the Show all Resources field to YES and press Enter.
   The Select Resources panel is displayed, as shown in Figure 30 on page 51, with all valid application group resources:
   • Sysplex Application Group resources in the same sysplex: CICSAOR/APG and NET/APG
   • System Application Group resources on one of the systems in the sysplex: CICS/APG/SYSTEM2, CICS/APG/SYSTEM3, DB2/APG/SYSTEM2, and DB2/APG/SYSTEM3
Step 3: Creating the Application Groups

However, this is not yet what you want. The purpose of your group is to start all of these components and the DWH application on either SYSTEM2 or SYSTEM3.

You need to define two additional application groups with the same contents, one for SYSTEM2 and one for SYSTEM3. Their names will be DWHAPG2 and DWHAPG3. DWHAPG will have these two APGs as members and start them alternatively.

You must define these groups as System Application Groups because they must contain the System Application Group CICS on SYSTEM2 and SYSTEM3.

9. Press PF3 until you return to the Entry Name Selection panel and enter NEW DWHAPG2 at the command line (you could also use the fast path =APG;NEW DWHAPG2).

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

10. Enter the required values on this panel:
   a. DWHAPG2 should run on a single system, so change the Type to SYSTEM.
   b. Because you want to start all of the components of DWHAPG2, the Nature must be BASIC.
   c. To generate resources you need an automation name, so type DWHAPG2 in the Automation Name field.
   d. Press PF3 to exit the panel.

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

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

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

The application resource DWH/APL/SYSTEM2 is generated and linked to the application group resource DWHAPG2.

14. Select the RESOURCES policy item. The Select Resources panel with the APL Resource DWH/APL/SYSTEM2 is displayed. Select this APL Resource, if it is not already selected.
Step 3: Creating the Application Groups

15. Change the **Show all Resources** field to YES and press Enter.

As a result, two application group resources are displayed together with the one application resource, as shown in Figure 31.

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

Select CICS on SYSTEM2 (the resource CICS/APG/SYSTEM2), DB2 on SYSTEM2 (the resource DB2/APG/SYSTEM2) and DWH on SYSTEM2 (the resource DWH/APL/SYSTEM2).

**Background: Defining Loops in Groups**

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

17. The Select Resources panel with one application resource and two application group resources selected should now be as shown in Figure 32 on page 53.
Step 3: Creating the Application Groups

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

19. Now create the DWHAPG3 system application group in the same way as for the DWHAPG2 application group, as described in steps 9 through 12, and link it to system SYSTEM3.

20. You want the same application members for the DWHAPG3 system application group as you already defined for DWHAPG2, but with their focus on SYSTEM3. You can do this by copying DWHAPG2. Select the COPY policy item and then select DWHAPG2 in the Select Entry for Copy panel.

   Note:
   You should be aware that the COPY function only copies certain source object data and links. For more details, see “Information that Is Copied” on page 28.

The Command Progress Display pop-up window shows the resource being created.

After the copy process you are at the Policy Selection panel for DWHAPG3.

21. Select the RESOURCES policy item and link the resources that you need for SYSTEM3. The result should look like the panel shown in Figure 33.

Press PF3 until you return to the Entry Name Selection panel.
Step 3: Creating the Application Groups

22. Now you have all the components for the DWHAPG move group. Select DWHAPG on the Entry Name Selection panel, and then select the RESOURCES policy item (alternatively, you can enter res in the Action field for DWHAPG).

23. The Select Resources panel for DWHAPG is displayed.
   Change the Show all Resources field to YES and press Enter.
   The Select Resources panel with eight APG resources is displayed, as shown in Figure 34.

24. To set the preferences, define:
   a. The DWHAPG2 group on SYSTEM2 (DWHAPG2/APG/SYSTEM2) as primary by typing p in its Action field
   b. The DWHAPG3 group on SYSTEM3 (DWHAPG3/APG/SYSTEM3) as secondary by typing 500 in its Preference field

   This is shown in Figure 35 on page 55.
Step 3: Creating the Application Groups

Press Enter. Figure 36 shows the Select Resources panel after pressing Enter, with the P action converted to the corresponding value of 700.

Note:

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

- (P)RIMARY sets a primary preference of 700.
- SECON(D)ARY sets a secondary preference of 400.

Press PF3 to return to the Policy Selection panel for DWHAPG. A pop-up window indicated that the preference values have been saved.
Step 3: Creating the Application Groups

Background: Using Preferences to Define Specific Roles

The Resources panel for MOVE groups looks slightly different to the panel 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 for the resources that are linked to a group.

You want DWHAPG2 on SYSTEM2 be started and, if this fails, you want to start DWHAPG3 on SYSTEM3. To define this, start with preference values 700 for DWHAPG2 and 500 for DWHAPG3. They have the following meaning:

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

500  The resource is started when the higher preference alternative becomes unavailable. It continues running when normal preference alternatives (that is, with a value of 700) become available again. It stops at the next (scheduled) outage. (Use for a backup resource).

For more details about preference values, see “RESOURCES Policy Item” on page 132.

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

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

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

26. Press PF3 to return to the Entry Name Selection panel for application groups.

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

Step 3f: Create Sysplex Application Group WBSRVAPG: In this step you create the Sysplex Application Group WBSRVAPG as a SERVER group.
Step 3: Creating the Application Groups

Background: What Is a SERVER Group?
SERVER groups are created when only a certain number of linked resources need to be available to consider the application group as being available.

A SERVER group with an Availability Target of 1 and appropriate preference values can behave like a MOVE group.

A SERVER group with an Availability Target of *ALL and all the preferences of its members set to 700 acts like a BASIC group.

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

1. Enter NEW WBSRVAPG at the command line.
   The Define New Entry panel is displayed with the Entry name field already as WBSRVAPG.

2. Enter the required values on this panel:
   a. The WBSRVAPG application group runs mainly on SYSTEM1. However, WBSRVAPG will have some members that run on other systems in the sysplex, so leave the Application Group Type field as SYSPLEX.
   b. WBSRVAPG should start with all of its components, but it is acceptable if DB2 cannot be started. So change the Nature of the WBSRVAPG group to SERVER.
   c. Change the default preference in the Default Preference field to 444. This is the initial value for all resources that will be linked automatically.
   d. For resource generation you need an automation name; so enter WBSRVAPG in the Automation Name field.

The input in the Define New Entry panel should now look similar to Figure 37.
Step 3: Creating the Application Groups

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

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

5. Select the sysplex group and press PF3. The Command Progress Display pop-up window shows that the resource WBSRVAPG/APG has been created.

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

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

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

   You can see that the preference for the linked resources is the value that you entered when you created the WBSRVAPG application group.

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

   a. Change the Show all Resources field to YES and press Enter to see the complete list of selectable resources.

   b. Now you need to change the preferences so that the WBSRVAPG application group is started on SYSTEM1 with WBSRV and NET as mandatory components and DB2 as optional. To achieve this:

      1) Set the Availability Target to 2.

      2) Enter 1000 in the Preference field for the mandatory components (WBSRV/APL/SYSTEM1 and NET/APG).

      3) Enter m in the Action field for WBSRV/APL/SYSTEM2 and WBSRV/APL/SYSTEM3 to deselect them.

      4) Enter a value between 501 and 999 in the Preference field for DB2/APG/SYSTEM2 and DB2/APG/SYSTEM3.

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

   If you change the Show all Resources field to NO, your definitions should look like Figure 38 on page 59.
Step 3: Creating the Application Groups

8. Press PF3, select the APPLGROUP INFO policy item in the Policy Selection panel and change the \textbf{Automatically link Application-Resources into APG} field to \textbf{NO} on the Application Group Information panel.

The WBSRVAPG application group is complete. If a start request is issued against WBSRVAPG, both WBSRV and NET must be successfully started for WBSRVAPG to be available.

\textbf{How to Use Schedules (Service Periods)}

You can extend the scenario described in \textit{“A Grouping Scenario” on page 33} using service periods that automatically control the startup and shutdown of resources. Service periods can be linked to APLs, APGs, and MTRs.

You will extend the scenario so that:

\begin{itemize}
  \item The data warehouse application is available from 9 a.m. to 10 p.m.
  \item The Web server is available from 4 a.m. through 11 p.m.
\end{itemize}

Three service periods are needed:

\begin{itemize}
  \item SWBSRV linked to the WBSRVAPG (Web server) application group
  \item SDWH linked to the DWHAPG (data warehouse) application group
  \item SDB2DWN linked to the DB2 application group
\end{itemize}

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

\textbf{Schedule Override}

The Web server needs to be available until midnight. So the operator just needs to override the SWBSRV service period (changing 11 p.m. to 12 a.m.). 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 11 p.m. To remove this request at midnight, you could either use INGREQ command with the expiration
How to Use Schedules (Service Periods)

time parameter, which will automatically remove the request. Or, at 12 a.m., you could use the INGVOTE command to remove this request.

Early Start

At 3 a.m. the Web server application needs to be started.

The operator simply creates an override for SWBSRV that says to start the Web server at 3 a.m. and keep it up until 11 p.m.. 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 4 a.m., the resources will not be stopped because 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 section describes a scenario on how to define automatic starts and stops that are based either on fixed time intervals (service periods) or on special conditions (events).

This scenario is based on the environment that you created in “How to Work with Resources” on page 33 and describes:

- What environment you need and when it should be available, see “Scenario Environment and Behavior”
- What policy objects and links you need and why you need them, see “A Possible Solution” on page 61
- How to enter these in the customization dialog “Implementing the Solution” on page 63

Note that, like service periods, triggers can be linked to APLs, APGs, and MTRs

Scenario Environment and Behavior

Assume that you are in the environment that you created in “How to Work with Resources” on page 33 and that you want to set up the following CICS applications:

- One terminal-owning region (TOR) on system SYSTEM3
- One application-owning region (AOR) on SYSPLEX1 to support this TOR

They should meet these requirements:

1. Between 7 a.m. and 6 p.m. on weekdays, and 8 a.m. and 2 p.m. on Saturdays:
   a. The TOR should be available.
   b. The AOR should be available on one of the systems in SYSPLEX1.

2. Before any of the CICS applications are started, certain related data sets need to be backed up.

3. A TOR does not make sense without an AOR, so you need relationships that ensure that:
   a. The TOR should not be started until the AOR is available on one of the systems.
b. If the AOR has to be shut down (for example, in an emergency) so that it is no longer available on any of the systems in the sysplex, the TOR must be shut down first.

**Note:**

It does not matter which system the AOR is available on and it need not be the same system throughout the specified periods. Thus it is enough to start the AOR on one of the systems and switch to another system when the AOR cannot or does not continue to run on its current system.

### A Possible Solution

Figure 39 shows the policy objects and connections that you need to define in the customization dialog.

**Legend:**

- **A** = Policy object defined in a previous scenario
- **A** = Policy object to be defined in this scenario
- **SYS** = System
- **GRP** = Group
- **APL** = Application
- **SVP** = Supervisor
- **TRG** = Trigger
- **EVT** = Event

For all lines and arrows:
- **thin** = defined in a previous scenario
- **thick** = to be defined in this scenario

**Values:**

- **A** = Link between A and B

**Resources:**

- **A** = The resource generated from A has the resource generated from B as a member resource
- **B** = The resource generated from A is a member of the resource generated from B

**Relationships:**

- **A** = Every resource generated from A is dependent on one resource generated from B in a HASPARENT relationship

**Figure 39. Policy Objects in the Solution**

**Note:**

Some of the policy objects that you need here were already created in “How to Work with Resources” on page 33. So you can start from that existing policy database. If you do, you will have more entries and links in your policy database than are present in Figure 39, but the additional ones will not affect the solution of the current task.
How to Use Triggers and Events

Figure 40 shows the resources that will be generated from your policy objects.

The AOR can run on different systems in the sysplex. This means you create one policy object of type APL to represent the AOR, but SA z/OS generates three resources from this policy object (one resource per system in the sysplex).

**Background:**

SA z/OS makes a distinction between applications (policy objects of type APL) and application resources.

You define applications in the customization dialog, SA z/OS generates application resources from the applications that you define.

You can use the application CICS1AR1 as the AOR, and CICS1TOR for the TOR. You defined both of these in “How to Work with Resources” on page 33.

Resources will only be generated from a policy object if it is linked to a system or sysplex. For entry type APL, the customization dialog does not allow you to create a direct link. What you need to do is link both the application and the system or sysplex to the same application group. In the plan shown in Figure 39 on page 61, this is done as follows:

- Application group AOR_MOVE provides indirect linking between application CICS1AR1 and sysplex SYSPLEX1
- Application group TOR_GRP provides indirect linking between application CICS1TOR and system SYSTEM3.

To have the CICS applications available when required, you need to define a service period (a policy object of type SVP). This is CICS_SVP in Figure 39 on page 61. There are two ways that you can use the service period:

1. Link the service period to both application groups AOR_MOVE and TOR_GRP.
2. Put AOR_MOVE and TOR_GRP together in another application group and link the service period to it.
Implementing the Solution

1. On the Customization Dialog Primary Menu panel, open the policy database that you created in the scenario described in “How to Work with Resources” on page 33.

2. Create a new application group called AOR_MOVE with these values:
   - Application Group Type: SYSPLEX
   - Nature: MOVE
   - Automation Name: AOR_MOVE

3. Use the WHERE USED policy item to link the application group AOR_MOVE to SYSPLEX1.
   SA z/OS generates the APG resource: AOR_MOVE/APG

4. Link application CICS1AR1 to application group AOR_MOVE.
   SA z/OS generates the following APL resources:
   CICS1AR1/APL/SYSTEM1
   CICS1AR1/APL/SYSTEM2
   CICS1AR1/APL/SYSTEM3

5. In application group AOR_MOVE, select policy item RESOURCES and set the following preferences for the new APL resources:
   - 900 for CICS1AR1/APL/SYSTEM1, so that starting AOR_MOVE/APG starts CICS1AR1/APL/SYSTEM1
   - 600 for CICS1AR1/APL/SYSTEM2, so that it starts if CICS1AR1/APL/SYSTEM1 fails
   - 500 for CICS1AR1/APL/SYSTEM3, so that it starts if CICS1AR1/APL/SYSTEM2 also fails

6. Create a new application group called TOR_GRP with these values:
   - Application Group Type: SYSTEM
   - Nature: BASIC
   - Automation Name: TOR_GRP

7. Use the WHERE USED policy item to link the application group TOR_GRP to system SYSTEM3.
   SA z/OS generates the APG resource: TOR_GRP/APG/SYSTEM3

8. Link application CICS1TOR to application group TOR_GRP.
   SA z/OS generates the APL resource: CICS1TOR/APL/SYSTEM3

9. Define a HASPARENT relationship between CICS1TOR/APL/SYSTEM3 as the child (dependent resource) and AOR_MOVE/APG as the parent (supporting resource) as follows:
   a. For application CICS1TOR, select policy item RELATIONSHIPS.
b. On the Relationship Selection List panel, enter new hasparent at the command line.

c. On the Define Relationship panel, enter resource AOR_MOVE/APG in the Supporting Resource field.

10. Create a new application group called CIC$_APL with these values:
   - Application Group Type: SYSPLEX
   - Nature: BASIC
   - Automation Name: CIC$_APL

11. Use the WHERE USED policy item to link application group CIC$_APL to group SYSPLEX1.

   Result: SA z/OS generates the APG resource: CIC$_APL/APG

12. Make TOR_GRP/APG/SYSTEM3 and AOR_MOVE/APG members of CIC$_APL/APG as follows:
   a. In application group CIC$_APL, select policy item RESOURCES.
   b. Overtype NO in the Show all Resources field with YES and press Enter.
      This displays the resources that have been generated so far.
   c. Use action s to select the following resources:
      - AOR_MOVE/APG
      - TOR_GRP/APG/SYSTEM3

13. Create a new event called BCKUPFIN.
    This event is set whenever the backup process that is required is completed.

    Note:
    SA z/OS can automatically set this event by trapping message IEF404I from the backup process. Use the INGEVENT command to set events.

14. Select policy item CONDITION and, on the Unset Condition panel that appears, enter the option that suits your needs in the Unset Condition field, for example UP.
Background:
The condition for unsetting an event will be checked on the observed status. There are eleven possible values for the observed status: six that indicate availability and five that indicate unavailability (see [IBM Tivoli System Automation for z/OS User's Guide](https://www.ibm.com) for details). Among those values that indicate availability, there is one named STARTING. The “UP” condition, for example, means: the observed status has a value that indicates availability, but is not the value STARTING.

The observed status belongs to generated resources, not to policy objects. In other words: a policy object does not have an observed status. Only the resources that are generated from the policy object have an observed status. So there is an observed status for each of the boxes in Figure 40 on page 62, but not for the boxes in Figure 39 on page 61.

As a result of all this, the condition for unsetting an event only has an effect when the event is related to a generated resource because only then is there an observed status that the condition can be checked on.

The only way that an event can be related to generated resources is via a trigger, as follows: when the event is linked to a trigger and the trigger is linked to an APL, APG, or MTR policy object, the event will be indirectly related to every resource that is generated from that APL, APG, or MTR policy object.

The key to understanding the effect of the unset condition is that SA z/OS maintains a separate event value for each generated resource that an event is indirectly related to. So an event may be set for one generated resource, but unset for another at the same time.

The effect of UP, for example, is that the event will be unset for a particular generated resource as soon as the observed status of that particular generated resource is given a value that indicates availability, but is different from the value STARTING.

15. Create a new trigger called CHCKBCKP, then enter trigger details as follows to make the trigger check backup completion before startups:
   a. On the Policy Selection panel for the new trigger, select policy item CONDITION.
   b. On the Trigger Conditions panel, select condition STARTUP 1.
   c. On the STARTUP Condition for Trigger panel that follows, use action s to change the status of event BCKUPFIN to SELECTED.

16. Link trigger CHCKBCKP to application CICS1AR1 as follows:
   a. Select policy item WHERE USED for trigger CHCKBCKP.
   b. On the Where Used panel, select the application CICS1AR1.

   As a result of linking trigger CHCKBCKP to both application CICS1AR1 and event BCKUPFIN, SA z/OS maintains a separate BCKUPFIN value for each of the three resources that were generated from application CICS1AR1 (that is, CICS1AR1/APL/SYSTEM1, CICS1AR1/APL/SYSTEM2, and CICS1AR1/APL/SYSTEM3). When you use INGEVENT to set the event, you can supply the name of one of these resources so that the event is only set for that particular resource, but remains unset for the others. If you do not supply
a resource parameter with INGEVENT, the event will be set for all three resources. They will, however, still be independent of each other, so they might be unset at different times.

17. Create a service period called CICS_SVP, then enter schedule details as follows:
   a. On the Policy Selection panel for the new service period, select policy item SERVICE WINDOW.
   b. On the Service Windows panel, scroll down to SATURDAY and WEEKDAY and fill in the following values:

<table>
<thead>
<tr>
<th>Day</th>
<th>U</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sat</td>
<td>08:00</td>
<td>14:00</td>
</tr>
<tr>
<td>Wed</td>
<td>07:00</td>
<td>18:00</td>
</tr>
</tbody>
</table>

18. Link service period CICS_SVP to application group CICS_APL as follows:
   a. For service period CICS_SVP, select policy item WHERE USED.
   b. On the Where Used panel, select the application group CICS_APL.

How the Solution Works
The key to understanding how this works is the distinction in SA z/OS between starting a resource and setting a resource's desired status to AVAILABLE (see IBM Tivoli System Automation for z/OS User’s Guide for more information on the concept of desired status).

A desired status of AVAILABLE is a prerequisite for starting a resource, but it may not be enough. Further startup prerequisites may be attached to individual resources, for example, conditions that are implicit in a HASPARENT relationship.

You also need to be aware of what it means to start a resource. For an APL resource, this is obvious: it means that a start order is sent to the appropriate automation agent. For an APG resource, however, it means something different: the start request is propagated to some or all (depending on the APG resource’s nature) of its member resources. Provided that the propagated start request wins over any other request that might be simultaneously sent to the same member resource, its desired status is set to AVAILABLE.

The resources in our scenario and the special prerequisites attached to them can be seen in Figure 40 on page 62. This is what happens:
1. Whenever the time of day enters the time window defined for availability in service period CICS_SVP, SA z/OS generates a start request to resource CICS_APL/APG.
2. If this start request wins over any concurrent requests to CICS_APL/APG from elsewhere, SA z/OS sets the desired status of CICS_APL/APG to AVAILABLE.
3. Resource CICS_APL/APG has no further start prerequisites attached to it, so it is started immediately.
4. CICS_APL/APG has a nature of BASIC, so starting CICS_APL/APG means that the start request is propagated to every resource that is a member of it. In this scenario, these are AOR_MOVE/APG and TOR_GRP/APG.
5. If the start request that is propagated to AOR_MOVE/APG and TOR_GRP/APG wins, SA z/OS sets the desired status of both application groups to AVAILABLE.
6. Neither AOR_MOVE/APG nor TOR_GRP/APG has any further start prerequisites attached to it, so they are started immediately.

7. Starting TOR_GRP/APG means that the start request is propagated to its only member, CICS1TOR/APL/SYSTEM3.

8. If the start request that is propagated to CICS1TOR/APL/SYSTEM3 wins, SA z/OS sets its desired status to AVAILABLE.

9. Because of the HASPARENT relationship that is attached to CICS1TOR/APL/SYSTEM3, a desired status of AVAILABLE is not yet enough to start CICS1TOR/APL/SYSTEM3. So nothing happens as long as its supporting resource, AOR_MOVE/APG, has not started up.

10. AOR_MOVE/APG is a move group, so starting AOR_MOVE/APG means that the start request is only propagated to one of its member resources. This is the one with the highest preference (referred to as its primary member). In this scenario, it is CICS1AR1/APL/SYSTEM1.

11. If the start request that is propagated to CICS1AR1/APL/SYSTEM1 wins, SA z/OS sets its desired status to AVAILABLE.

12. Because of the trigger, a desired status of AVAILABLE is not yet enough to start CICS1AR1/APL/SYSTEM1. So nothing happens before its backup event is set. As soon as the event is set for resource CICS1AR1/APL/SYSTEM1, it starts up.

13. Starting CICS1AR1/APL/SYSTEM1 means that the automation manager sends a start order for CICS1AR1/APL/SYSTEM1 to the automation agent on SYSTEM1.

14. When CICS1AR1/APL/SYSTEM1 is fully up and running, the automation agent on SYSTEM1 reports the fact to the automation manager, which sets the observed status of CICS1AR1/APL/SYSTEM1 to an observed status value that indicates availability, but is different from STARTING.

15. The new observed status passes up from CICS1AR1/APL/SYSTEM1 to AOR_MOVE/APG. So AOR_MOVE/APG, too, now has an observed status value that indicates availability, but is different from STARTING.

16. This supplies the second startup prerequisite for CICS1TOR/APL/SYSTEM3 that is implicit in its HASPARENT relationship, so CICS1TOR/APL/SYSTEM3 is now started, too.

17. Starting CICS1TOR/APL/SYSTEM3 means that the automation manager sends a start order for CICS1TOR/APL/SYSTEM3 to the automation agent on SYSTEM3.

18. When CICS1TOR/APL/SYSTEM3 is fully up and running, the automation agent on SYSTEM3 report the fact to the automation manager, which sets the observed status of CICS1TOR/APL/SYSTEM3 to an observed status value that indicates availability, but is different from STARTING.

19. For the reasons given in step 14 on page 64 and step 16 on page 65, the observed status of CICS1TOR/APL/SYSTEM3 is checked to determine when the BCKUPFIN event for CICS1TOR/APL/SYSTEM3 will be unset.

How to Use User E-T Pairs

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

In this example, the data for a job card is modified.
How to Use User E-T Pairs

Step 1: Specifying User Data in the Customization Dialog

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

If you select the E-T DATA policy item from the Policy Selection panel for the JOBCTIONS User E-T Pairs, the UET Entry-Type Selection panel similar to Figure 41 is displayed.

![Figure 41. UET Entry-Type Selection Panel](image)

Enter S for Select in the Action column for the User E-T Pairs pair that you want to process. In this example, you want to process the JOBCARD of type DEFAULTS.

SA z/OS displays the UET Keyword-Data Specification panel, where you can enter the keywords and user data for each Entry-Type pair data, as shown in Figure 42. For more details about this panel, see "E-T DATA Policy Item" on page 295.

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.

![Figure 42. UET Keyword-Data Specification Panel](image)

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 script as shown in the following example:
How to Use User E-T Pairs

```
Get_keyword: /* name of a subroutine in the REXX exec */
    Arg keyword.
    len = Length(keyword)
    /* get the data within a pipe */
    PIPE NetView ACFFQRY JOBCARD DEFAULTS | STEM ALL_DATA.
        | SEPARATE | LOCATE i.'||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
JOBCARD 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 section describes how user-defined data is used to automatically execute commands if an application issues a certain message.

Consider the following scenario:

The HSM application issues the ARC0909E message that indicates the utilization percentage of the journal data set. You want to automate recovery of the HSM journal data set if it is more than 97% full. You therefore want SA z/OS to automatically issue two commands to HSM that will perform the recovery:

- **HOLD ALL**: To stop all further activities for HSM.
- **BACKVOL CDS**: To create a backup of the full journal data set and allocate a new one.

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

**Step 1: Specify the User-Defined Data**

Select the application that you want to automate the recovery for and then select its MESSAGES/USER DATA policy item. This leads you to the Message Processing panel where you enter the message ID ARC0909E as shown in Figure 43 on page 70.
How to Implement Message Processing with User-Defined Data

To specify your data, type U or USR in the Action column of the Message Processing panel shown in Figure 43. Pressing Enter leads to the User Data Processing panel where you specify your input as shown in Figure 44.

In the REXX script that you will write in “Step 4: Write the REXX Script” on page 73, you will compare the percentage value returned from the ARC0909E message with the value specified here, and, if it exceeds your specified value, you will cause SA z/OS to issue the commands specified in “Step 3: Update the NetView Automation Table” on page 71. If you later decide to start recovery if 95% usage is exceeded, you need not edit your REXX script, but only your customization dialog input and then rebuild your system operations configuration files.

Press PF3 to store your input and return to the Message Processing panel.

Step 2: Specify the Commands

On the Message Processing panel (Figure 43) enter C or CMD in the Cmd column for message ARC0909E. This leads to the Command Processing panel where you
How to Implement Message Processing with User-Defined Data

specify your input as shown in Figure 45.

Figure 45. Scenario: Command Processing Panel

Step 3: Update the NetView Automation Table

You need to specify a NetView automation table (AT) entry to trigger a REXX script called YOUR_CLIST, which you will create in "Step 4: Write the REXX Script" on page 73.

You can specify the AT entry with the Automation Processing panel to override the predefined AT entry. To do this, enter A or AT in the Action column for message ARC0909E in the Message Processing panel, as shown in Figure 46.

Figure 46. Scenario: Message Processing Panel for AT Entry

This leads to the Message Automation Overview panel, as shown in Figure 47 on page 72. This shows the default values for the message.
You need to change the action so enter AO at the command line to open the AT entry in an editor and update the entry, as shown in Figure 48.

Saving the entry and pressing PF3 returns you to the Message Automation Overview panel, where you can see that the entry has been updated with your override, as shown in Figure 49.

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

IF MSGID = 'ARC0909E'
THEN EXEC('YOUR_CLIST') ROUTE(ONE *) ;
Step 4: Write the REXX Script

The value specified on the User Data Processing panel (Figure 44 on page 70) is accessed in the PIPE statement using the ACFFQRY command and stored in the variable limit.

To issue the commands specified on the Command Processing panel (Figure 45 on page 71), the ACFCMD command is used.

```rexx
/* REXX */
/* debugging ... */
......
lrc = 0
/* Get the message from the safe */
"GETMLINE txt 1"
svmsgid = msgid()
svjobname = jobname()
Parse Var txt . type . ' IS ABOUT ' usage '% ' .
/* This sample recovers just type JRNL */
If type \= 'JRNL' Then Exit 0
/* Check the RECOVERY automation flag */
"AOCQRY "svjobname"."svmsgid" RECOVERY"
lrc = rc
Select
/* Automation allowed */
When lrc = 0 Then Do
   /* Get TGlobal variable from AOCQRY */
   "GLOBALV GETT SUBSAPPL"
   /* Get the user-data from the ACF */
   "PIPE (STAGESEP | NAME GETDATA)" ,
   "NETV ACFFQRY "subappl"."svmsgid" ,
   "SEPARATE"
   "LOCATE 1.6 /LIMIT=" ,
   "EDIT 7.2 1" ,
   "VAR limit"
   /* If no limit was defined use default */
   If symbol('limit') = 'LIT' Then limit = '95'
   /* Check if limit is exceeded. If so, issue commands */
   If usage >= limit Then Do
      "ACFCMD MSGTYP="svmsgid
      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
      ...
```

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

End
End
Exit lrc
Chapter 4. Entry Type Reference

This chapter describes how to define SA z/OS entry types. They are described in the following sections in the order they appear on the Entry Type Selection panel shown in Figure 50.

This chapter discusses the following entry types:

- "Enterprise Entry Type" on page 77
- "Group Entry Type" on page 83
- "SubGroup Entry Type" on page 100
- "System Entry Type" on page 101
- "ApplicationGroup Entry Type" on page 116
- "Application Entry Type" on page 140
- "Events Entry Type" on page 212
- "Service Periods Entry Type" on page 215
- "Triggers Entry Type" on page 218
- "Processor Entry Type" on page 220
- "Monitor Resources Entry Type" on page 230
- "zEnterprise Ensembles Entry Type" on page 241
- "Messages Entry Type" on page 255
- "Timers Entry Type" on page 257
- "Tape Attendance Entry Type" on page 261
- "MVS Component Entry Type" on page 263
- "MVSCOMP Defaults Entry Type" on page 267
- "System Defaults Entry Type" on page 268
- "Application Defaults Entry Type" on page 271
- "Automation Operators Entry Type" on page 272
- "Notify Operators Entry Type" on page 276
- "Network Entry Type" on page 279
- "Sysplex Defaults Entry Type" on page 288
- "Resident CLISTS Entry Type" on page 290
- "Status Display Entry Type" on page 291
- "User E-T Pairs Entry Type" on page 294
Product Automation (PRD) from the Entry Type Selection panel with the following Entry Types:
- “OPC System Detail Entry Type” on page 244
- “OPC Controller Details Entry Type” on page 245
- “OPC Special Resources Entry Type” on page 247
- “OPC Workstation Domains Entry Type” on page 248
- “CICS CONNECTION Policy Item” on page 250
- “CCN MONITOR PERIOD Entry Type” on page 251

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 51 where you can select the product that you want to define automation policy for. You can select from the following products:
- TWS, see “Defining Automation for TWS Components”
- CICS, see “Defining Automation for CICS Components” on page 77

For TWS (OPC) components refer to “Defining Automation for TWS Components,” and for CICS components, refer to “Defining Automation for CICS Components” on page 77 for the specific product entry types:

---

**MENU HELP**

---

AOFGEPOM  Entry type selection for Product Automation
Option ===>

OPC components
30 OEN  OPC System Details
31 OCS  Controller Details
32 OSR  Special Resources
33 ODM  Workstation DomainID

CICS components
40 CCN  CICS Connections (SA 3.2 or below only)
41 CVP  CCN Monitor Periods (SA 3.2 or below only)

---

**Figure 51. Entry Type Selections for Product Automation Panel**

For TWS (OPC) components refer to “Defining Automation for TWS Components,” and for CICS components, refer to “Defining Automation for CICS Components” on page 77 for the specific product entry types:

---

**Defining Automation for TWS Components**

For TWS 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 brings 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. This is described in more detail in “OPC System Detail Entry Type” on page 244.

**(OCS) Controller details**

With this option, you can link the TWS controller subsystem name with the NetView domain where it is located. 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. This is described in more detail in “OPC Controller Details Entry Type” on page 245.
Defining Automation for CICS Components

It is recommended that you use event-based monitoring for your CICS link and health monitoring rather than the following policy items. Event-based CICS link and health monitoring is implemented using CICSPlex® System Manager (CICSPlex SM or CPSM) objects.

Whenever an event is received from CICSPlex SM, message ING150I is issued. This acts as a trigger that allows you to set the health status of a Monitor Resource based on the monitored object name and job. You can define Monitor Resources for CPSM objects (for example, connections), and codes for message ID ING150I in the MESSAGE/USER DATA policy of the Monitor Resource to map the CPSM severities to valid health states. For more details see the section “Event-Based Monitoring Using Monitor Resources” in IBM Tivoli System Automation for z/OS Customizing and Programming.

Note: The Entry Types for CICS components can only be used to maintain systems that are running an earlier version of SA z/OS (3.2 or below).

For CICS components, the following entry types are available:

(CCN) Link definitions
Use this option to define the interregion and intersystem communication (IRC and ISC) links so that they can be monitored by SA z/OS. This is described in more detail in CICS Connections Entry Type on page 249.

(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 the (CCN) Link definition option. This is described in more detail in CCN MONITOR PERIOD Entry Type on page 251.

Enterprise Entry Type

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 52 on page 78 shows the policy items that are displayed when Enterprise is selected from the Entry Type Selection panel.
The policy items for Enterprises are:

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

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

For more information, see "SEND COMMAND OPERS Policy Item" on page 79.

**INGSEND PARMS**
This policy item allows you to define default parameters for the INGSEND command.

For more information see "INGSEND PARMS Policy Item" on page 80.

**PROCOPS FOCAL POINTS**
This policy item allows you to define focal point domain IDs. For more information, see "PROCOPS FOCAL POINTS Policy Item" on page 81.

**AUTO MSG CLASSES**
This policy item allows you to define your own message classes. Message classes that are defined here can be assigned to operators by using policy NOTIFY OPERATORS of the entry type NFY. For more information, see "AUTO MSG CLASSES Policy Item" on page 82.

**SYMBOL DESCRIPTION**
This policy item allows you to enter system automation symbol descriptions. For more information, see "SYMBOL DESCRIPTION Policy Item" on page 83.

**DOCUMENTATION**
This policy item allows you to enter or update any text with the ISPF editor. You can use it to:

- Enter overall policy database descriptions
- View documentation that is shipped together with predefined SA z/OS sample add-on policies
The policy database is locked for updates by other users while you are editing data in this policy item.

SEND COMMAND OPERS Policy Item

Figure 53 shows the panel that is displayed when the SEND COMMAND OPERS policy item is selected.

This policy item allows you to define default RMTCMD autotasks for target domains, so that operators 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 issue a command to. Use the Operator Profile for Sending Commands panel to define an association between an origin operator and a target RMTCMD autotask. The INGSEND 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 INGSEND 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 INGSEND, see IBM Tivoli System Automation for z/OS User’s Guide and IBM Tivoli System Automation for z/OS Operator’s Commands. For more information on RMTCMD, see Tivoli NetView for z/OS Command Reference Volume 1.
The definitions you enter here apply to all systems in the enterprise. They are included in the automation control file built for each system. At run time, INGSEND on each origin domain will use only the information intended for that particular origin Netid.Domin.

On the Operator Profile for Sending Commands panel (Figure 53 on page 79), 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. This is the NetView default.

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.

**INGSEND PARMS Policy Item**
If you select policy item INGSEND PARMS from the Policy Selection panel for enterprises, a panel similar to Figure 54 on page 81 is shown.
The System Operations INGSEND command is used to route commands across domains. It exploits the NetView® RMTCMD command.

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

**TARGET NETID**
The Target Network ID is optional and, if entered, must be eight characters or less. The first character must be alphanumeric. The rest must be either alphanumeric or national characters. If left blank, it defaults to the wildcard value of "*".

**TARGET DOMAIN**
The Target Domain is optional and, if entered, must be five characters or less. The first character must be alphanumeric. The rest must be either alphanumeric or national characters. If left blank, it defaults to the wildcard value of "*".

Wildcard value of "*" is allowed.

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

**PROCOPS FOCAL POINTS Policy Item**
A panel, as shown in Figure 55 on page 82, is displayed if you select policy item PROCOPS FOCAL POINTS from the Policy Selection panel for enterprises.
PROCOPS FOCAL POINTS Policy Item for Enterprises

The panel contains the following fields:

**Primary/Backup focal point name**
- Enter the domain name of the primary and backup NetView that is hosting processor operations. The customization dialog does not verify your entry.
- This name is shown in the processor operations ISQXDST status display command.

**AUTO MSG CLASSES Policy Item**

A panel, as shown in Figure 56, is displayed if you select policy item AUTO MSG CLASSES on the Policy Selection panel for enterprises.

This panel defines the user message classes that are available to be assigned to notify operators by using policy NOTIFY OPERATORS of entry type NFY. The Auto Msg Classes entry type allows you to define your own message classes.

You can also enter a short description of each message class. Message classes must consist of an alpha and a numeric character. All double numeric message classes are reserved for use by IBM. See “Notify Operators Entry Type” on page 276 for more information on the message classes that SA z/OS defines.
AUTO MSG CLASSES Policy Item for Enterprises

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

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

SYMBOL DESCRIPTION Policy Item

If you select policy item SYMBOL DESCRIPTION from the Policy Selection panel for enterprises, a panel similar to Figure 57 is shown.

Figure 57. System Automation Symbols (AOCCLONEx) Description Panel

This panel allows you to enter the description of the general use of each system automation symbol.

AOCCLONEx
This allows you to describe the general use of each system automation symbol. This description is shown in the AUTOMATION SYMBOLS policy of all systems (entry type SYS) in the policy database.

Group Entry Type

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

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

Creating a New Group

How to create new policy objects in general, is described in "How to Create New Policy Objects" on page 19.

Groups can be defined as either standard groups or sysplex groups. A standard group contains any combination of systems and subgroups. A sysplex group contains only systems or subgroups that belong to the same XCF group. For a sysplex group, there are additional policy items to describe certain sysplex-specific resources. A sysplex group may also have Sysplex Application Groups connected to it. Sysplex Application Groups are groups of applications that can be located 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 58 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 IBM Tivoli System Automation for z/OS Planning and Installation).

The unique fields are:

**Entry name**

This is the name of the entry as used by the SA z/OS dialogs. Each group must be assigned a unique, non-blank name. The name must start with an alphabetic character. The name must be unique among the names of all zEnterprise ensembles, groups, subgroups, systems (including both the entry names and the system/image names), and processors (including both the entry names and processor operations Target HW names) in the current policy database.
Creating a New Group

**Note:**
This name can be used in the TARGET parameter of SA z/OS commands. If the specified name is longer than 8 characters, truncation of the name may occur on certain SA z/OS fullscreen command panels.

When running in local mode (sysplex() function returns LOCAL) you must define a Group name if you want to use SA z/OS commands that target other systems via the TARGET parameter.

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

The available values are:

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

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

**ProcOps Commands**
Indicates whether processor operations commands issued against this group are to be accepted for processing. The default is NO. See [IBM Tivoli System Automation for z/OS Operator's Commands](#) for information about the processor operations commands.

**Note:**
If you specify YES, 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 59 on page 86 shows the policy items that are displayed for a standard group.
Policy Items for Standard Groups

The unique policy items for groups are:

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

For more information, see "GROUP INFO Policy Item" on page 88.

SUBGROUPS
This policy item displays a list of all Subgroups indicating which subgroups are selected for this group.

For more information, see "SubGroup Entry Type" on page 100.

SYSTEMS
This policy item displays a list of all Systems, indicating which systems are selected for this group.

For more information, see "System Entry Type" on page 101.

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 60 on page 87 is an example of the Policy Selection panel for Sysplex groups.
The policy items for sysplex groups are the same as those for standard groups with the following additions:

**SYSPLEX**
This policy item allows you to describe certain resources associated with the sysplex. For more information, see “SYSPLEX Policy Item” on page 89.

**NMC DEFINITIONS**
This policy item allows you to make NMC definitions for the primary and secondary NMC Focal Point™ Domains. For more information, see “NMC DEFINITIONS Policy Item” on page 91.

**APPLICATION GROUPS**
This policy item displays a list of all Sysplex Application Groups, indicating which application groups are selected for this sysplex. For more information, see “ApplicationGroup Entry Type” on page 116.

**SYSPLEX DEFAULTS**
This policy item displays a list of all Sysplex Defaults, indicating which sysplex defaults are selected for this sysplex. For more information, see “Sysplex Defaults Entry Type” on page 288.

**LOCAL PAGE DATA SET**
With this policy item you can predefine local page data sets that are used in recovery situations to prevent auxiliary storage shortage outages. For more information, see “LOCAL PAGE DATA SET Policy Item” on page 92.

**JOB DEFINITIONS**
With this policy item you can define the jobs that can or cannot be canceled (KEEP/CANCEL), in case the job is responsible for the shortage condition. For more information, see “JOB DEFINITIONS Policy Item” on page 93.
Policy Items for Sysplex Groups

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

For more information, see "JOB/ASID DEFINITIONS Policy Item" on page 94.

**COMMAND DEFINITIONS**
With this policy item you can define which commands you want to monitor and define a maximum time that a command can be blocked by a job or another command before recovery action takes place.

You can also specify an action (KEEP, DUMP or NODUMP) to be taken against this command if it is found to be blocking other commands from executing.

For more information, see "COMMAND DEFINITIONS Policy Item" on page 95.

**COMMAND FLOODING**
With this policy item you can define individually the interval time for snapshot processing for each command class. When message IEE806A triggers the recovery, the first snapshot is taken after the interval has expired.

For more information, see "COMMAND FLOODING Policy Item" on page 97.

**RESOURCE DEFINITIONS**
With this policy item you can define DUMP and other miscellaneous recovery options, including the message delay time for message ING920I (issued when option KEEP is specified), and IEADMCxx symbols.

For more information, see "RESOURCE DEFINITIONS Policy Item" on page 97.

**RECOVERY OPTIONS**
With this policy item you can define DUMP and other miscellaneous recovery options including message delay time for message ING920I (issued when option KEEP is specified), and IEADMCxx symbols.

For more information, see "RECOVERY OPTIONS Policy Item" on page 98.

**GROUP INFO Policy Item**
A panel, as shown in Figure 61 on page 89 is displayed if you select policy item GROUP INFO on the Policy Selection panel for groups.
GROUP INFO Policy Item for Groups

This panel displays the following information:

Group Type

This field cannot be changed; it shows the value SYSPLEX or STANDARD as it was specified when this group was created. For more details, see "Group Type" on page 85.

ProcOps Commands

For more details, see "ProcOps Commands" on page 85.

SYSPLEX Policy Item

A panel, as shown in Figure 62, is displayed if you select policy item SYSPLEX on the Policy Selection panel for sysplexes.

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

Sysplex Name

Defines the name of the sysplex.
SYSPLEX Policy Item for Groups

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

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

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

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

Couple Data Set HLQ
Specify the high level qualifier to be used for the allocation of alternate couple data sets (CDS). The qualifier can consist of up to three parts according to the z/OS data set naming rules. When allocating an alternate CDS, this qualifier is appended with the type (for example, SYSPLEX) and 5vvvvvvv, where vvvvvvvv is the volume name from the list of Alternate Volumes.

CDS type Alternate volumes
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 Alternate volumes column you can enter up to 8 volume names for allocating alternate couple data sets for types Sysplex, ARM, CFRM, LOGR and SFM (you cannot specify any alternate volumes for WLM). The names must be in accordance with the z/OS volume naming rules. When allocating alternate couple data sets, the allocation starts at the first specified volume and continues in the specified order.

Desired monitoring
The information you enter in this column determines when the SA z/OS NMC workstation 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 that 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, the icon for QUAL1.ARM.CDS0 is marked as being unsatisfactory (colored red).
You specify the monitoring that is to be done for each type of couple data set as follows:

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

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

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

**NMC DEFINITIONS Policy Item**
A panel as shown in Figure 63 is displayed if you select policy item NMC DEFINITIONS on the Policy Selection panel for sysplexes.

<table>
<thead>
<tr>
<th>Entry Type</th>
<th>PolicyDB Name</th>
<th>Entry Name</th>
<th>Enterprise Name</th>
<th>NMC Focal Points</th>
<th>Anchor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td>USER_PDB</td>
<td>SYSPLEX_GROUP_01</td>
<td>USER_ENTERPRISE</td>
<td>DOMS1,DOMS2</td>
<td>WTOR</td>
<td>NMC anchor for WTORs</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TAPE</td>
<td>NMC anchor for tape mount requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SYSPLEX</td>
<td>NMC anchor for sysplex related events</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CRITMSG</td>
<td>NMC anchor for critical messages</td>
</tr>
</tbody>
</table>

*Figure 63. NMC Definitions Panel for Sysplex Groups*

The NMC Definitions panel displays the following information:

**NMC Focal Points**
Defines a list of NetView domain IDs that act as focal points. The domain IDs should be in the form `netid.domain` or `domain` and be separated by commas. You can include system symbols and system automation symbols.

**Anchor**
Allows you to specify anchors that are special objects in RODM and displayed on the topology client of the SA z/OS Graphic Interface. They are used to display and group minor resources that are derived from events such as WTORs and tape mount requests.
Anchors are special objects that exist in RODM. They are displayed as icons on the NMC topology client. They are not automated by the automation manager. They play the role of major resources and serve these purposes:

- Anchors are used to display and group minor resources that are derived from events such as WTORs and tape mount requests.
- Lost&Found anchors are used to display minor resources that either the specified major resource or sysplex anchor does not exist for in RODM.
- Couple data sets (CDSs), coupling facilities (CFs) and external timer references (ETRs) are displayed as minor resources and are attached to the sysplex anchor.
- It is possible to attach a minor resource to a user-defined anchor with the INGPOST command. The major resource specified to INGPOST can be a regular resource (for example, APL, APG, SYS, etc.) or an anchor.
- Heartbeats (minor resources) are attached to a heartbeat anchor.

Specify the name of an anchor on this panel for these cases:

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

Do not specify an anchor on this panel for:

- Heartbeats. The anchor is automatically created by SA z/OS.
- Lost&Found anchors.

**Description**

Allows you to specify a description for the anchor. The value is only shown in the SA z/OS customization dialog panel. It is not displayed in the SA z/OS Graphic Interface.

**LOCAL PAGE DATA SET Policy Item**

With the automation of local page data sets, SA z/OS prevents auxiliary storage shortage outages, by dynamically allocating spare local page data sets when needed. When you select the LOCAL PAGE DATA SET policy item the Local Page Data Set Recovery panel is displayed, as shown in [Figure 64 on page 93]().
Fill in the following fields:

**Page Data Set HLQ**

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

**Number of Cylinders**

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

**Page Data Set Volumes**

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

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

Defines the pre-formatted spare page data set to be used in the recovery situation. Specify the names without quotes. You can include system symbols and system automation symbols (see "Assigning System Automation Symbols (AOCCLONE)" on page 8). The data sets must be allocated on a volume shared by all systems in the sysplex and must be catalogued in the master catalog. In case the systems in the sysplex do not share the master catalog, the page data set must be re-catalogued on each system except the one which created the data set using the following IDCAMS statement:

```idcams
DEF PAGESPACE(NAME(lpdsn) VOLUME(volume) RECATALOG)
```

**JOB DEFINITIONS Policy Item**

You can use the JOB DEFINITIONS policy item to customize the automation to cancel or keep jobs that are blocking a resource. When you select the JOB DEFINITIONS policy item the Local Page Data Set Recovery Job Definition panel
JOB DEFINITIONS Policy Item for Groups

is displayed, as shown in Figure 65.

Figure 65. Local Page Data Set Recovery Job Definition Panel

Fill in the following fields:

Job Name/*
Defines the jobs that can or cannot be canceled if they are one of the jobs that are responsible for the shortage condition.

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

JOB/ASID DEFINITIONS Policy Item

Entries specified in the JOB/ASID DEFINITIONS policy item apply to long-running ENQ, hung-command and command-flooding recovery. Specify the names of Jobs or ASIDs that require special treatment (KEEP, DUMP or NODUMP) when a TCB in the Job or ASID is found to be causing a long-running ENQ lockout, a hung-command situation or command flooding.

When you select the JOB/ASID DEFINITIONS policy item the Long Running ENQ Job/ASID Definitions panel is displayed, as shown in Figure 65.
Fill in the following fields:

**Job Name/ASID/**

Defines the jobs and address space IDs that can or cannot be canceled if they are responsible for a long running ENQ or hung command.

**Action**

Specify:

- **KEEP** if the job or ASID cannot be canceled. Message ING920I is issued at user-defined intervals as long as the ENQ or hung command situation persists (see “Message Delay field” on page 99).
- **NODUMP** if the job or ASID can be canceled without a dump.
- **DUMP** if the job or ASID can be canceled with a dump. DUMP options are specified with the RECOVERY OPTIONS policy item (see “RECOVERY OPTIONS Policy Item” on page 98).

You can specify a trailing wildcard (*).

For long-running ENQ recovery you must define a default entry with Job Name/ASID=* together with the action that will be appropriate for all jobs and ASIDs that are not explicitly defined. For hung-command recovery this default entry is ignored.

**COMMAND DEFINITIONS Policy Item**

With this policy item you can define which commands you want to monitor and define a maximum time that a command can be blocked by a job or another command before recovery action takes place.

You can also define an action (KEEP, DUMP or NODUMP) to be taken against this command if it is found to be blocking other commands from executing.

When you select the COMMAND DEFINITIONS policy item the "Hung" Command Definition panel is displayed, as shown in Figure 67 on page 96.
COMMAND DEFINITIONS Policy Item for Groups

Fill in the following fields:

Command/*
Specify the name of a z/OS system command that you want to check whether it is a long-running command (for example, the z/OS command SET).

Note:
The MODIFY command (F) cannot be processed by hung-command recovery because it does not enqueue on any system resource. There is no need to automate the MODIFY command.

Command Parameter(s)/*
Specify the parameter (or parameters) of a z/OS command that is to be checked whether it is a long running command (for example, PROG as the parameter of the z/OS command SET).

Time/EXCL
Defines the execution time that elapses before a command is treated as a long-running command. Valid value is 30-999 seconds.
If the command should be excluded from long-running command checking, specify EXCL.

Action
Defines the action to be taken (KEEP, DUMP or NODUMP) when the specified command blocks other commands from executing.

The action specified here will be overridden by the action that is specified in the JOB/ASID DEFINITIONS policy item for the job or ASID that issued the blocking command (often *MASTER* or CONSOLE).

For example, if a TSO user named FRED issued a D TRACE command that hung and blocked other commands from executing, and there is a Hung Command Definition of DISPLAY TRACE,30,NODUMP then, after 30 seconds the following would happen for different Job/ASID Definitions:
COMMAND DEFINITIONS Policy Item for Groups

<table>
<thead>
<tr>
<th>Job/ASID Definition</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRED,KEEP</td>
<td>FRED’s D TRACE command would not be canceled but message ING920I would be issued because specific Job/ASID definitions take precedence over COMMAND DEFINITIONs.</td>
</tr>
<tr>
<td></td>
<td>The D TRACE command would be canceled without a dump.</td>
</tr>
<tr>
<td>* ,KEEP</td>
<td>The D TRACE command would be canceled without a dump because COMMAND DEFINITIONs take precedence over default Job/ASID definitions.</td>
</tr>
</tbody>
</table>

COMMAND FLOODING Policy Item

You can define individually the interval time for snapshot processing for each command class. When message IEE806A triggers the recovery, the first snapshot is taken after the interval has expired.

When you select the COMMAND FLOODING policy item the Command Flooding Definition panel is displayed, as shown in Figure 68.

COMMANDS HELP

AOFGENQF Command Flooding Definition

Entry Type : Group
Entry Name : SYSPLEX_GROUP_01
PolicyDB Name : USER_PDB
Enterprise Name : USER_ENTERPRISE

Define the snapshot time (1-60 sec) after that the commands in this class are evaluated for command flooding.

Command class M1...
M2...
M3...
C1...
C2...
C3...

Figure 68. Command Flooding Definition Panel

Fill in the following field with the snapshot time for each command class:

Snapshot time

Defines the time the recovery routine should wait before it takes the next snapshot of the commands that are being executed or waiting in the command class. The time should be long enough to allow a task that issues a set of commands to finish before the first snapshot is taken. Valid value is 1-60 seconds. A default value of 3 seconds is used if nothing is specified.

RESOURCES DEFINITIONS Policy Item

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

When you select the RESOURCE DEFINITIONS policy item the Long Running ENQ Resource Definition panel is displayed, as shown in Figure 69 on page 98.
RESOURCE DEFINITIONS Policy Item for Groups

Fill in the following fields:

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

**Minor Resource/***
- Defines the minor resource name of the resource to be checked for long-running enqueues.

**Time/EXCL**
- Defines the execution time that elapses before an enqueue is treated as a long-running ENQ. A valid value is 30-999 seconds. If the resource should be excluded from long-running enqueue checking, specify EXCL.

### RECOVERY OPTIONS Policy Item

If you want to automate long-running ENQs, "hung" commands, or command flooding, you can specify dump options, a title for each dump being taken, and job names or address space IDs of the jobs that are kept or canceled.

Note that the dump options do not apply to command flooding because the recovery routine of the job being canceled can suppress the dump.

With this policy item you can define DUMP and other miscellaneous recovery options including message delay time for message ING920I (issued when option KEEP is specified), and IEADMCxx symbols.

When you select the RECOVERY OPTIONS policy item the Recovery Options panel is displayed, as shown in [Figure 70 on page 99](#).
Fill in the following fields:

**Message Delay**
Defines the number of minutes that automation waits before it reissues the message indicating that a job or a command is still blocking other jobs or commands but that the blocker must not be canceled.

**Note:**
Leaving the field blank results in a delay of 15 minutes.

**Dump Options**
Defines the DUMP option that is used for the SDATA parameter on the dump command. The following options are allowed:

- ALLNUC
- LPA
- NOSUM
- SERVERS
- TRT
- COUPLE
- LSQA
- NUC
- SQA
- WLM
- CSA
- NOPSA
- PSA
- SUM
- XESDATA
- GRSQ
- NOSQA
- RGN
- SWA

**Note:**
This value applies only to the jobs or address space IDs that are defined with DUMP in the following list.

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

**Job Name/ASID/***
Defines the jobs and address space IDs together with their symbols and substitution value for use with the dump command parmlib members IEADMCxx.

Specification of a trailing wildcard (*) is allowed.
RECOVERY OPTIONS Policy Item for Groups

Symbol and Value
Defines the symbol names and their substitution value in the format &symbol='value'.
Valid characters for the symbol are A – Z, 0 – 9, @, # and $. The value must be enclosed in single quotes and its length must not be longer than the symbol name.

SubGroup Entry Type
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 19.

Figure 71 is an example of the Define New Entry panel for subgroups.

![Figure 71. Define New Entry Panel for a Subgroup](image)

The unique field on this panel is:

ProcOps Commands
Indicates whether processor operations commands issued against this subgroup are to be accepted for processing. The default is NO.
Creating a New SubGroup

Note:
If a group with which this subgroup is associated is enabled for processor operations commands, 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 72 shows the policy items that are displayed when a subgroup is selected from the Entry Name Selection panel.

<table>
<thead>
<tr>
<th>ACTION</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFGEPOL</td>
<td>Policy Selection</td>
</tr>
<tr>
<td>Entry Type: Subgroup</td>
<td>PolicyDB Name: USER_PDB</td>
</tr>
<tr>
<td>Entry Name: TARGET_SUBGROUP</td>
<td>Enterprise Name: USER_ENTERPRISE</td>
</tr>
<tr>
<td>Action</td>
<td>Policy Name</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>Enter description</td>
</tr>
<tr>
<td>SUBGROUP INFO</td>
<td>Display subgroup information</td>
</tr>
<tr>
<td>SYSTEMS</td>
<td>Select systems for subgroup</td>
</tr>
<tr>
<td>WHERE USED</td>
<td>List groups linked to this entry</td>
</tr>
<tr>
<td>COPY</td>
<td>Copy data from an existing entry</td>
</tr>
</tbody>
</table>

Figure 72. 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 z/OS enterprise is displayed, indicating those systems that are related to this subgroup. You can add systems to the subgroup, or remove systems from the subgroup.

System Entry Type

A system is any host system in the SA z/OS enterprise. The operating systems can be either MVS or non-MVS. For system operations, the system must be MVS. For processor operations functions, the system can be MVS, VM, VSE, LINUX, or CF (coupling facility). A system definition includes:

- The operating system type
- Processor operations information
- Details about which IBM automation products are operating on the system
- Member application groups
- NetView and z/OS-related information
- Automation policy for the system (optional)

The information you supply here will determine what automation policy set objects are included in your system and how your system is automated.
System Entry Type

The SA z/OS customization dialog allows concurrent user access capability for the System entry type. See Appendix A, “Concurrent Customization by Multiple Users,” on page 357 for further information.

Creating a New System

How to create new policy objects in general, is described in “How to Create New Policy Objects” on page 19.

Figure 73 shows the panel that is displayed when you create a new system from the Entry Name Selection panel for systems.

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

---

COMMANDS HELP
-----------------------------------
AOFGDNK3 Define New Entry
Command ===> Define New Entry of type System

Entry name . . . . . . . . . . . . SYS1

Operating system . . . . MVS (MVS VM VSE LINUX CF)
Image/System name . . . SYS1

Specify information for NMC Focal Point Communication (MVS systems only):
Heartbeat Interval . . . . . 5 (1 - 60 minutes)
Missing Heartbeat Delay . . . 30 (1 - 3600 seconds)
Short description . . . . . . . .
Long description 1 . . . . . . .
Long description 2 . . . . . . .
Long description 3 . . . . . . .
Long description 4 . . . . . . .
Long description 5 . . . . . . .

Figure 73. Define New Entry Panel for a System

The fields that can be filled in are:

**Entry name**

This required name gives a unique name to the system that you are defining.

Valid characters are any alphanumeric character or any of the at sign (@), number sign (#), dollar sign ($), question mark (?), or underscore (_), with the first character being alphanumeric, or @, #, or $.

The entry name must be unique among the names of all zEnterprise ensembles, groups, subgroups, systems (including both the entry names and the system/image names) and processors (including both the entry names and the ProCops Target HW names) in the current policy database with the exception that the image/system name may match the entry name.

**Operating system**

Defines the operating system to be run on the system. Valid values are MVS, VM, VSE, LINUX, or CF.

CF indicates the coupling facility control code for the coupling facility in a Parallel Sysplex.
Creating a New System

LINUX identifies the Linux for zSeries operating system distributed for 390-CMOS and zSeries processors.

This field is required and its default value is MVS.

Image/System name
Identifies the system. For MVS systems, this value must match the system name (SYSID) as specified in the IEASYSxx member of SYS1.PARMLIB.

Valid characters are any alphanumeric character or any of @, #, or $. The name may start with a numeric character except for systems of type CF.

This name must be unique among the names of all zEnterprise ensembles, groups, subgroups, systems (including both the entry names and the system/image names) and processors (including both the entry names and the ProCops Target HW names) in the current policy database with the exception that the image/system name may match the entry name.

Note:
To support an alternate configuration of a system, special notation with a trailing plus sign (+) is allowed. This trailing character is removed when the automation files are built. If you are going to enable automation for IXC102A/IXC402D the Image/System name must not start with a numeric character. For more information see IBM Tivoli System Automation for z/OS Customizing and Programming.

Heartbeat Interval
This specifies the time interval at which the SA z/OS system operations functions sends a generic alert from this system to the NMC focal point system. The SA z/OS system operations functions on the NMC focal point system use heartbeat alerts to verify the status forwarding path from each sysplex or single remote system. For more information see "The SA z/OS Heartbeat Function" on page 104.

The heartbeat interval is a required field in the range 1 to 60 minutes that defaults to 5 minutes. Heartbeat information is relevant for z/OS systems only.

The smaller the heartbeat interval, the faster SA z/OS can detect a status forwarding path failure. For example, if you define an interval of two minutes then, on average, SA z/OS will take one minute, plus the missing heartbeat delay time, to detect a failure in the status forwarding path.

Missing Heartbeat Delay
This specifies the time that the SA z/OS system operations functions on the NMC focal point system will wait after a heartbeat was expected. If this time expires without receiving the heartbeat, the system operations functions will begin missing heartbeat processing for the remote system.

The field is provided as a way to buffer against temporary network delays. For more information see "The SA z/OS Heartbeat Function" on page 104.

The missing heartbeat delay is a required field in the range 1 to 3600 seconds that defaults to 30 seconds. Heartbeat information is relevant for z/OS systems only.

The smaller the missing heartbeat delay, the more sensitive SA z/OS is to temporary status forwarding path delays. The longer the delay, the slower SA z/OS will be to react to a missing heartbeat. Again, you will have to balance these two considerations.
Creating a New System

Consideration must be given to the number of systems and subsystems being monitored in your enterprise. The larger the enterprise being monitored, the larger the delay should be. This is to ensure that at resync time with high status forwarding to the focal point, any delays due to this do not cause unnecessary missing heartbeat processing.

The SA z/OS Heartbeat Function

The heartbeat function validates the path between remote systems and the focal point system. This path is known as the status forwarding path.

SA z/OS validates the status forwarding path by sending an alert, called the heartbeat alert, from each single remote system or from a designated system in the sysplex at a regular interval.

If the heartbeat is missing from the designated sysplex system, the SA z/OS topology manager tries to connect to another system in the sysplex. Only if all systems are not responding, SA z/OS invokes the missing heartbeat processing.

SA z/OS on the NMC focal point system expects heartbeats from each remote system at the interval defined for each system, but it expects a heartbeat only from one system in the sysplex. This interval is known as the heartbeat interval. Receiving a heartbeat on the NMC focal point system indicates that the status forwarding path from the remote system is up, and hence status for resources associated with that system can be considered to be current.

If SA z/OS on the NMC focal point system does not detect an expected heartbeat, it waits to allow for temporary network slowdowns, or any other temporary problems in the status forwarding path that may have delayed the heartbeat. The wait period is known as the missing heartbeat delay. When the missing heartbeat delay time expires, (or for a sysplex, when the missing heartbeat delay time expires for all systems in the sysplex) SA z/OS invokes its missing heartbeat processing for the remote system or the sysplex and changes the status of all associated resources to Unknown because SA z/OS is no longer able to receive status updates for those resources.

When setting the missing heartbeat delay period, consideration must be given to the number of sysplexes or single systems, and subsystems being monitored in your enterprise. The larger the enterprise being monitored, the larger the delay should be.

When SA z/OS detects that heartbeats from a remote system have resumed, it resynchronizes the NMC workstation status displayed for resources associated with the system by requesting that the remote system should resend all status information.

You can define both the heartbeat interval and the missing heartbeat delay for each MVS system in your enterprise.

Policy Items for Systems

Figure 74 on page 105 shows the z/OS policy items that are displayed when an MVS system is selected from the Entry Name Selection panel.

Notes:
1. Depending on the screen size not all policies may be displayed on one screen.
   Press PF8 to scroll down to see the second part of this panel.
Policy Items for Systems

2. Several of the policy items available from this panel allow you to select and manage the links between the system and a variety of entry types. The panels associated with these policy items are all very similar and straightforward to use and are not covered in this section.

#### SYSTEM INFO

This displays and allows you to update the following:

- The operating system type for this system definition.
- The heartbeat interval and the missing heartbeat delay times.
- Details required for interacting with NetView. This information is required only if you are using enterprise monitoring.
- Details of your operating environment.

See "SYSTEM INFO Policy Item" on page 108 for more information.

#### AUTOMATION SYMBOLS

You can enter or update SA z/OS symbols that can be set to various system-specific values. These can then be substituted into several values, such as MVS automatic restart management element names, MVS workload manager resource names, job names, user specified commands and others. See "AUTOMATION SYMBOLS Policy Item" on page 109 for more information.
Policy Items for Systems

AUTOMATION CONSOLE
The automation console definitions are used when SA z/OS notification messages are to be sent as write-to-operator (WTO) messages to the MVS consoles. This policy item defines the route codes and MCSFLAGS necessary for this process. WTO messages are sent to the Automation Console when none of the notification operators defined in the NOTIFY OPERATORS policy item are logged on or when SYSOP is defined as a notification operator.

For more information on this policy item see "AUTOMATION CONSOLE Policy Item" on page 110.

APPLICATION GROUPS
When you select this policy item, a list of all System Application Groups that are linked to this system is displayed. You can add application groups to or remove them from the system.

MONITOR RESOURCES
This policy item allows you to manage links to Monitor Resource entries. See "Monitor Resource Entry Type" on page 230 for more information.

AUTOMATION TIMERS
This policy item displays a list of defined Automation Timer sets and allows you to choose which sets are to be associated with this system object. See "Timers Entry Type" on page 257 for more information.

USER E-T PAIRS
This policy item displays a list of defined User E-T Pairs sets and allows you to choose which sets are to be associated with this system object. See "User E-T Pairs Entry Type" on page 294 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 Entry Type" on page 290 for more information.

TAPE ATTENDANCE
This policy item displays a list of defined Tape Attendance sets and allows you to choose which sets are to be associated with this system object. See "Tape Attendance Entry Type" on page 261 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 Entry Type" on page 271 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 Entry Type" on page 268 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 Entry Type" on page 267 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 Entry Type” on page 263 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 Entry Type” on page 279 for more information.

AUTOMATION OPERATORS
This policy item displays a list of defined Automation Operator sets and allows you to choose which sets are to be associated with this system object. See “Automation Operators Entry Type” on page 272 for more information.

STATUS DISPLAY
This policy item displays a list Status Display Facility (SDF) Status Definition sets and allows you to choose which sets are to be associated with this system object. See “Status Display Entry Type” on page 291 for more information.

NOTIFY OPERATORS
This policy item displays a list of defined notification operator sets who receive SA z/OS notification messages. See “Notify Operators Entry Type” on page 276 for more information.

OPC SYSTEM DETAILS
This policy item allows you to manage links to OPC System Details. Only one entry should be linked to a system, otherwise multiple entries will be built with SA z/OS only loading the last one that is processed. See “OPC System Detail Entry Type” on page 244 for more information.

CONTROLLER DETAILS
This policy item allows you to manage links to OPC Controller Details. Only one entry should be linked to a system, otherwise multiple entries will be built with SA z/OS only loading the last one that is processed. See “OPC Controller Details Entry Type” on page 245 for more information.

WORKSTATION DOMAINS
This policy item allows you to manage links to workstation domain IDs. Only one entry should be linked to a system, otherwise multiple entries will be built with SA z/OS only loading the last one that is processed. See “OPC Workstation Domains Entry Type” on page 248 for more information.

PROCESSOR
Selection of this policy item displays a panel that lists all processors and partitions defined in the enterprise. You can select one processor or partition that the system should run on. See “PROCESSOR Policy Item” on page 112 for more information.

TARGET SYSTEM INFO
This policy item displays basic information about the target system, for example, time zones, operators receiving messages from the system and operator consoles. See “TARGET SYSTEM INFO Policy Item” on page 112 for more information.
Policy Items for Systems

IPL INFO
With this policy item you can define IPL information for the MVS or VM target system. See “IPL INFO Policy Item” on page 114 for more information.

SYSTEM INFO Policy Item
The panel shown in Figure 75 is displayed when you select policy item SYSTEM INFO on the Policy Selection panel for Systems (Figure 74 on page 105).

You can enter the following information:

Operating system
Operating system shows the operating system that has been specified when creating this system entry. It cannot be changed.

Image/System name
Identifies the system. See “Image/System Name” on page 103 for more details.

Primary JES
The name of the primary Job Entry Subsystem as specified for a JES2 or JES3 subsystem. The named subsystem is the default scheduling subsystem.

System Monitor Time
The time between system monitoring cycles. The default is every 59 minutes. Monitoring may be suspended by specifying NONE.

Figure 75. System Information Panel
You can enter the following information:

Operating system
Operating system shows the operating system that has been specified when creating this system entry. It cannot be changed.

Image/System name
Identifies the system. See “Image/System Name” on page 103 for more details.

Primary JES
The name of the primary Job Entry Subsystem as specified for a JES2 or JES3 subsystem. The named subsystem is the default scheduling subsystem.

System Monitor Time
The time between system monitoring cycles. The default is every 59 minutes. Monitoring may be suspended by specifying NONE.
**SYSTEM INFO Policy Item for Systems**

**Gateway Monitor Time**
The time between gateway monitoring cycles. The default is 15 minutes. Monitoring may be suspended by specifying NONE.

**Automation Table(s)**
You may specify multiple NetView automation tables that are loaded and used for message automation. The automation tables are activated in the sequence as specified. The sequence of the automation tables is not changed during a configuration refresh (INGAMS REFRESH).

The first automation table of the list defined here replaces the SA z/OS initial automation table INGMSG00 that was loaded during initialization. The default is INGMSG01.

**SDF Root Name**
Defines the name used by SDF for the system name. This is the root component that is used during initialization. If nothing is entered the default is the MVS System Name.

**Exit name(s)**
An installation exit or exits invoked to perform additional environment setup. This item is optional. See IBM Tivoli System Automation for z/OS Customizing and Programming for more information.

**USS automation path**
Defines the path where the SA z/OS USS automation programs (for example, INGCCMD) are installed. The SA z/OS installation default is /usr/lpp/ing/ussauto/lib.

**SA NetView Domain**
This field is optional. It allows you to enter the domain ID of the NetView on the local system where SA z/OS runs. The field may be used for customer-specific and documentation purposes.

**Network NetView Domain**
This field is optional. It allows you to enter the domain ID of the NetView on the local system where the network automation runs. The field may be used for customer-specific and documentation purposes.

**Heartbeat Interval**
See “Heartbeat Interval” on page 103 for more details.

**Missing Heartbeat Delay**
See “Missing Heartbeat Delay” on page 103 for more details.

**AUTOMATION SYMBOLS Policy Item**
The panel shown in Figure 76 on page 110 is displayed when you select policy item AUTOMATION SYMBOLS from the Policy Selection panel for Systems Figure 74 on page 105.
This panel allows you to enter or update system automation symbols that can be set to various system-specific values. These can then be substituted into several values, such as MVS automatic restart management element names, MVS workload manager resource names, job names, user-specified commands and others.

To specify that AOCCLONE value \(x\) is to be substituted, enter \&AOCCLONE\(\_x\) at the appropriate point in the field. You must specify both the leading ampersand (\&) and the trailing period (\.).

You may, additionally, specify the tilde (\~) to get the 0th (unnumbered) AOCCLONE value substituted. For more details see “Using System Symbols and System Automation Symbols” on page 9.

This panel has the following fields:

**General Use**
This shows the description of the general use of the system automation symbol. It is defined in the SYMBOL DESCRIPTION policy item of the enterprise definitions (entry type ENT), see “SYMBOL DESCRIPTION Policy Item” on page 83.

**AOCCLONE\(\_x\)**
This allows you to specify the value of the system automation symbol (\&AOCCLONE\(\_x\) variable).

**Description**
This allows you to specify a specific description of the automation symbol for this system only.

**AUTOMATION CONSOLE Policy Item**
The panel shown in Figure 77 on page 111 is displayed when you select policy item AUTOMATION CONSOLE from the Policy Selection panel for Systems Figure 74 on page 105.
In order to define an automation console, you need to specify:

**Console Name**
This parameter defines the 1-8 alphanumeric console name specified in the CONSOLnn member of the z/OS PARMLIB.

**Route Codes**
The appropriate z/OS route codes can be selected from this panel. These route codes define which kind of messages the automation console is to receive. Each notification message has already been defined as one of these kinds of WTOs. The console can also be defined to receive all of the notification messages regardless of how they have been defined. The last four codes on the panel are reserved for user-defined codes.

The default routing code is 1 (only messages that appear as Master Console Action WTOs are displayed on the console). Further information on the route codes can be found in the MVS System Messages books.

**MCSFLAGS**
Select the appropriate MVS MCSFLAG from the list. These codes are used to provide additional information for the WTO messages (for example, if flag 7 is set, the time is not included with the WTO).

If flag 1 or 2 is set, the console name is required and the messages are queued to the specified console.

Further information on the MCSFLAG codes can be found in *Tivoli NetView for z/OS Customization: Using REXX and the NetView Command List Language*.
PROCESSOR Policy Item

The panel shown in Figure 78 is displayed when you select policy item PROCESSOR on the Policy Selection panel for Systems (Figure 74 on page 105).

This panel allows you to link a system to the target hardware. It displays a list of:

- All processors together with their LPARs
- All processors in basic mode

The panel displays the following information:

**Processor**
The processor entry name.

**CPC Name**
The CPC Name/NAU as specified on the Processor Information panel. This field may be blank if data for the Processor Information policy has not yet been specified.

**LPAR Name**
The name of the logical partition as specified on the LPAR Definitions panel. This field is empty for basic processors.

---

**TARGET SYSTEM INFO Policy Item**
The panel shown in Figure 79 on page 113 is displayed when you select the TARGET SYSTEM INFO policy item on the Policy Selection panel for Systems (Figure 74 on page 105).
Fill in the following fields:

**Initialize target system**
Enter YES to indicate that communication to the target system is automatically initialized when SA z/OS processor operations starts.

Enter NO if the target system communication should not be established when processor operations starts and the target status should be set to UNKNOWN. In this status the target system communication will be automatically initialized as soon as another target system that is running on the same LPAR mode hardware is initialized.

Enter BYPASS to set the target system status to CLOSED and to prevent automatic initialization communication to the target system when processor operations starts.

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

**Load Profile Name**
Enter the load profile name defined at the support element. The name you choose must correspond with the load profile defined at the support element. Leave this field blank to take the default defined at the support element. The profile name can contain only the characters A to Z and 0 to 9.

**Target time same as host**
If the target system is in the same time zone as the focal-point system, type YES or Y. If not, leave this field blank. For systems that run as guest systems under VM this field must be set to YES.

**Target time earlier**
Enter the number of hours that the target system is earlier than the focal point system. (A target system in Los Angeles is three hours earlier than a focal-point system in New York.)
TARGET SYSTEM INFO Policy Item for Systems

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.)

System Console
Enter the user IDs of the operators you want to receive system console messages. The operators listed receive messages only from the system console. You can enter any valid NetView user ID. This field is optional.

Operator Console
Enter the user IDs of the operators you want to receive operator console messages. You can enter any valid NetView user ID. This field is optional.

Notes:
1. For OCF-based systems using only console integration, the operating system messages are forwarded to the system console interested operator list.
2. Operators not on these lists do not see messages from these consoles unless their user IDs are added using the ISQXMON command when SA z/OS is running.

IPL INFO Policy Item
The panels that you can use to define your target system IPL information depend on the operating system that you are running.

MVS Target System IPL Information
The panel shown in Figure 80 is displayed when you select the IPL INFO policy item on the Policy Selection panel for Systems Figure 74 on page 105 for MVS.

Figure 80. MVS Target System IPL Information Panel
IPL INFO Policy Item for Systems

Use this panel to define responses that are used during the MVS or z/OS target IPL process.

During the start process, the target system requests information from the operator. By providing the responses at this panel, the operator enables SA z/OS to respond to these messages. Target systems using VSE, LINUX, or CF do not request information from the operator during startup.

IEA101A specify system parameters
Enter the desired response to the IEA101A message. All IEA101A message responses begin with the characters R 0. You do not have to begin your response with these characters, they are provided by SA z/OS automation. Quotation marks are not necessary. The customization dialog does not verify your entry.

IEA347A specify master catalog parameter
Enter the desired response to the IEA347A message. The customization dialog does not verify your entry.

IEA212A or IEA213A/IEA214A DUPLICATE VOLUME
Specify the device addresses to be kept online if duplicate DASD volumes exist on a z/OS system at IPL time. You have 24 entries to specify real or generic DASD device addresses. Example for a generic device address: 012*. This indicates that all addresses from 0120 to 012F must remain online.

This entry works in conjunction with the automatic response selection.

Provide default response when neither device is specified. Specify YES or NO to indicate whether a default response is required if neither or both of the devices in the IEA213A/IEA214A message match those identified as required devices. For IEA214A messages, the current IPL (SYSRES) device is considered as being implicitly defined. If YES is selected, an automated response will always be provided. For a SYSRES device, the device shown in the message is dismounted. Otherwise the following applies:
- If neither device is in the reply list, the first device is demounted.
- If only one device is in the reply list, the remaining device is demounted.
- If both devices are in the reply list, the first device in the list as scanned from left to right remains mounted and the other device is demounted.
- If both devices are in the reply list due to the same generic device mask in the list as scanned from left to right, the second device is demounted.

A selection of NO will result in a response allowing the IPL to continue only if one and only one device is identified as being required. Otherwise, message ISQ1203 will be issued and the IPL process will be suspended. In such a case, the operator needs to react to system message ISQ1203.

A selection of NO and a generic device address of **** will always stop the IPL process if a duplicate volume is detected.
IPL INFO Policy Item for Systems

Notes:
1. For valid responses, see z/OS MVS System Messages, Vol 6 (GOS-IEA).
2. Responses to IEA101A, IEA347A, IEA212A, and IEA213A can be changed without generating and loading new configuration files by the use of the host ISQXOPT command before IPLing the target system. However, these changes are not saved if SA z/OS processor is stopped or restarted.
3. You receive either message IEA212A or IEA213A depending on the level of your MVS system.
4. All message responses begin with the characters R 0. You do not have to begin your response with these characters; SA z/OS automation provides them. Quotation marks are also not necessary.

VM Target System IPL Information
To define the response to the VM IPL START message, use the panel shown in Figure 81. During the start process, the target system requests information from the operator. By providing the responses at this panel, the operator enables SA z/OS to respond to these messages.

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

Note:
The VM IPL START message response field can be changed without generating and loading new configuration files. You can do this by issuing the host ISQXOPT command before you IPL the target system. However, these changes are not saved if processor operations or SA z/OS is stopped or restarted.

ApplicationGroup Entry Type
An application group is a set of applications that you assign a name to. 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 or more sysplexes. Sysplex Application Groups are used to monitor and automate applications and application groups that can run on multiple systems in a sysplex. Using Sysplex Application Groups means you do not need to connect the application group to each system of the sysplex. Instead, all applications of this Sysplex Application Group are known on all systems of the sysplex. You can then monitor the group at the sysplex level, rather than at system level.

The SA z/OS customization dialog allows concurrent user access capability for entry type ApplicationGroup. See also Appendix A, “Concurrent Customization by Multiple Users,” on page 357 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 19.

Figure 82 shows the panel that is displayed when you create a new application group with the New command from the Entry Name Selection panel for application groups.

The unique fields are:

**Entry Name**

This is the name of the application group. It must be unique within the policy database. Valid names may contain up to twenty alphanumeric characters.

**Application Group Type**

This indicates whether the Application Group will be used for Systems or for a Sysplex. This attribute sets the boundaries for the selection of Resource members of the Application Group, and should be set as follows:
Creating a New ApplicationGroup

**SYSPLEX**
Use this type if the Application Group is being used to define sets of Applications and Resources that may be on multiple systems within a sysplex or sysplexes.

**SYSTEM**
Use this type if the Application Group is being used to define sets of Applications and Resources that be on single systems.

Application Group Types cannot be changed after this panel has been left.

**Nature**
In this field you can define the availability behavior of application groups and applications. You can specify one of the following attributes:

**BASIC**
The group is available when all of its resource members are available.

**MOVE**
The group is available when one of its resource members is available.

All members should be instances of the same application.

To avoid the following conflict, you should not define an application to be a member in two MOVE groups: Assume that resource RES1 is a member in two MOVE groups. One of the MOVE groups starts a member that 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. The move group will ensure that only one member of the group will be active at any point in time.

**SERVER**
This is like a move group. Additionally, you need to define an availability target that determines how many of its members should be available to define the server group as being available, as well as a satisfactory target that specifies the number of members in the group that must be active before the automation manager treats the group as being in a satisfactory state.

All members should be instances of the same application.

If you do not make a specification, the default is BASIC.

Depending on your selection in the **Nature** field, the process of entering resources into a group is different, as described in “How to Work with Resources” on page 33. This section also provides examples of how to use the different natures appropriately.

**Default Preference**
This field specifies what preference should be used for application resources when automatically linking applications to ApplicationGroups (note that the **Automatically link** field below must be set to YES for automatic linking and that preference values are only used for SERVER and MOVE ApplicationGroups). The *DEF preference corresponds to default preference 700. This is the default for this field. For information about preference values and their meanings see “RESOURCES Policy Item” on page 132.
Creating a New ApplicationGroup

If you want to automate and monitor the current application group an
automation name is required. If possible, the specified entry name is used
as a default and automatically inserted into the field but can be removed
again.

At the time an application group is created, it is not yet known which
systems you want to link it to. So you need to use its WHERE USED policy
item to connect it to the systems you want. Suppose you created a System
Application Group called SYST_APG and linked it to systems SYSTEM1
(with the same MVS SYSNAME) and SYSTEM2 (with the same MVS
SYSNAME). After pressing the END key, SA z/OS generates the following
resources:

SYST_APG/APG/SYSTEM1
SYST_APG/APG/SYSTEM2

If you do not provide an automation name, an ACF fragment is not built
for the APG and thus the APG is not included in the ACF main include
member (ACFZxxx). If you change the automation name of an APG to or
from blank (that is, no name is specified), you need to rebuild the main
include member to reflect this change.

Automatically link

This entry field specifies whether applications that will later be linked to
the current ApplicationGroup should also be automatically linked as
resources. This field should be changed to NO if no more resources should
automatically be added if the group is linked to another system or sysplex.
Otherwise the automation behavior might not be as expected for complex
group definitions (that is, groups with manually adapted preference values
or availability targets other than *ALL).

Note: This field is only relevant for application groups that have a
specified automation name. If no automation name is specified the
value cannot be set to NO. In this case the resources are always
linked and cannot be handled individually.

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.
Creating a New ApplicationGroup

PASSIVE
Does not propagate the requests to the group members. This enables monitoring and disables the automation of the application group.

Prepare Move
Specifies whether the downtime during the move or switch of a resource should be kept as short as possible.

For an application group of nature MOVE this means that the group member that is currently running will not be stopped until the replacement member is ready to be started, for example, all parents are available.

For an application group of nature SERVER this means that a group member will not be stopped until the replacement member that has just started is available.

To have the highest possible availability, it is recommended that you set this value to YES unless the application cannot support this.

If you leave this the field blank the value can be inherited from the system or sysplex defaults definition.

Note: This field is only relevant for application groups with a nature of MOVE or SERVER. If nothing is specified or inherited the default value is YES.

Move Mode
Specifies how a move is made from an old member to a new one:

PARALLEL
The new member is started at the same time as the old member is being stopped. This is the default.

SERIAL
The new member is started after the old member has been stopped completely.

Note: This field is only relevant for groups of type MOVE. The default value is PARALLEL. For groups of any other type it is forced to blank.

Policy Items for ApplicationGroups
As described in “Creating a New ApplicationGroup” on page 117, there are two types of application groups:
• System Application Groups
• Sysplex Application Groups

Figure 83 on page 121 shows the policy items that are displayed when a System Application Group is selected from the Entry Name Selection panel.
The unique policy items for application groups are:

**APPLGROUP INFO**
This policy item displays information about the Application Group, as specified when creating the application group. The panel is very similar to the one used for creating the application group. You may update indicated values in this panel. See “APPLGROUP INFO Policy Item” on page 122 for more information.

**APPLICATIONS**
When you select this policy item, a list of all applications in the SA z/OS enterprise is displayed, indicating which, if any, are associated with this 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, 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 “APPLICATIONS Policy Item” on page 125 for more information on this policy item.

**TRIGGER**
Each application group may be linked to one trigger. With this policy item, you can use the Trigger for ApplicationGroup panel to link the application group to a trigger so that all of its applications will be started or stopped depending on the defined trigger conditions.

**SERVICE PERIOD**
With this policy item, you can use the Service Period for ApplicationGroup panel to link a service period to the application group. Its resources are then automatically started up or shut down in the time frame specified by the service period.

**RELATIONSHIPS**
This policy item has the same functionality for entry types Application, ApplicationGroup, and Monitor Resource. For each application, application group, or monitor resource you may specify several relationships. They will be used to automate the startup or stopping of those applications,
application groups, and monitor resources. If you select this policy item, in
the Relationship Selection List panel, you can view existing relationships or
define new relationships for the current application, application group, or
monitor resource.

You define relationships between dependent resources and supporting
resources. In the customization dialog you define relationships always from
the point of view of the dependent resource. So the resource that you have
currently selected will become the dependent resource, when you define a
new relationship.

See “RELATIONSHIPS Policy Item” on page 126 for more information on
viewing existing or defining new dependency relationships.

MESSAGES/USER DATA
This policy item has the same functionality as for applications and MVS
components. It allows you to specify automated actions, such as the
issuing of a command or reply, to be taken in response to a message.
However, because application groups do not produce ‘real’ messages, this
policy behaves differently. Currently only the special message ID,
INGALERT, together with the command definitions assigned to it are
honored by SA z/OS.

How to proceed in specifying this policy item is described in
“MESSAGES/USER DATA Policy Item” on page 165, where the policy is
described for applications.

RESOURCES
This policy item allows you to select resources for current
ApplicationGroup and set their preferences or name.

The panels displayed look different depending on the nature of the current
application group. See Chapter 3, “Scenarios on How to Use the
Customization Dialog,” on page 33 and “RESOURCES Policy Item” on
page 132 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 “GENERATED RESOURCES Policy Item” on
page 138 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 that the
current application group resource is a member of. See the Member of
panel in “MEMBER OF Policy Item” on page 139 for more details.

APPLGROUP INFO Policy Item
A panel like Figure 84 on page 123 is displayed if you select policy item
APPLGROUP INFO from the Policy Selection Panel for Application Groups.
You can update the following values that are shown in the lower part of the panel:

**Nature**

See "Nature" on page 118 for more details.

**Default Preference**

See "Default Preference" on page 118 for more details.

**Automation Name**

See "Automation Name" on page 119 for more details.

**Behaviour**

See "Behaviour" on page 119 for more details.

**Desired Available**

Lets you specify the default desired status of the resource. The desired state of each resource is either Available or Unavailable, which is the goal that automation tries to achieve.

You can specify the following values:

**ALWAYS**

The desired state of the resource is set to Available, unless it is dependent on a resource that has a Desired Available setting of ONDEMAND. In this case the resource behaves as if it had a Desired Available setting of ONDEMAND itself.

This is the same behavior as in earlier releases of SA z/OS.

**ONDEMAND**

If there is demand for the resource to be available, its desired status is set to Available, otherwise its desired status is set to Unavailable.

Demand arises either from propagated MakeAvailable votes or implicitly through membership of a non-passive basic application
group (APG) that has a desired status of Available. Demand does not arise from dependent resources with a Desired Available setting of ALWAYS.

A MakeAvailable vote that is propagated to the resource overrides any demand considerations.

An active ONDEMAND member of a move or server group is always sent a vote that sets its desired status, thus overriding any demand considerations.

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

If you leave the field blank the value can be inherited from the system or sysplex defaults definition.

**Note:** If nothing is specified or inherited the default value is ALWAYS.

**Prepare Move**  
See “Prepare Move” on page 120 for more details.

**Move Mode**  
See “Move Mode” on page 120 for more details.

**Status Determination**  
Specifies how the compound status of the application is going to be determined:

**CSONLY**  
The worst compound status of any of the group members is used to define the compound status of the application group.

Leave this field blank if the application group’s compound status should be determined from several statuses, like observed status, desired status, automation status, startability status and others.

**Inform List**  
This field allows you to specify:
- Where the application is registered to
- Where all status changes are propagated to
- Whether SMF records are written

Valid receivers are SDF, NMC, IOM, SMF, EIF, TTT and USR. You can specify more than one receiver.

The list of values can be blank or comma separated. Neither system automation symbols (AOCCLONEs) nor system symbols are allowed.

If you leave the field blank, the value from the system defaults (SDF) or sysplex defaults (XDF) will be used if specified there.

Specify NONE to prevent the default being used from the system defaults definition (for application groups of type SYSTEM, see “System Defaults Entry Type” on page 268) or the sysplex defaults definition (for application groups of type SYSPLEX, see “Sysplex Defaults Entry Type” on page 288).

**Owner**  
Specifies who the operator should contact if there is an error.
APPLGROUP INFO Policy Item for ApplicationGroups

Info Link
This field can be used to specify a location (for example, a URL) where additional information about the application can be found.

Automatically link Application-Resources
Specifies whether linked applications should also be automatically linked as resources. See also 119.

APPLICATIONS Policy Item
If you select an application group's policy item APPLICATIONS, you reach the Applications for ApplicationGroup panel. Figure 85 shows an example.

Here you can add or remove applications from an application group. If you add an application to an application group that is already linked to a system, SA z/OS will generate a resource for this application. If you previously set the option Automatically link Application-Resources to YES when creating the application group, SA z/OS will also add the new resource into this application group. Otherwise, you simply have an independent application resource that you can start, stop, or automate as required.

TRIGGER Policy Item
If you select an application group’s TRIGGER policy item, you reach the Trigger for ApplicationGroup panel if triggers have been defined for the application group. Figure 86 on page 126 shows an example. Use the panel to add triggers to or remove them from an application group. For more details about triggers, see the section “Event and Trigger Support” in IBM Tivoli System Automation for z/OS User’s Guide.
SERVICE PERIOD Policy Item

If you select an application group's SERVICE PERIOD policy item, you reach the Service Period for ApplicationGroup panel if service periods have been defined for the application group. Figure 87 shows an example. Use the panel to add service periods to or remove them from an application group.

RELATIONSHIPS Policy Item

The Relationship Selection List panel (Figure 88 on page 127) is displayed if you select policy item RELATIONSHIPS from the Policy Selection panel for application groups. It shows information about existing relationships and allows you to define new relationships.

The ACTION pull-down choice offers actions to edit and delete selected relationships. The Delete command immediately deletes the relationship. The Edit command leads to the Define Relationship panel (as shown in Figure 88 on page 127).
This panel displays information about existing relationships. In all of the relationships for the current resource, this resource is the dependent resource. The columns have the same meaning as described in “Defining Relationships.”

Enter the NEW command on this panel to define new relationships as described in “Defining Relationships.”

**Defining Relationships**

You define relationships between dependent resources and supporting resources. In the customization dialog you always define relationships from the point of view of the dependent resource. So the resource that you are currently editing becomes the dependent resource for the relationship dependency that is being defined.

If you select this policy item, the Define Relationship panel is displayed, as shown in Figure 89. You specify the name of the supporting resource in the **Supporting Resource** field.

![Figure 88. Relationship Selection List Panel](image)

This panel displays information about existing relationships. In all of the relationships for the current resource, this resource is the dependent resource. The columns have the same meaning as described in “Defining Relationships.”

**Defining Relationships**

You define relationships between dependent resources and supporting resources. In the customization dialog you always define relationships from the point of view of the dependent resource. So the resource that you are currently editing becomes the dependent resource for the relationship dependency that is being defined.

If you select this policy item, the Define Relationship panel is displayed, as shown in Figure 89. You specify the name of the supporting resource in the **Supporting Resource** field.

![Figure 89. Define Relationship Panel](image)

This panel shows either the automation name (**Autoname**) for ApplicationGroups, or the subsystem name (**Subsystem**) for Applications.

This panel contains the following fields:
RELATIONSHIPS Policy Item for ApplicationGroups

Description
You can enter text up to 40 characters to describe the relationship. It is stored only in the policy database and is not created in the automation configuration files.

Relationship Type
Define the relationship between the application group, referred to as the dependent resource, with any other resource, referred to as the supporting resource. The supporting resource can be an application, application group, or monitor resource.

Relationships and conditions are only observed by automation if the requested goal of the resources that are involved allow them. Dependencies are only checked if they are of importance to the existing desired goal.

Enter one of the following values:

MAKEAVAILABLE
The dependent resource cannot be started until the specified Condition has been fulfilled for the supporting resource. For example, if you specify WhenAvailable as the relationship condition, SA z/OS starts the dependent resource as soon as the supporting resource itself is available, if no request or goal contradicts this.

Note: If a goal is not explicitly specified, the default goal is to 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, 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, the SHUTINIT commands for the dependent resource are issued as soon as the specified Condition has been satisfied for the supporting resource.

FORCEDOWN
This forces a shutdown of the dependent resource if the supporting resource has reached its specified satisfactory condition.

EXTERNALLY
At certain times the supporting resource performs various automated actions for the dependent resource. These actions are specified through the relationship condition (see "Condition" on page 131). For example, with this selection you specify that an external resource is responsible for starting or stopping the current (dependent) resource.

HASMONITOR
Use this for relationships that define connections between application or application group resources and monitor resources.
HAS PARENT
With this selection you specify that the current application or application group has a parent. The parent is the supporting resource that you specify in the Supporting Resource field (see "Supporting Resource").

The dependent resource cannot be started until the supporting resource is available.

The supporting resource cannot be stopped before the dependent resource is unavailable.

At certain times the supporting resource performs various automated actions for the dependent resource. These actions are specified through the relationship condition (see "Condition" on page 131). For example, if you specify StartsMe as the relationship condition, the supporting resource starts the dependent resource whenever the supporting resource is started.

A request to start the dependent resource results in issuing a start request against the supporting resource if the supporting resource is down.

HAS PASSIVE PARENT
The only difference between the HasPassiveParent and HasParent relationship is that a request to start the dependent resource does not result in the issuing of a start request against the supporting resource if the supporting resource is down. A start request must be issued against the supporting resource before the start request for the dependent resource is processed.

PEER OF
The PEEROF relationship means that any MakeAvailable votes that arrive at the supporting resource are propagated to the dependent resource.

Any MakeUnavailable votes that arrive at the supporting resource are not propagated to the dependent resource unless SCOPE=ALL has been specified.

Supporting Resource
Here you enter the name of the resource that the dependent resource has the previously defined relationship with. This name consists of two or three parts, separated by slashes, where the last part shows the system name. You can use the equal sign (=) which then stands for the local system of the dependent resource. The syntax for valid formats is similar to the syntax for generated resources:

- APG-AutomationName/APG (for sysplex APG)
- APG-AutomationName/APG/MVSSysname (for system APG)
- APG-AutomationName/APG/= (for system APG)
- SubSystemName/APL/MVSSysname (for APL)
- SubSystemName/APL/= (for APL)
- MonitorName/MTR/MVSSysname (for MTR)
- MonitorName/MTR/= (for MTR)

Note: If the supporting resource is an APL resource, an MTR resource, or a system APG resource, one system automation symbol (&AOCCLONE., ~ or &AOCCLONE_{x} with x=1 through 9, A through Z) can be specified in the name. System symbols cannot be specified.
RELATIONSHIPS Policy Item for ApplicationGroups

For MVSSysname the special notation with a trailing plus sign for an alternate system is allowed.

Sequence Number
This number is used when the ACF is built. Each relationship that has a number assigned to it is put in the ACF PARENT=(...) field. The first subsystem name in the list where the subsystem is defined in the current sysplex is used for AOCQRY. The position in the list is given by this number. Relationships with the same number are sorted within their sequence number. Valid values are 1 - 99 and blank.

Note: Specifying this number is only valid for APL resources with a local system specification, that is, subsysname/PL/*. The number is forced to blank for relationships where the supporting resource is an APG, an MTR or an APL without a local system specification.

Automation
Defines the processing of SA z/OS to satisfy the specified relationship as follows:

ACTIVE
Specifies that SA z/OS attempts to bring the resource to the desired condition to satisfy the dependency. This is the default for MAKEAVAILABLE and MAKEUNAVAILABLE.

PASSIVE
Specifies that SA z/OS waits until the resource is in the desired condition to satisfy the dependency. That is, the request to bring the resource into the desired condition needs to come from outside, such as an operator command.

This field is forced to blank for the following relationships:
• HASMONITOR
• HAPARENT
• HASPASSIVEPARENT
• EXTERNALLY
• FORCEDOWN
• PEEROF

It is forced to PASSIVE for the following conditions:
• WhenGroupHasNotFailed
• WhenHardDown
• Any WhenObserved... condition

Chaining
Defines whether the status of the supporting resource or attached subtree is to be considered:

STRONG
SA z/OS considers the status of the attached subtree.

WEAK
SA z/OS considers only the status of the supporting resource (this is the default for MAKEAVAILABLE and MAKEUNAVAILABLE).

This field is forced to blank for the following relationships:
• HASMONITOR
• HAPARENT
• HASPASSIVEPARENT
RELATIONSHIPS Policy Item for ApplicationGroups

- EXTERNALLY
- FORCEDOWN
- PEEROF

It is forced to WEAK for PREPAVAILABLE and MAKEAVAILABLE relationships with a WhenGroupHasNotFailed condition.

Strong chaining causes SA z/OS to check all the relationships of the resources along the attached subtree. Weak chaining simply requires that the action is allowed on the dependent resource.

In the example shown in Figure 90 for strong chaining, SA z/OS also considers the dependencies between resources B and C, and B and D: A is only started if B, C, and D are up, or C is only stopped if A and B are down.

![Figure 90. Example of Strong Chaining](image)

**Condition**

Defines the condition that the supporting resource must have to satisfy the relationship. Enter ? to go to the Relationship Condition Selection panel from where you can select an appropriate value.

See “Relationship Conditions” for more information on available condition values and the context that they may be used in. If no value is entered for this field the following default values are set:

<table>
<thead>
<tr>
<th>Relationship</th>
<th>Default Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORCEDOWN</td>
<td>WhenObservedDown</td>
</tr>
<tr>
<td>HASMONITOR</td>
<td>Forced to blank</td>
</tr>
<tr>
<td>MAKEAVAILABLE</td>
<td>WhenAvailable</td>
</tr>
<tr>
<td>MAKEUNAVAILABLE</td>
<td>WhenDown</td>
</tr>
<tr>
<td>PREPAVAILABLE</td>
<td>WhenAvailable</td>
</tr>
<tr>
<td>PREPUNAVAILABLE</td>
<td>WhenDown</td>
</tr>
<tr>
<td>HASPARENT</td>
<td>blank</td>
</tr>
<tr>
<td>HASPASSIVEPARENT</td>
<td>blank</td>
</tr>
<tr>
<td>PEEROF</td>
<td>Forced to blank</td>
</tr>
<tr>
<td>EXTERNALLY</td>
<td>StartsMe</td>
</tr>
</tbody>
</table>

**Relationship Conditions:** The relationship conditions that are available depend on the relationship.

See the online help for details of available relationship conditions.
RESOURCES Policy Item for ApplicationGroups

RESOURCES Policy Item

When you select the RESOURCES policy item, you are first presented with a selection panel if the application group is linked to more than one system or sysplex. There are three different panels depending on whether the application group is of type Server, Move, or Basic. The panel shown in Figure 91 is for an application group of type Server.

If you want to add resource members to an application group of nature MOVE or SERVER, you can attach preference values to those resources (using the Preference column in Figure 91).

This panel has the following fields (unless stated, the fields apply to all application group types):

**Number of selected Resources**
This shows the current number of selected Resources.

**Availability Target**
This Availability Target is different depending on the type of application group:

**Server**
Valid values for server application groups are:

*ALL  This means that all selected resources must be available before the current application group-derived resource is considered as available.

number  This (positive) number is the number of selected resources that must be available before the current application group-derived resource is considered as available. This number must be less than or equal to the value in the Number of selected Resources field.

If you enter a value that is greater than the value in the Number of selected Resources field, it is changed to the maximum allowed when you leave the panel.

-number  This (negative) number is the number of selected resources less than the total that must be available before the current application group-derived Resource is considered as available.
RESOURCES Policy Item for ApplicationGroups

available. This number must be less than the value in the Number of selected Resources field.

If the value entered is less than or equal to the minimum value that is allowed, it will be changed to the minimum when you leave the panel.

**Move** The Availability Target is always 1. This field cannot be modified for move application groups.

**Basic** The Availability Target is not applicable and is not displayed for basic application groups.

**Satisfactory Target**

The Satisfactory Target is only applicable to server application groups. It specifies the number of members in the group that must be active before the automation manager treats the group as being in a satisfactory state.

Valid values for server application groups are the same as for the Availability Target.

The Satisfactory Target may be less than, or equal to the Availability Target, but cannot be higher.

If the Satisfactory Target is of type -number then it is treated relative to the adjusted Availability Target.

**Show all Resources**

This can be:

**YES** Shows all selectable resources.

**NO** Only shows selected resources.

**Show only Resources with String**

If you enter a string in this field, only matching resources are listed. Wildcards are not supported.

**Action**

The following actions are valid from the ACTIONS menu and Action prompt:

(P)RIMARY Sets the preference value to 700.

SECON(D)ARY Sets the preference value to 400.

(A)DD Sets the preference value to 700.

RE(M)OVE Removes the preference value.

(S)ELECT Sets the preference value to 700.

**Preference**

A resource becomes SELECTED when a positive number is entered to this field. The preference value says what "role" the selected resource has. The maximum valid value is 3200. Any value greater than 3200 will be changed to that valid maximum.

The following roles are related to the preference values given:

>2600 The resource always selected regardless of its state. Use this value if the APL or APG resource must run.

≥2000 The resource always selected unless the observed status of the resource is SYSGONE or HARDDOWN. It is only selected when no other viable member exists. A viable member is a resource that can be started and that is allowed to be started. In principle, the observed status is one of the following: SoftDown, Starting, Degraded, Available or WasAvailable.
The resource always started, other resources stopped. (Use if you really want the APL or APG resource to run).

700  The resource is selected for startup after an IPL and after recovery situations. Members with preference 300 will be stopped in favour of this group. It is not stopped unless a very high preference alternative becomes available. This value is the default. Use it in a MOVE group for a resource that normally runs.

600  Used as a threshold. A resource with a preference of 600 and lower will not be selected at group startup.

500  This resource is not selected for startup after IPL but it is started as a backup after failure and continues running even though normal preference (700) alternatives become available. It is stopped at the next (scheduled) outage. Use this value in a MOVE group for the backup resource.

400  This resource is not selected for startup after IPL but it is started as a backup after failure. The difference to preference 500 is that the resource is stopped when normal preference (700) alternatives become available again.

300  Like 200, but a move to an alternate backup (Preference 500) would only occur at a (scheduled) outage.

200  Resource started as backup, stopped as soon as better alternative (Preference 450 and greater) becomes available. (Use where you can have a backup, but you don’t really want one).

1  This means that the member will always be deselected and will always have a MakeUnavailable vote propagated to it. This can be useful if you want complete manual control of the move process.

0  This means that the member is a passive member and will never get a vote propagated to it by the group. If a member has its preference changed so it becomes passive, any votes that have been propagated to it from the group will be withdrawn.

Note:
SA z/OS offers the possibility to consider the workload in its decision making via resource aware application move. This only takes place if the applications in a system or sysplex have the same effective preference values. It also does not affect any preference values that you have defined for your resources. For more details see “Resource Aware Application Move” in IBM Tivoli System Automation for z/OS User’s Guide.

Preferences other than those listed are accepted. You can find more information on the meaning of preferences in Preference field description on page 133 and in “Controlling Application Move with Preference Values” in IBM Tivoli System Automation for z/OS User’s Guide.

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

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

1000
RESOURCES Policy Item for ApplicationGroups

System Automation for z/OS: Defining Automation Policy
RESOURCES Policy Item for ApplicationGroups

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

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

<table>
<thead>
<tr>
<th>Value</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>3600</td>
<td>The member has an adjusted preference of 0 (indicating a passive member) and it has an observed status of Starting, Degraded, Available or WasAvailable.</td>
</tr>
<tr>
<td>100</td>
<td>The member has a working preference between 2–2599 and requires a resource on an Avoided system.</td>
</tr>
</tbody>
</table>
| 1     | The member has:  
• An adjusted preference of 0 (indicating a passive member) and its observed status is none of Starting, Degraded, Available or WasAvailable.  
• A working preference of 2000–2599, there is a viable alternative resource and the member either has an observed status of HardDown or SysGone or the system that it is on is being stopped.  
• A working preference <2000 and has observed status of HardDown or SysGone.  
• A working preference <2600 and has observed status of StandBy, Unknown or Problem.  
• A working preference <2600, has observed status of Stopping or SoftDown and its automation flag is turned off.  
• A working preference <2600 and requires a member on an excluded system. |

After all of these adjustments have been made, this produces the member’s effective preference. Normally a Move or Server group will pick the members with the highest effective preferences to be available, however there are three special rules to take into consideration:

1. A member with an effective preference of 1 may never be picked.  
SA z/OS sends a MakeUnavailable vote.

2. A member with an effective preference <600 may only be picked if the group is in recovery mode (that is, it is recovering from a failure).

3. A member requiring a resource on an excluded system must be sent a MakeUnavailable vote, even if it has been picked. This can occur with resources that have adjusted preferences >2600.

Various scenarios are described in Table 6 on page 136.
## RESOURCES Policy Item for ApplicationGroups

### Table 6. Preference Value Scenarios

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute manual control</td>
<td>Member to be activated gets preference 3200, all others get preference 1. The member with the 3200 preference will be selected. All others will always have a MakeUnavailable vote propagated to them. This does not change even if the system with the selected member leaves the sysplex. A shutdown of the primary system will not initiate a move of the application, as the 400 point penalty is not sufficient to drop its preference below 2600.</td>
</tr>
<tr>
<td>Move only after system failure</td>
<td>Primary member defined preference 2800, backups preference 500. As long as the primary is not SYSGONE it will be selected. If the primary is SYSGONE, its preference drops to 2575 (2800 – 400 + 175), allowing an alternative viable member to be selected. The backups will only be viable if the resource was previously active because of the groups recovery mode. A shutdown of the primary system will not initiate a move of the application. Although the primary systems preference will be below 2600, there will be no viable alternatives as the backups may only be used if the group is in recovery mode.</td>
</tr>
<tr>
<td>Move only if system down or stopping</td>
<td>Primary member defined preference 2800, backups preference 700. As long as the primary member is not SYSGONE and its system is not being stopped it will be selected. If the primary member is SYSGONE or its system is being stopped, its preference drops to 2575 (2800 – 400 + 175), allowing an alternative viable resource to be selected. The backups are always viable. A shutdown of the primary system will initiate a move of the application during the shutdown.</td>
</tr>
</tbody>
</table>
**RESOURCES Policy Item for ApplicationGroups**

Table 6. Preference Value Scenarios (continued)

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Description</th>
</tr>
</thead>
</table>
| Sticky move only after system down or stopping | Primary member defined preference 2801, backups preference 2800.  
As long as the primary member is not SYSGONE and its system is not being stopped, it will be selected. If the primary becomes SYSGONE or its system is being stopped, its preference drops to 2576 (2801 – 400 + 175), allowing one of the backups to be selected. The sticky bonus raises its preference to 2975 (2800 + 175), meaning that even when the primary returns the selected backup system will remain selected.  
It will only be deselected if it goes to SYSGONE or its system is being stopped, in which case the primary will be re-selected if it is available otherwise another backup would be picked.  
A shutdown of the primary system will initiate a move of the application during the shutdown. |
| Move only after system or application failure | Primary member defined with preference 2400, backups preference 500.  
As long as the primary member does not fail to SYSGONE or HARDDOWN it remains selected. If it gets to HARDDOWN or SYSGONE without failing, there are no viable alternatives, so it will not be moved. If it does fail, its preference is between 2600 and 2000 and the backups are viable, so one of them can be picked.  
The backups are only viable if the group is in recovery mode (that is, the member was active when the failure occurred). |
| Move only if system down, being stopped or application failure | Primary member defined with preference 2400, backups preference 700.  
As long as the primary member is not SYSGONE or HARDDOWN and its system is not being stopped it remains selected. If it is HARDDOWN or SYSGONE or its system is being stopped its preference is between 2600 and 2000, so a viable alternative can be selected.  
A shutdown of the primary system will initiate a move of the application, as it will have lost its 225 point bonus for being available and have incurred a 400 point penalty, bringing its preference down to well below 2600, allowing the selection of an alternative resource if there is a viable one. |
### Table 6. Preference Value Scenarios (continued)

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sticky move if system down or being stopped, move after application failure</td>
<td>Primary member defined with preference 2401, backups preference 2400. As long as the primary is not SYSGONE or HARDDOWN and its system is not being stopped it remains selected. If it is SYSGONE or HARDDOWN or its system is being stopped its preference is between 2000-2600, so a viable alternative may be chosen. The sticky bonus raises the preference of the selected member to 2575 (2400 + 175), so it will remain selected until it gets to a state of SYSGONE or HARDDOWN or its system is being stopped, at which point another viable alternative will be selected, with the primary member having preference. The individual shutdown of the active member will not trigger a move. A shutdown of the primary system will result in a move of the application. It will have a 400 point penalty, dropping its preference to between 2600 and 2000, permitting the selection of a viable alternative.</td>
</tr>
<tr>
<td>Sticky move group, unless member stopped</td>
<td>Primary member defined with preference 701, backups preference 700. As long as the primary member is startable, it remains selected. If it becomes non-startable, another member is selected and that remains selected as long as it is startable. Individually stopping the active member will trigger a move. If the members were defined with preferences &gt;1500, the bonus would be 175 points rather than 25 points.</td>
</tr>
<tr>
<td>Sticky move group, until told otherwise</td>
<td>Primary member defined with preference 3200, backups with preference 1600. With base preferences of 3200, you get it picking a member when the sysplex is IPLd and retaining that one until an operator uses INGGROUP to make it pick another one. Beware of timing problems as the first system with a member to register will probably get it.</td>
</tr>
</tbody>
</table>

### GENERATED RESOURCES Policy Item

The Generated Resources panel appears if you select the GENERATED RESOURCES policy item from the Policy Selection panel for ApplicationGroups. This panel shows all resources that are generated by SA z/OS for the application group itself and all the resources for the applications that are members of this application group. This panel is for information only. No actions are allowed.
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.

### MEMBER OF Policy Item

The Member of panel appears if you select the MEMBER OF policy item from the Policy Selection panel for application groups.

**Figure 92. Generated Resources Panel**

**Figure 93. Member of Panel for ApplicationGroups**

This panel lists all resources where this ApplicationGroup is a member "resource-wise", that is, it lists all relations from resources that have been derived from the current ApplicationGroup to other resources. This panel is for information only. No data can be changed.

A Member of relation can be added or removed in the RESOURCES policy item of the entry that is given in the "is a member of" column (that is, in the RESOURCES policy of the APG Entry Name that the "member of" resource was derived from).
MEMBER OF Policy Item for ApplicationGroups

If the owning APG resource (given is a member of) is of nature SERVER or MOVE, the Preference column 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, the Preference column shows "SELECTED". For more details about available preference values see “Creating a New ApplicationGroup” on page 117.

In the example in Figure 93 on page 139, the resource NET/APG/SYSTEM1 is a member of resource WBSRVAPG/APG.

Application Entry Type

An application is a z/OS subsystem, started task, application, batch job, or non-MVS resource that runs on a system in the SA z/OS enterprise. SA z/OS monitors applications through the messages they issue to the operator. The term application is also used to refer to a subsystem or process on an SNMP-capable machine. Note that SA z/OS 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 z/OS ensures that the application is defined only once in the automation control file.

The SA z/OS customization dialog allows concurrent user access capability for the Application entry type. See Appendix A, “Concurrent Customization by Multiple Users,” on page 357 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 19.

Figure 94 on page 141 shows the panel that is displayed when you create a new application from the Entry Name Selection panel for applications.
Creating a New Application

When you define a new application, you provide the following information:

**Entry 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 assign the right characteristics of an application to a specific system. The first character of the application name may not be numeric.

**Subsystem Name**

The subsystem name is required. It defaults to the application's entry name if the entry name complies with the restrictions for subsystem names (see below). Press Enter to have the default displayed in the field.

The subsystem name provides a unique, eleven-character name for the Application that you are defining. This name is used by SA z/OS automation. You can have multiple application instances with the same subsystem name, however, you should take care to ensure that only one of them is linked to any given system. The subsystem name of an application class must be unique across the whole policy database, that is, it cannot be the same as the subsystem name of any other application instance or class.

Note that the subsystem name of an existing application can be modified by overtyping the **Subsystem Name** field in the Application Information policy panel.

The restrictions on the subsystem name are:

- The length is limited to eleven characters.
- Valid characters are any alphanumeric character or any of @, #, $, ? or _ (underscore), with the first character being alphanumeric or one of @, #, $ or ?.
Creating a New Application

- See the online help for a list of values that are reserved for SA z/OS and cannot be used.

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.

For applications of type IMAGE, the same value is required for the subsystem and the job name. Therefore the subsystem name must comply with the restrictions for job names (see "Job Name" on page 144).

**Object Type**

**CLASS**

If the entry represents common policy that is to be inherited by one or more other application entries that are defined as INSTANCES, select this object type. CLASSes are used only to define inherited policy; they do not represent actual applications that are managed by SA z/OS.

**INSTANCE**

If the entry represents an application that is to be managed by SA z/OS, select this object type. If you link an INSTANCE to a CLASS you can omit policy items from the INSTANCE, which allows that policy to be inherited from the specified CLASS.

Use of the CLASS object type allows you to define a common policy for multiple applications. This is advisable if these applications share common characteristics and functions. It decreases effort during:

- Initial definition
- Administrative changes (using the customization dialog)
- Run time changes (using the SA z/OS automation engine)

An application instance can be linked to a class using the UPWARD CLASS policy item. If a policy item is not specified for the instance, it is inherited from the class. Conversely, a policy item that is 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 takes precedence over those at class and default level. Likewise a class definition takes precedence over a default. The definitions for a specific flag type are not merged.

The inheritance of policy from an application class occurs during SA z/OS initialization. If an application policy item is not specified in either the application instance or the class, defaults are used from the application defaults, system defaults, and environment setup objects. Some policy items are required to be specified for an application instance. Except for job name, these same items need to be specified for an application class, but note they will never be inherited.

SA z/OS supports run time changes to an 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 not required. You may specify one of the following values:
Creating a New Application

- JES2 if the application represents a JES2 application, see "Policy Items for JES2 Subsystems" on page 197.
- JES3 if the application represents a JES3 application, see "Policy Items for JES3 Subsystems" on page 201.
- DB2 if the application represents a DB2 application, see "Policy Items for DB2 Subsystems" on page 204.
- CICS if the application represents a CICS application, see "Policy Items for CICS Subsystems" on page 203.
- IMS if the application represents an IMS application, see "Policy Items for IMS Subsystems" on page 207.
- OPC if the application represents a TWS application, see "Policy Items for TWS Subsystems" on page 208.
- USS if the application represents an USS application, see "Policy Items for USS Subsystems" on page 210.
- IMAGE if the application represents a system image (to enable enhanced Parallel Sysplex functions). For details, see IBM Tivoli System Automation for z/OS Customizing and Programming.
- TCPIP if the application represents a TCPIP application.
- INFOSPHERE if the application represents an INFOSPHERE application.
- LIFELINE if the application represents a LIFELINE application.

If you specify one of these application types, except for CICS, additional policy options specific to their automation are displayed on the SA z/OS Policy Selection panel. Note that once you have specified the value and created the Application entry, the value cannot be changed. Only the blank value can be changed afterwards.

Subtype

Specifies the region subtype for CICS, DB2 and IMS or the controller for OPC, or the subtype for LIFELINE. The following values may be specified:

- **CICS**: AOR, TOR, FOR, DOR, QOR, CMAS, WUI
- **DB2**: MSTR, SPAS, IRLM, DBM1, DIST, WLMS
- **IMS**: CTL, TP, DBRC, DLS, FP, BMP, FDR
- **OPC**: CONTROLLER, TRACKER, SERVER, DATASERVER
- **LIFELINE**: ADVISOR AGENT

For applications of type JES2, JES3, USS, TCPIP and IMAGE it is forced to blank.

For applications, where INFOSPHERE or no type has been specified it can be any value containing alphabetic and numeric characters.

Job Type

This can be one of:

- **MVS** This is the standard MVS job type that SA z/OS normally handles. It is typically monitored using the ASCB checker. This job type can be specified explicitly with JOBTYPE=MVS for an application, however, it is the assumed default if the Job Type entry is omitted from the definition of the application.

- **NONMVS** This is a type of resource that may be started by SA z/OS but which cannot be monitored using an ASCB check or the MVS D A method.
Creating a New Application

If the AOFATMON routine cannot be used to monitor this type of resource, either a user-provided application monitor routine is required or the message automation table must be correctly set up to maintain the job status accurately. This status is the only way of monitoring this class of job. NONMVS jobs must have STARTUP and SHUTDOWN commands defined.

**TRANSIENT**

This is a type of job that may be started by SA z/OS but terminates without the intervention of SA z/OS. It is a short running job that will generally, but not necessarily, terminate before SA z/OS does. It can be monitored using an ASCB check.

As far as monitoring is concerned, a TRANSIENT job is treated as a z/OS job, but it has special properties: transient jobs must START and END before their dependent applications are started.

You can specify shutdown passes for transient jobs if there is a possibility that they may need to be shut down by SA z/OS. These passes are optional.

**Job Name**

The job name is required. It defaults to the subsystem name if this complies with the restrictions on job names. Press Enter to have the default displayed in the field. For applications of type IMAGE, the job name is forced to the subsystem name. Thus, for applications of type IMAGE, the subsystem name must comply with the restrictions on job names.

The job name gives a unique name to the application to be run. There is a one-to-one mapping between the MVS job name and the subsystem name. SA z/OS automation interprets messages from MVS by taking the MVS job name and mapping it to the subsystem name.

You can have multiple applications with the same job name but you must ensure that only one of them is linked to a particular system. Duplicates are ignored by the BUILD function.

The restrictions on the job name are:

- It can be up to eight alphanumeric and national characters, including @, #, and $, after clone resolution, provided the first character is not numeric.
- If the job type is NONMVS, you can also use _ and ?.
- You can include system symbols and system automation symbols.
- The special value *MASTER* is allowed for applications of type blank. If the object type is CLASS, a job name is not allowed.

For details about creating the start command, see “JCL Procedure Name” on page 145.

**Transient Rerun**

This field specifies whether a TRANSIENT application may be rerun within the lifetime of z/OS. The Transient Rerun option involves the concept of a one-shot transient job.

- **YES** This indicates that the job will be rerun as appropriate. SETSTATE can 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** This indicates that the job will only be run once in the lifecycle of a particular z/OS. A value of NO affects the processing of the SETSTATE command. NO is the default value for transient jobs.
Creating a New Application

**Note:**
On an initial automation policy load, such as at IPL, 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 a one-shot transient is only allowed to run once in the lifecycle of a particular z/OS.

A one-shot job remains in an ENDED state across a SA z/OS recycle.

There are some situations when even one-shot transient jobs will be rerun:

- If the automation status file is deleted and NetView is recycled, a one-shot transient job will be restarted.
- If a job is deleted from the automation policy, status information for that job is deleted during a refresh of the policy. If a later refresh is used to redefine the application to automation, the application status is set to DOWN. This may also result in the restart of a one-shot transient job being restarted.

**Scheduling Subsystem**
This is optional. It is the name of the subsystem that is to schedule this application, as follows:

- **MSTR**
  - The application is scheduled directly by the MVS/ESA master scheduler.

- **other**
  - The application is scheduled by another scheduling subsystem. The value entered is the name of that subsystem. The subsystem that is specified must be defined as type JES2 or JES3.

If you leave this field blank, it indicates that the application is scheduled by the primary JES.

For applications of type IMAGE the value is forced to blank.

**JCL Procedure Name**
The JCL Procedure Name is optional. It specifies the name of the JCL procedure that is used to create this instance of the application. If specified, the default start command for the application is:

```mvs
S proclname,JOBNAME=jobname
```

Where `proclname` is the JCL Procedure Name and `jobname` is the Job Name.

If you do not specify a JCL procedure name, the default start command is:

```mvs
S jobname
```

You can include system symbols and system automation symbols (see "Assigning System Automation Symbols (AOCCLONE)" on page 8).

For applications of type IMAGE the value is forced to blank.

**Policy Items for Applications**
Figure 95 on page 146 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 via the application or system defaults. For example, Thresholds can be set to a default value for all applications. If the definition of an application has no value specified for Thresholds, the value specified in the Application Defaults is used. If no value is specified in the Application Defaults any value specified in System Defaults is used. See also “Application Defaults Entry Type” on page 271 and “System Defaults Entry Type” on page 268.

The unique policy items for Applications are:

**UPWARD CLASS**
This field is displayed if the application was defined as an INSTANCE or CLASS (see “Creating a New Application” on page 140). It lists all entries that have been defined as resource classes and allows you to select one class that this application should be linked to so that it inherits the policy values of that class or class chain.

See “UPWARD CLASS Policy Item” on page 149 for more information.

**DOWNWARD CLASS/INST**
This policy item is displayed if the application was defined as a CLASS (see “Creating a New Application” on page 140) and it lists all class or instance entries. You can add or remove classes or instances linked to this class.

See “DOWNWARD CLASS/INST Policy Item” on page 150 for more information.

**APPLICATION INFO**
This policy item displays the Application Information panel that shows the policy for the current application as entered when it was created. It is also used to define the application to SA z/OS automation. You can edit part of the policy information on this panel. The fields that contain policy...
Policy Items for Applications

information that you must not modify are read-only. To change read-only information, you must delete and recreate the application.

See “APPLICATION INFO Policy Item” on page 152 for more information.

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, see “AUTOMATION FLAGS Policy Item” on page 160 and the section “Automation Flags” in IBM Tivoli System Automation for z/OS User’s Guide.

APPLICATION SYMBOLS
Application symbols allow you to enter or update application symbols that can be set to various application-specific values. These can then be substituted using the runtime variables SUBSSYMBOL1 through SUBSSYMBOL9. Those variables can be used in a similar fashion to runtime variables like SUBSSPARM, SUBSSCHEDSS, SUBSJOB, etc.

For more information, see “APPLICATION SYMBOLS Policy Item” on page 162.

TRIGGER
Each application may be linked to one trigger. This policy item allows you to link the current application to a trigger. Only one trigger can be selected.

Triggers are used to control the starting and stopping of applications in a single system or a Parallel Sysplex. They determine the behavior of the requested action. For an application to be started, its desired status must be AVAILABLE and its startup trigger must be satisfied.

For an application to be stopped, its desired status must be UNAVAILABLE and its shutdown trigger must be satisfied.

For more information on how to use triggers, see “TRIGGER Policy Item” on page 163 and the section “Event and Trigger Support” in IBM Tivoli System Automation for z/OS User’s Guide.

SERVICE PERIOD
Service periods allow users to schedule the availability of applications, application groups, or monitor resources. With this policy item you can link a service period to the application so that it is automatically started up or shut down in the time frame specified with the service period.

For more information, see “SERVICE PERIOD Policy Item” on page 163.

RELATIONSHIPS
Relationships are used to define dependencies between resources that are used to perform in a controlled way startup or shutdown actions of dependent and supporting applications, application groups, and monitor resources.

This policy item has the same functionality for entry types Application, ApplicationGroup, and Monitor Resource. For each application, application
group, or monitor resource you may specify several relationships. If you select this policy item, in the upcoming Relationship Selection List panel, you can view existing or define new relationships for the current application, application group, or monitor resource.

You define relationships between dependent resources and supporting resources. In the customization dialog you define relationships always from the point of view of the dependent resource. So the resource that you have currently selected will become the dependent resource, when you define a new relationship.

See “RELATIONSHIPS Policy Item” on page 126 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 the command of your choice.

How to proceed in specifying this policy item is described in “MESSAGES/USER DATA Policy Item” on page 165.

STARTUP
This policy item allows you to add, remove or modify startup procedures for the current application.

See “STARTUP Policy Item” on page 181 for more information how to specify the startup automation policy.

SHUTDOWN
You must specify NORMAL shutdown commands for any application that you want to automate with SA z/OS. You may also specify INIT, IMMED, FORCE, and FINAL shutdown commands.

Specifying shutdown commands or replies for the application allows the automation to perform application shutdowns. You can specify different shutdown commands or replies to be issued for the different types of shutdown situations.

How to achieve this is described in “SHUTDOWN Policy Item” on page 185.

THRESHOLDS
Error threshold definitions for an application define how many abend errors can occur before a message is logged or restart processing is stopped. If no error thresholds are set here, the application uses the thresholds that are defined in the Application Defaults entry type.

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 the section “How SA z/OS Uses Error Thresholds” in IBM Tivoli System Automation for z/OS User's Guide and “THRESHOLDS Policy Item” on page 192.

MINOR RESOURCES
This policy item allows you to define flags and thresholds for minor
Policy Items for Applications

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

See the description of the MINOR RESOURCES policy item in "MVS Component Entry Type" on page 263 for information on how to manage minor resources and "MINOR RESOURCES Policy Item" on page 193.

The automation flags that are associated with minor resources are sometimes called extended automation flags. See "Automation Flags and Minor Resource Flags" in [IBM Tivoli System Automation for z/OS User’s Guide](#) for information about them.

See also the section "Thresholds for Minor Resources" in [IBM Tivoli System Automation for z/OS User’s Guide](#).

SYSTEM ASSOCIATION
This policy item allows you to define primary and secondary systems for the application. Primary and secondary categorizations help SA z/OS determine whether an application should be started by SA z/OS when SA z/OS is initialized on a system. These categories allow SA z/OS to perform automation, such as monitoring or shut down, on applications that can be moved by an external agent.

For a detailed description on how to define system categories with this option see "SYSTEM ASSOCIATION Policy Item" on page 195.

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 92 on page 139](#) except that the section, Resources for THIS entry, showing the application group resource is missing.

MEMBER OF
This policy item lets you see the APG Resources that the current application is a member of.

See "MEMBER OF Policy Item” on page 139 for more details.

UPWARD CLASS Policy Item
If you select the UPWARD CLASS policy item from the Policy Selection panel for Applications, the Upward Link to Class panel is displayed, as shown in [Figure 96 on page 150](#)
This panel allows you to manage the upward link from an application class or instance to a class of the same type. This link allows the instances or classes to inherit any policy data that has been defined in the selected class or other classes further up in the class chain.

- Use the Action field to manage the link by entering:
  - Add (A) or Select (S) to link the class or instance to this class
  - Remove (M) to unlink the class or instance from this class

- The Status column indicates whether this class or instance has been linked up to the class in the list.

- The Type column indicates what application type the class in the list was defined as.

- The Upward Class Link column indicates the class, if any, that the entry in the list is currently upwardly linked to.

To avoid circular links, a check is made to ensure that the class you select is not already linked in the downward tree of the current class.

See “Nested Classes” on page 11 for more details about nested class support.

**DOWNWARD CLASS/INST Policy Item**

If you select the DOWNWARD CLASS/INST policy item from the Policy Selection panel for Applications, the Downward Link to Classes or Instances panel is displayed, as shown in Figure 97 on page 151.
This panel allows you to manage the downward links from an application class to classes or instances of the same type. These links allow the instances or classes to inherit any policy data that has been defined in this class or other classes further up in the class chain.

- Use the **Action** field to manage the links by entering:
  - Add (A) or Select (S) to link the class or instance to this class
  - Remove (M) to unlink the class or instance from this class

- The **Status** column indicates whether the entry in the list has been linked down to by this class.

- The **C** column indicates with an asterisk (*) that the entry is a class because this may not be obvious from the entry name.

- The **Type** column indicates what application type the entry was defined as. Note that an application class that you are working with can only be linked to entries of the same type as itself.

- The **Upward Class Link** column indicates the class, if any, that the entry in the list is currently upwardly linked to. If you select such an entry, you would change its class chain, so be careful not to break its data inheritance.

To avoid circular links, a check is made to ensure that the class you select is not already linked in the upward tree of the current class.

Both link panels show the application type because class-instance links are only allowed if there is no type mismatch in the class hierarchy, that is, either the specified types must be the same or no type is specified.

To avoid type mismatches, the types of the link partner or the inherited types are checked. A link is invalid if the resulting class chain includes type specifications that do not match (for example, a class or instance of type IMS and another class or instance of type CICS). Thus, if both link partners (the current entry and the linked class or instance) have a type specified it is checked that they are identical. If either link partner has no type specified, the class hierarchy that leads to this entry is scanned to check whether an inherited type may cause a mismatch. If this is the case, a detailed link error message that includes the names of the conflicting entries is displayed.
APPLICATION INFO Policy Item for Applications

See "Nested Classes" on page 11 for more details about nested class support.

APPLICATION INFO Policy Item

If you select the APPLICATION INFO policy item from the Policy Selection panel for Applications, the Application Information panel is displayed, as shown in Figure 98.

Any policy values that have been inherited from linked classes are displayed in a different color. You can find out which class a value has been inherited from by typing the command `classname` (or its abbreviation `cn`) on the command line, moving the cursor to the field of interest and pressing Enter. You can override any inherited values by overtyping them. If you override any values for a class, they are individually applied to the current application and change color to turquoise to indicate this.
APPLICATION INFO Policy Item for Applications

This panel displays the following information:

**Application Type**
See “Application Type” on page 142 for more details.

**Subtype**
See “Subtype” on page 143 for more details.

**Subsystem Name**
This gives a unique, eleven-character name to the application that you are defining. This name is used by SA z/OS automation. You can have multiple application instances with the same subsystem name, however, only one of them can be linked to a system. The subsystem name of an application class must be unique across the whole policy database, that is, it cannot be the same as the subsystem name of any other application instance or class.

You can change the subsystem name by overwriting the existing name. If the application instance is linked, the name is checked to be unique across all subsystem names in the policy database. If you want to change the subsystem name to a name that already exists, you first need to un-link the application instance from all application groups.

If no subsystem name is specified, the default is the value of the Entry name (if this name is not longer than 8 characters for applications of type IMAGE, and not longer than 11 characters for all other application types).

For applications of type IMAGE the subsystem name can only be an 8-character name.

**Note:** Certain values are reserved for SA z/OS and cannot be used. See the online help for a complete list.

**Job Type**
See “Job Type” on page 143 for more details.

**Job Name**
See “Job Name” on page 144 for more details.

**Transient Rerun**
See “Transient Rerun” on page 144 for more details.

**Scheduling Subsystem**
See “Scheduling Subsystem” on page 145 for more details.

**JCL Procedure Name**
See “JCL Procedure Name” on page 145 for more details.

**Job Log Monitor Interval**
Specifies how frequently the Job Log should be monitored. Frequency must be specified using format mm:ss. The allowed minimum is 1 second (00:01), maximum is 60 minutes (60:00). If the field is left blank the value can be inherited from a class definition. If monitoring is not required, specify NONE, to prevent inheritance.

**Note:** If nothing is specified or inherited no monitoring will be done. If scheduling subsystem is MSTR, the specification is ignored.

**Captured Messages Limit**
Specifies the maximum number of captured messages that will be saved for the SA z/OS command DISPINFO.
APPLICATION INFO Policy Item for Applications

If the field is left blank the value can be inherited from a class, the application defaults (ADF), or the system’s default definition (SDF). If neither of these are specified a default value of 0 is used.

Desired Available
Let you specify the default desired status of the resource. The desired state of each resource is either Available or Unavailable, which is the goal that automation tries to achieve.

You can specify the following values:

ALWAYS
The resource’s desired status is set to Available, unless it is dependent on a resource that has a Desired Available setting of ONDEMAND. In this case the resource behaves as if it had a Desired Available setting of ONDEMAND itself.

This is the same behavior as in earlier releases of SA z/OS.

ONDEMAND
If there is demand for the resource to be available, its desired status is set to Available, otherwise its desired status is set to Unavailable.

Demand arises either from propagated MakeAvailable votes or implicitly through membership of a non-passive basic application group (APG) that has a desired status of Available. Demand does not arise from dependent resources with a Desired Available setting of ALWAYS.

A MakeAvailable vote that is propagated to the resource overrides any demand considerations.

An active ONDEMAND member of a move or server group is always sent a vote that sets its desired status, thus overriding any demand considerations.

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

If you leave the field blank the value can be inherited from a class or the system defaults definition.

Note: If nothing is specified or inherited the default value is ALWAYS.

Restart after IPL
Specifies how SA z/OS determines the initial status of the application on the first occasion that SA z/OS is started after a z/OS IPL or an automation status file reallocation.

Reallocating the automation status file has the same effect as a z/OS IPL because SA z/OS stores the last IPL date and time in the automation status file. When the automation status file is deleted, SA z/OS assumes that the next startup is the first startup following a z/OS IPL.

You can specify four values for Restart after IPL:

START
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
APPLICATION INFO Policy Item for Applications

applications such as JES, VTAM, and TSO. Replying NOSTART to message AOF603D (specify initialization options) overrides this setting.

NOSTART
This enforces a STOP request with priority FORCE to be injected during SA z/OS initialization. This results in a desired status of UNAVAILABLE and an observed status of SOFTDOWN, which produces a compound status of SATISFACTORY. If the observed status before IPL was HARDDOWN this status remains in HARDDOWN, which produces a compound status of PROBLEM.

NONE
This value is the same as leaving the field blank, except that if the application is an instance, it does not inherit the value from its related application class.

If you leave this field blank, it indicates that the application is startable, depending on its status. If the status is STOPPED, BROKEN or CTLDOWN, it is not changed and the application is not started. Any transient status (for example, BREAKING) is changed to the corresponding final status (for example, BROKEN). Any other status is changed to DOWN during initialization and, if the INITSTART flag is set to YES, the application is started when all its relationships are fulfilled.

Start Delay
Specifies an interval which begins when a start command for the application is issued by SA z/OS. After the expiration of this interval, a check occurs to determine whether the application recorded a status of STARTED and/or ACTIVE depending on the MVS active/inactive state at that time. Message AOF 571I is issued.

Note: If nothing is specified the default supplied by the automation agent is used, which is 2 minutes.

Start Cycles
Specifies the number of times to cycle the start timeout period before posting the subsystem as a problem which means the application is given a STARTED2 or INACTIVE status.

Note: If nothing is specified the default value is 1.

Restart Option
The circumstances that the application should be restarted under. The options are:

Always
The application is restarted if it was terminated without using the INGREQ command. Note that, if the advanced automation option AOFRESTARTALWAYS is set to 0, the application restart thresholds are checked. The application is restarted only if it has not exceeded the critical threshold. By default the application is restarted every time it is stopped outside the control of SA z/OS.

Abendonly
The application is eligible for restart only if it abnormally ends. The application is restarted only if its critical error threshold has not been exceeded.

Never
The application is never eligible for restart by SA z/OS.
You can use this option to force a specific application state. For example, if an application is defined with the restart option set to Always and an operator tries to shut down the application without using the INGREQ command, the application will be restarted when it has completed its shutdown. The AOFRESTARTALWAYS advanced automation option can be used to modify this behavior.

**External Startup**

Specifies whether the application is started externally or via a specified startup procedure.

- **INITIAL**
  The application will only initially be started externally.

- **ALWAYS**
  The application will always be started externally.

- **NEVER**
  The application will never be started externally.

If you leave this field blank, the value can be inherited from a class definition. For applications of type IMAGE where the monitor routine is INGMTSYS the value is forced to ALWAYS.

**Note:** If nothing is specified or inherited the default supplied by the automation agent is used, which is NEVER.

**Skip ACTIVE status**

Specifies whether the application has a specific indication for the UP status.

- **YES**
  Sets the status to UP for standard applications if message IEF403I is trapped. For transient applications the status is changed from ACTIVE to RUNNING.

  For USS applications with a job type of NONMVS it also sets the status to UP if the USS process exists.

- **NO or blank**
  It does not automatically set the status.

**Startup Parameters**

Specifies the desired subsystem startup parameters. These values are added to the MVS START command for the subsystem. For example, suppose the desired start command for VTAM is:

```
S VTAM,,,(LIST=00)
```

If this command is to be submitted from SA z/OS, the **Startup Parameters** field should contain:

```
,,,(LIST=00)
```

Whatever you enter in this field is appended directly to the MVS START command for the subsystem, therefore, the commas preceding (LIST=00) are necessary.

You can include system symbols and system automation symbols. See “Assigning System Automation Symbols (AOCCLONE)” on page 8 for details on using system automation symbols.
APPLICATION INFO Policy Item for Applications

You may also include run time variables, such as SUBSAPPL, SUBSJOB, SUBSPROC, SUBSSHUTDLY and others. For a complete list of run time variables, see SA z/OS documentation.

External Shutdown
Specifies whether the application is stopped externally or via a specified shutdown procedure.

FINAL
The application will be stopped externally if the external supporting resource (parent) is also being stopped.

ALWAYS
The application will always be stopped externally.

NEVER
The application will never be stopped externally.

If you this field is left blank, the value can be inherited from a class definition. For applications of type IMAGE where the monitor routine is INGMTSYS, the value is forced to ALWAYS.

Note: If nothing is specified or inherited, the default supplied by the automation agent is used, which is NEVER.

Shutdown Pass Interval
Used to tell SA z/OS how long to wait for attempts to shutdown the subsystem. These attempts are defined in the SHUTDOWN policy. If the field is left blank, the value can be inherited from a class or the application defaults definition.

Note: If nothing is specified or inherited, the default supplied by the automation agent is used, which is 2 minutes.

Cleanup Delay
Used to specify an interval which begins when the final termination message (TERMMSG FINAL=YES) for the subsystem is processed. The purpose of this delay is to allow for additional processing performed by or on behalf of the subsystem to the message being issued. If the field is left blank the value can be inherited from a class or the application defaults definition.

Note: If nothing is specified or inherited the default supplied by the automation agent is used, which is 12 seconds.

Command Prefix
This is the console prefix character for the application. Many applications, such as JES2, JES3, and NetView support the use of a command prefix character which identifies a command as belonging to that application. If such support is available for this application, enter the corresponding command character.

Any command that would normally be issued at a z/OS console should be prefixed with MVS. NetView commands to control NetView automation do not require the MVS prefix.

You can include system symbols and system automation symbols, see "Assigning System Automation Symbols (AOCCLONE)" on page 8.

Note: The single quotation character (’) is reserved for SA z/OS and should not be used.
APPLICATION INFO Policy Item for Applications

Message Prefix
Specifies one or more message prefixes for this application. You only have to specify these message prefixes if the application issues messages that do not contain the MVS job name. These message prefixes must be unique for this application.

Sysname
The system name for the subsystem. This is only required for JES subsystems where the Sysname for the subsystem is not the same as the MVS Sysname. If Sysname is not specified, it will default to the name specified in the SYSTEM INFO policy item of the System entry type.

Monitor Routine
The name of the application monitor routine that is used to monitor the application's status.

The following specifications are valid:

**AOFADMON**
Monitors the status via MVS D A commands.

**AOFAPMON**
Routine used to monitor a PPI receiver.

**AOFATMON**
Monitors the status of a task in the NetView environment.

**AOFCPSTM**
Dedicated routine used to monitor SA z/OS Processor Operations.

**AOFNCMON**
Routine used to monitor the connection between the NMC focal point NetView and the NMC server application.

**AOFUXMON**
Routine used to monitor USS applications.

**INGMTSYS**
Dedicated routine used to monitor IMAGE applications for BCPII usage.

**INGROMON**
Dedicated routine used to monitor OMVS which should not be used for any other application.

**ISQMTSYS**
Dedicated routine used to monitor SA z/OS processor operations target system resources.

**INGPJMON**
Monitors the status via ASCB checking.

**INGPSMON**
Monitors the status via IEFSSI service.

**INGVMON**
Monitors the status of a VOST via LIST STATUS=VOST.

**NONE**
No monitoring routine is to be used.

**username**
A monitoring routine that has been defined and created by the user.
APPLICATION INFO Policy Item for Applications

Note: If a user-defined routine is not found by automation, the SA z/OS routine INGPJMONT is used except for applications of type IMAGE, when INGMTSYS is the default. For non-MVS applications, make sure that either a user-defined monitoring routine or NONE is specified.

Monitor Interval
This is the amount of time between application subsystem monitoring cycles. Monitoring may be suspended by specifying NONE.

Inform List
This field allows you to specify:
- Where the application is registered to
- Where all status changes are propagated to
- Whether SMF records are written

Valid receivers are SDF, NMC, IOM, SMF, EIF, TTT and USR. You can specify more than one receiver.

The list of values can be blank or comma separated. Neither system automation symbols (AOCCLONEs) nor system symbols are allowed.

If the field is left blank the value is obtained from the systems default definition (see “System Defaults Entry Type” on page 268).

Specify NONE to prevent the default from the systems default definition. This prevents inheritance from the system's default definitions (SDF).

ARM Element Name
An application defined to MVS Automatic Restart Manager must have an element name that is unique within each sysplex where the application runs. Each element name consists of up to 16 alphanumeric and national characters including @,#,$ and _ (underscore) after clone resolution, provided the first character is not numeric. Clone resolution occurs at automation control file load.

You can include system symbols and system automation symbols. For more details, see “Assigning System Automation Symbols (AOCCLONE)” on page 8.

The element name can be:

Constant
The element name is the same, regardless of which system the application is running on. This is required for applications that are restartable on multiple systems in the sysplex. An application with a constant Automatic Restart Manager element name may at the same time have a cloned job name.

Note: An application with a cloned job name and a constant Automatic Restart Manager element name can be started on only one system within a sysplex.

For applications of type IMAGE the value is forced to blank.

Cloned
The application has a different element name on each system. The element name is derived from some attribute that is unique to the system that the application runs on. Applications with cloned element names must be defined to MVS Automatic Restart
Manager in such a way that MVS will only attempt to restart them on the same system and not move them to other systems in the sysplex.

Applications with cloned element names may have constant job names. This can be useful if you want to have the same application running on multiple systems with Automatic Restart Manager recovery enabled.

For example, you have a USERPROC that runs on all your systems and you want Automatic Restart Manager recovery for it. Automatic restart management requires that USERPROC have a different element name on each system. Defining USERPROC to SA z/OS with a constant job name, but with a cloned element name (and perhaps some parameters to define the element name) allows you to define a single SA z/OS application with a different element name on each system.

**WLM Resource Name 1–3**

These are the names of the WLM Resources that are associated with the application.

The same WLM resource name can be used for multiple applications. If any of these applications are in the UP or ENDED automation status, the WLM resource status is set to ON. When all the applications that are associated with the same WLM resource name are in any other status, the resource is set to OFF.

You can include system symbols and system automation symbols, see ["Assigning System Automation Symbols (AOCCLONE)" on page 8](#).

You can specify up to three different WLM resource names, but only one per field.

No WLM resource names are accepted for applications of type IMAGE.

All three WLM resource names are inherited together. Thus if you specify one for an instance and another for a class, this is not inherited.

**Owner**

This specifies information for the operator about who to contact in case of error.

**Info Link**

This field can be used to specify a location (for example, a URL) where additional information about the application can be found.

**Runtokens**

This field can be used to specify one or more blank delimited tokens (each up to 20 characters long) to define a runmode qualification for the application.

**AUTOMATION FLAGS Policy Item**

The Flag Automation Specification panel (Figure 99 on page 161) is displayed if you select policy item AUTOMATION FLAGS from the Policy Selection panel for applications. For more details about automation flags, see the section “Automation Flags” in [IBM Tivoli System Automation for z/OS User’s Guide](#).
This panel allows you to specify:

- whether automation should not occur in certain states
- whether automated operator functions will operate in log assist mode
- whether user-provided exit routines are to be called to determine whether automation is to continue
- whether automation should be disabled during certain time intervals

There are six types of automation flags supported by SA z/OS:

**Automation**
Is the global automation flag that must be YES for the remaining flags to be used. If it is NO, there is no automation for the resource.

**Recovery**
Indicates whether recovery of the resource (actions on messages that are neither for the initialization nor the ending of the resource) is allowed.

**Start**
Indicates whether the start command and actions or responses to initialization for the resource are allowed.

**Terminate**
Indicates whether the shutdown command and actions on or responses to ending messages are allowed.

**InitStart**
Indicates whether a resource can be started when its state is DOWN (that is, it controls initial starting of the resource).

**Restart**
Indicates whether the restarting of a resource that abnormally ended or was automatically shut down is allowed.

For each flag you may specify:

**Cmd**
Select one of the following values:

- **EXI(E)** permits specification of automation flag exit names and descriptions.
- **DIS(D)** permits specification of automation disable times.
AUTOMATION FLAGS Policy Item for Applications

CN the name of the class where data are inherited from (Application Entry Type only).

Auto Use this field to control what particular automated responses are allowed:
YES Automation is active.
NO Automation is not active.
LOG LOG assist mode is active. Actions that should be taken under the chosen type of automation do not occur, but messages describing what would have happened are written to the NetView log.

EXITS One or more user-provided exit routines have been defined.

The Exits and DisableTimes fields display the number of exit routines and disable time intervals which have been specified for the particular automation flag. If the data is defined on a linked class then the numbers are shown in the color of the inherited data.

For all flags, if EXITS is specified as a value, the defined exits are called to determine if automation is active.

APPLICATION SYMBOLS Policy Item

If you select the APPLICATION SYMBOLS policy item from the Policy Selection panel for Applications, the Application Symbols panel is displayed, as shown in Figure 100.

This panel allows you to enter or update application symbols that can be set to various application-specific values. These can then be substituted using the runtime variables SUBSSymbol1 through SUBSSymbol9. Those variables can be used in a similar fashion to runtime variables like SUBSPARM, SUBSSCHEDSS, SUBSSJOB, etc.

To specify that SUBSSymbol value x is to be substituted, enter &SUBSSymbolx at the appropriate point in the field. You must specify only the leading ampersand (&).

SYMBOLx Allows you to specify the value of the application symbol. You can enter
**APPLICATION SYMBOLS Policy Item for Applications**

any character but you should avoid using any SA z/OS reserved runtime variables, such as &SUBSJOB or others.

**Note:**
The entered data is not automatically translated to upper case.

**Description**
Allows you to specify a description of the application symbol.

**TRIGGER Policy Item**
The Trigger for Application panel [Figure 101] is displayed if you select policy item TRIGGER from the Policy Selection panel for applications. For more details about triggers, see the section “Event and Trigger Support” in [IBM Tivoli System Automation for z/OS User’s Guide](#).

This panel allows you to add triggers to or remove them from an application.

**SERVICE PERIOD Policy Item**
If you select an application’s SERVICE PERIOD policy item, you reach the Service Period for Application panel. Figure 102 shows an example.

This panel allows you to add service periods to or remove them from an application.
RELATIONSHIPS Policy Item

RELATIONSHIPS Policy Item

The Relationship Selection List panel (Figure 88 on page 127) is displayed if you select policy item RELATIONSHIPS from the Policy Selection panel for applications. It shows information about existing relationships and allows you to define new relationships.

The ACTION pull-down choice offers actions to edit and delete selected relationships. The Delete command immediately deletes the relationship. The Edit command leads to the Define Relationship panel (as shown in Figure 103).

This panel displays information about existing relationships. In all of the relationships for the current resource, this resource is the dependent resource. The columns have the same meaning as described in “Defining Relationships” on page 127.

If the application is linked to a class, the External Startup and External Shutdown fields in the subheader area show inherited data individually, however the relationships are inherited only as a whole if no relationships are defined for the child object.

You can find out which class a value has been inherited from by typing the command classname (or its abbreviation cn) on the command line, moving the cursor to the field of interest and pressing Enter.

If the application has inherited relationships from a class, only these are displayed in a different color. If you change one of these relationships, all the inherited relationships become specific to that particular class or instance (and thus change color to reflect this).

Entering the NEW command from this panel to define new relationships takes you to the Define Relationship panel described in “Defining Relationships” on page 127.

The External Startup and External Shutdown fields are the ones from “APPLICATION INFO Policy Item” on page 152 (see “Startup Parameters” on page 156 and “External Shutdown” on page 157 for details).
MESSAGES/USER DATA Policy Item

If you select policy item MESSAGES/USER DATA from the Policy Selection panel for Applications (Figure 95 on page 146), the Message Processing panel shown in Figure 104 is displayed.

On the Message Processing panel you specify how SA z/OS should react if the application issues the specified message. If the application is linked to a class, the panel shows both instance and class data sorted alphabetically, with the inherited data shown in a different color. For more details see “Nested Class Details” on page 180. You can find out which class a value has been inherited from by typing the command `classname` (or its abbreviation `cn`) on the command line, moving the cursor to the field of interest and pressing Enter.

Use the CANCEL command (abbreviated to C) to exit the panel without saving any changes to the data.

You can define message responses for each message ID. By default, the first 14 characters of the message ID is shown. The length of the description that is displayed depends on your screen width (for a 24x80 screen, it is 40 characters).

You can change the length of the message ID field with the Message ID field length field. Note that this value is not saved across sessions. If you increase the message ID field length, you can view any information on the right of the panel that is no longer displayed by scrolling right with the PF11 key.

You can enter different line commands in the Cmd column. Depending on what you type here, SA z/OS displays a follow-on panel where you further specify SA z/OS's automated reaction to the message. Note that the message ID must not end with an asterisk and cannot have any commas in it.

Generally, SA z/OS requires an entry in the NetView automation table that specifies which routine SA z/OS should use to process your specified action. SA z/OS is able automatically to build the NetView AT entries for all CMD, REP, and AT actions as well as for certain USR and COD actions.
MESSAGES/USER DATA Policy Item for Applications

Information about how to manage the NetView automation table is given in [IBM Tivoli System Automation for z/OS Customizing and Programming](#).

The following line commands are available:

**CMD** To issue commands as a response to a message, see [“Command Processing” on page 167](#).

**REP** To issue replies, see [“Reply Processing” on page 169](#).

**COD** To check for certain codes within a message. For different sets of code values, you can define a corresponding value that is returned to the calling REXX script that checks for a matching code definition. See [“Code Processing” on page 170](#).

**USR** You can specify your own data that is needed when SA z/OS runs user-specified actions if the current message requires more specific automation than provided by the generic routines (see [“User Data Processing” on page 173](#)).

Using the ACFFQRY common routine from within a REXX exec, you can retrieve the keywords entered into this panel and process them as needed. Syntax rules for specifying user data are available in the online help. For information on how to use ACFFQRY in a REXX exec see [ACFFQRY in IBM Tivoli System Automation for z/OS Programmer’s Reference](#).

**AT** Use the associated panels to customize the AT entry that is built for the message ID. See [“Message Automation Definitions” on page 174](#).

**MRT** Use the associated panels to customize the NetView message revision table (MRT) entry. See [“Message Automation Definitions” on page 174](#).

The **A** and **M** columns indicate whether there are any specifications that are related to entries for the NetView automation table or message revision table as follows:

* User data has been defined for the message ID. The * is always turquoise because the data cannot be associated with a class because there may be other instances with entries for the same message ID.

**P** There is a predefined entry for this message ID, shown in the color for inherited data.

You can enter the LOCATE (abbreviated to L) command to position to a specific message ID.

You can enter FIND (abbreviated to F) at the command line on the Message Processing panel to search the displayed data for a specific string, for example, f abc. All occurrences of the string are highlighted and the cursor is positioned at the start of the first one. You can use the RESET command to remove the highlighting.

Note that only the data that is visible is searched through in the Message ID column. You should therefore consider expanding the column width so that the complete message ID is visible before performing a search.

You can use the RFIND function key (PF5) to move to the next occurrence of the string. If you enter the RFIND command at the command line it repeats the last find.

For more details, see [“How to Find Data in a Policy Item” on page 23](#).
Working with Message IDs
If you add a new message ID that already exists elsewhere in the policy database and it has AT or MRT data specifications, pressing Enter causes the AT or MRT data indicators to appear in the corresponding columns. Furthermore, if you do not specify a message description for a message that already exists in policy database, the description is filled in from either a user-specification or a predefined entry in +SA_PREDEFINED_MSGS.

If you change a message ID, not only are the commands, replies, etc. kept, but also any AT or MRT data. If you change the message ID to one that already exists, there may be a conflict. For example, if message ID ABC123 is changed to DEF456:

- If ABC123 has AT data, but DEF456 does not, the AT data from ABC123 is copied to DEF456.
- If ABC123 does not have any AT data, but DEF456 does, the AT data from DEF456 are kept.
- If ABC123 and DEF456 both have AT data that differs, the message ID change is not possible and the message Conflicting AT data is issued.

You should also bear in mind that if the message ABC123 has also been defined for another application, it keeps any AT or MRT data that has already been defined.

If you delete a message ID that has been specified in the MESSAGES/USER DATA policy item of just one application or MVS component, this deletes all related data. The message ID no longer exists in the policy database, and is no longer displayed in MESSAGE AUTOMATION policy item of entry type ENT, and any AT or MRT specifications are removed from the policy database.

If you delete a message ID that has been specified in the MESSAGES/USER DATA policy item for multiple objects, and AT or MRT data exists, this does not delete the AT or MRT data. Any AT or MRT data is deleted only if the message is deleted from the last remaining object in the policy database, or the object itself is deleted.

Command Processing
SA z/OS displays the Command Processing panel (Figure 105) if you enter the action code CMD in the **Cmd** column on the Message Processing panel.

<table>
<thead>
<tr>
<th>Cmd</th>
<th>Ps/Select</th>
<th>AutoFn/*</th>
<th>Command Text</th>
<th>Scroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAST</td>
<td></td>
<td></td>
<td>IMSMD &amp;SUBSAPPL /STA OLDS 99</td>
<td></td>
</tr>
<tr>
<td>SYSTEM</td>
<td></td>
<td></td>
<td>IMSMD &amp;SUBSAPPL /LOG WAITING</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Bottom of data</td>
</tr>
</tbody>
</table>

Figure 105. Command Processing Panel for Applications

Use this panel to enter commands that are executed when the application issues the specified message. If you specify one of the automation commands that are provided by SA z/OS (see [IBM Tivoli System Automation for z/OS Operator’s](#))
Command Processing for Applications

**Commands**, they should be queued using the NetView CMD LOW command. This ensures that task global variables that are in use by the calling automation procedure are not changed by the calling command.

For a list of runtime variables that can be substituted into each command prior to execution, see ACFCMD in **IBM Tivoli System Automation for z/OS Programmer’s Reference**.

For a full description of using line commands, refer to “Line Commands on Message Processing Panels” on page 26.

The value that you enter in the **Ps/Select** field defines when the command or reply should be issued, as follows:

**Pass**  Enter a number from 1 through 99 or an asterisk (*) in this field to specify a pass value. During the automation control file build, this value is prefixed with PASS to create the selection criterion for the CMD or REP entry.

**Note:** For more information about pass processing see the description of the PASSES parameter for the ISSUEACT generic routine in **IBM Tivoli System Automation for z/OS Programmer’s Reference**. Pay attention when specifying a number because this is treated as a selection.

**Select**  An expression other than a pass value is considered a selection value and is not altered during the build process. A selection value can be a blank or any name that does not begin with a number or contain imbedded blanks, commas, quotes, or parentheses.

**Note:** For commands issued by SA z/OS during startup or shutdown, an attempt is made to issue only the commands that are relevant to the particular start or stop type. SA z/OS looks for command definitions with a selection field that matches the start or stop type. Commands that are defined without a selection value are issued or selected for each start or stop type. Where appropriate, the AAO AOFSTATUSCMDSEL is taken into consideration.

**AutoFn/***  Enter the name of the automated function that the command is to run under. The AT entry that is generated does not reflect the automated function that you enter, however the execution of the command is routed internally to the automated function that you specify. If you leave the field blank, the function defaults to the one currently running the command list. SA z/OS automated functions are mapped to NetView automation operators.

**Note:** Entering * enables return-code checking for the specified command. This means that processing continues if the command returns a return code of 0. In all other cases processing is stopped immediately. If you are running in LOG assist mode, return code checking has no effect because the commands are not executed.

**Command Text**  Enter the command or command list to be issued. You must preface a system command with MVS.
**Command Processing for Applications**

Pass or selection values can be repeated for several text entries. Each set of entries with the same pass or selection value forms a command group.

To process this message an entry in the NetView automation table is generated that uses the ISSUEACT generic routine. For more details, see ISSUEACT in *IBM Tivoli System Automation for z/OS Programmer’s Reference*.

**Reply Processing**

SA z/OS displays the Reply Processing panel if you enter the action code REP in the **Cmd** column on the Message Processing panel.

![Reply Processing Panel](image)

**Figure 106. Reply Processing Panel for Applications**

Use this panel to enter the replies that are to be issued when this resource issues the specified message. For a list of run time variables that can be substituted into each reply prior to execution, see ACFREP in *IBM Tivoli System Automation for z/OS Programmer’s Reference*.

You can use line commands in the **Cmd** field to manipulate the entries on this panel as described in “Line Commands on Message Processing Panels” on page 26.

The use of the **Ps/Select** field is similar to that described in “The Ps/Select Field” on page 168. Pass or selection values can be repeated for several text entries. Each set of entries with the same pass or selection forms a reply group.

To process this message an entry in the NetView automation table is generated that uses the ISSUEACT generic routine. For more details, see ISSUEACT in *IBM Tivoli System Automation for z/OS Programmer’s Reference*.

Additionally, there are the following fields:

**Ct**

This is the **Retry Count** field. Enter a number in the range 1–99 in this field to define the number of retry attempts that are made before the common routine decides that a reply cannot be issued. A retry occurs when the outstanding reply number is not available. For the first retry there is a one second time delay and for all further retries there is a two second time delay. The default retry count is 0.

**Reply Text**

Code the actual reply to the message in this field. Enter it in the same format as you would do from an MVS console (without the R and the reply number). To specify a null reply, type NULL in this field.
Code Processing for Applications

**Code Processing**

SA z/OS displays the panel shown in Figure 107 if you enter the action code COD (K) in the Cmd column on the Message Processing panel.

Use this panel to specify a series of code definitions. Each code definition consists of up to three code values and an associated Value Returned.

The code definitions can be used by automation procedures to check, if a set of up to three given values matches with one of the code definitions in the automation policy. The values to be checked may be extracted variable values from an automated message or any other retrieved values, for which a matching code definition is searched. The associated Value Returned of the matching code definition can be used to decide about the further processing in the automation procedure.

Code definitions are used by SA z/OS, for example, to determine the severity of received WTORs. Further on, when processing termination messages of resources, SA z/OS uses code definitions to decide what status to place the resource in. In your own automation routine, you can use the common routine CDEMATCH to compare a set of given values with a series of code definitions in the automation policy and to gather the Value Returned from the matching code definition. For more details about common routine CDEMATCH, see IBM Tivoli System Automation for z/OS Programmer’s Reference.

You can use line commands in the **Cmd** field to manipulate the entries on this panel as described in “Line Commands on Message Processing Panels” on page 26.

The fields on this panel are used by the CDEMATCH routine.

**Code 1, 2, 3**

Patterns in these fields are compared with a set of up to three given values that are provided by an automation procedure. If one of the defined sets of code patterns matches, the content of the corresponding Value Returned field is returned to the automation procedure and no further comparison is made.

If no match is found at all a null string is returned.

If the pattern is a single token, string comparison is performed. Such a pattern can contain wildcard characters in any position and combination. An asterisk (*) matches a string of arbitrary length and a percent sign (%) matches a single character.
Preceding the asterisk or percent sign with a backslash (\) prevents these characters from being considered as wildcard characters.

A pattern can also consist of two tokens. In this case, the first token is a string or numeric comparison operator and the second is the value to be compared to (wildcard characters are not supported).

The following string comparison operators are supported:

- $EQ equal to
- $NE not equal to
- $GT greater than
- $GE greater than or equal to
- $LT less than
- $LE less than or equal to

String comparison is done by character and in binary format after all the values have been translated to uppercase. For strings with different lengths, the shorter string is padded with blank characters at the end of the string to make it the same length as the longer string. Strings that consist of whole numbers with or without leading zeros are not considered as equal.

The following numeric comparison operators are supported:

- @EQ equal to
- @NE not equal to
- @GT greater than
- @GE greater than or equal to
- @LT less than
- @LE less than or equal to

Whole numbers up to an absolute value of $2^{31} - 1$ are accepted.

For numeric comparison, whole numbers with or without leading zeros considered as equal.

**Note:** The usage of the code fields depends on the invocation of the CDEMATCH routine.

**Value Returned**

If all the specified values in the code fields match the passed values from the automation procedure, the value in this field is returned to the automation procedure to be used for decision making.

**Note:** A special value of *IGNORE* is honored by the SA z/OS ISSUEACT command so that no operation is triggered.

Code match checking is used by SA z/OS in the following situations:

- Processing of messages with ISSUEACT to trigger appropriate commands or replies.
- Processing of messages or OMEGAMON® exceptions with INGMON to update the status of a monitoring resource or to issue appropriate commands or replies.
- Processing of termination messages with TERMMSG to determine which status to place the resource in.
- Processing of WTORs with OUTREP to assign a severity.
- Capturing of messages with AOFCPMSG to assign a severity.
- Sending events with INGALERT to event notification targets such as IOM, EIF, TTT, or USR.
Code Processing for Applications

For details see the description of these commands in *IBM Tivoli System Automation for z/OS Programmer’s Reference*.

- Processing of CICS abend messages using message ID ABCODESYSTM to determine whether or not to restart the subsystem.
- Processing of CICS or IMS transaction abend recovery using message ID ABCODETRAN.
- Processing of IMS BMP region abends using message ID ABCODEPROG.

For details see the description of these message IDs in *IBM Tivoli System Automation for z/OS Product Automation Programmer’s Reference and Operator’s Guide*.

- Processing of WTO buffer shortage recovery. For details see *IBM Tivoli System Automation for z/OS Customizing and Programming*.

**Example 1:**

<table>
<thead>
<tr>
<th>Code 1</th>
<th>Code 2</th>
<th>Code 3</th>
<th>Value Returned</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>IMSCTL</td>
<td>WARNING</td>
<td></td>
</tr>
<tr>
<td>*</td>
<td>IMSCTL</td>
<td>CRITICAL</td>
<td></td>
</tr>
</tbody>
</table>

When checking two values for a match with the code definitions of this example, the first code definition matches with the given values, if the second value is IMSCTL. In this case, WARNING is returned as the value.

If the second given value consists of IMS and CTL with any single character other than T between, a match with the second code definition occurs and CRITICAL is returned as the value.

Consider the following examples:

<table>
<thead>
<tr>
<th>Values checked</th>
<th>Value returned</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>DFS2013 and IMSCTL</td>
<td>WARNING</td>
<td>These values match the first code definition</td>
</tr>
<tr>
<td>DFS2013 and IMS1CTL</td>
<td>CRITICAL</td>
<td>These values match the second code definition</td>
</tr>
<tr>
<td>DFS2013 and IMS10CTL</td>
<td>null string</td>
<td>These code values do not match any of the code definitions</td>
</tr>
</tbody>
</table>

**Example 2:**

<table>
<thead>
<tr>
<th>Code 1</th>
<th>Code 2</th>
<th>Code 3</th>
<th>Value Returned</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>TEST*</td>
<td></td>
<td>WARNING</td>
</tr>
<tr>
<td>*</td>
<td>8</td>
<td>RGT &amp;</td>
<td>CRITICAL</td>
</tr>
<tr>
<td>*</td>
<td></td>
<td>WARNING</td>
<td></td>
</tr>
</tbody>
</table>

When checking three values for a match the code definitions of this example, the three values match the first code definition, if the second value begins with the string TEST.

Three given values match the second code definition, if the second value does not begin with string TEST and the third value is a whole number greater than 8.

For example, the three values IEF451I, TESTJOB1 and 4 match the first code definition and WARNING is returned as value.
When checking with the three values IEF451I, IMS1CTL and 12, value CRITICAL is returned from the matching second code definition.

When checking with the values IEF451I, IMS1CTL and 4, value WARNING is returned from the matching last code definition.

More information about this panel is given in the online help.

**User Data Processing**

SA z/OS displays the User Data Processing panel, as shown in Figure 108, if you entered the action code USR in the **Cmd** column on the Message Processing panel.

This panel allows you to manage the list of keyword-data pairs.

You can use this panel to let SA z/OS perform user specified actions in case the current message requires more specific automation than provided by the generic routines. Using the ACFFQRY common routine from within a REXX exec, 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 REXX exec see ACFFQRY in IBM Tivoli System Automation for z/OS Programmer’s Reference.

The following fields are available:

**Mixed case**

Specify whether the value entered in the **Data** field remains unchanged (YES) or is translated to upper case characters automatically (NO).

**Keyword length**

For better readability only the first 20 characters of the **Keyword** value and the first 55 characters of the **Data** value is shown. Enter a value in this field to change how much of each field is visible.

**Keyword**

A value is required and should represent the keyword that will be assigned to the user data.

**Data**

A value is required and can contain any characters. There are, however,
rules regarding how the data value must be delimited if it contains any special characters. For more information about these rules, see the online help.

You can use line commands in the **Cmd** field to manipulate the entries on this panel as described in “Line Commands on Message Processing Panels” on page 26.

### Message Automation Definitions

SA z/OS supports the automatic generation of NetView automation tables (ATs) and the message revision table (MRT). You enable the automatic generation of the ATs and MRT using the **AT/MRT scope** field when you edit the settings for policy databases (see Chapter 8, “The NetView Automation Table and Message Revision Table,” on page 327).

If you do not enable the automatic generation of the ATs and MRT, you do not need to specify anything on the panels that are described in this section. If you do enable automatic generation, you can use the Message Automation Overview panel to view what SA z/OS will generate for a particular message in the AT and MRT. If the AT or MRT entry is not appropriate, you can use the options on the panel to modify it as described in this section.

If you enter AT in the **Cmd** field for a message ID, the Message Automation Overview panel is displayed, as shown in Figure 109:

```
Message ID : IEF451I
Generated AT entry :
  * The variable SVJOB is set by the BEGIN statement for IEF4
  IF MSGID = 'IEF451I' & ATF('INGSQRY APPL,,JOB=VALUE(SVJOB)) ^= ''
  & TEXT = 'ENDED BY CC' CCODE . THEN
  EXEC(CMD('TERMMSG JOBNAME=' SVJOB ',CODE1=' SVJOB ',CODE2=' CCODE)
  ROUTE(ONE %AOFOPGSSOPER%));

No MRT entry will be generated because there are no specifications for this message
```

**Figure 109. Message Automation Overview Panel**

This panel allows you to:
- View the AT or MRT entry that SA z/OS will automatically generate for the message ID
- Define your own NetView Automation Table (AT) entries
- Override predefined AT entries for a given Entry Name and Message ID
- Modify the specifications that are used to generate NetView message revision table entries

You can use the options listed at the top of the panel to display further panels, as follows:
- **AC**: A panel to specify conditions for the AT entry, see “Specifying AT Entry Conditions” on page 175
Message Automation Definitions for Applications

AS  A panel to select the SA z/OS status that is used to generate the AT action, see “Selecting a Status for the AT Entry” on page 176
AO  A panel to specify the AT override, see “Specifying an AT Override” on page 177
MS  A panel to specify what action is taken with the MRT entry that is generated, see “Specifying an Action for an MRT Entry” on page 178
MO  A panel to specify an MRT override, see “Specifying an MRT Override” on page 179
IG  A panel to specify whether an entry is ignored for the AT, MRT, or MPFLSTSA Message Processing Facility (MPF) member, see “Specifying that an Entry Is Not Built” on page 178

The scrollable AT/MRT preview area shows the AT and MRT entries that would be generated based on the current specifications. If, for example, you modify the AT conditions on a subsequent panel and return, the AT preview is updated to reflect the new specifications.

If a generated AT entry is based on an SA z/OS definition (that is, it is defined in the MVC entry +SA_PREDEFINED_MSGS), it is shown in the color that has been selected for class data.

Specifying AT Entry Conditions: Enter the line command AC on the Message Automation Overview panel to display the Automation Table entry Conditions panel, as shown in Figure 110.

The fields on this panel are:

Message placement in AT

Allows you to specify whether the generated AT entry should be placed at the beginning or end of the AT. In both cases the generated AT entry is not placed in a BEGIN-END block. If you set Check for message id to YES or blank, the default placement is in a BEGIN-END block, and if you set it to NO the default placement is BOTTOM.

Check for message id

Allows you to exclude the check IF MSGID = ‘...’ from the AT condition. This may be necessary if for example you specify the complete condition in the Additional condition field. If checking for message is excluded, the AT entry is not placed in a BEGIN-END block.
Message Automation Definitions for Applications

Ignore message id character
Allows you to use only a portion of a message ID when checking for the message ID. Enter one of the following to control which parts of the message ID are ignored:
LEADING The beginning of the message ID is ignored, that is, the suffix is used.
TRAILING The end of the message ID is ignored, that is, the prefix is used.
BOTH Both the beginning and the end of the message ID are ignored, that is, the string must appear somewhere within the message ID.

This behavior is achieved by adding a leading or trailing period to the AT condition, as shown in the preview in the lower half of the panel.

Additional condition
Allows you to specify any text that is added to the AT condition. There is no syntax check for this field and it allows mixed case.

Pressing Enter updates the preview that is shown in the lower half of the panel.

If a preview is not generated, an explanation is displayed in the preview area, for example, No AT entry will be generated because there is no Command, Reply, or Status defined.

Press PF3 to save your changes or enter the CANCEL (C) command to discard them and then return to the Message Automation Overview panel.

Selecting a Status for the AT Entry: Enter the line command AS on the Message Automation Overview panel to display the AT Status Specification panel, as shown in Figure 111.

<table>
<thead>
<tr>
<th>COMMANDS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFQMAT1</td>
<td>AT Status Specification</td>
</tr>
</tbody>
</table>

Enter "S" to make a selection for: IEF451I

<table>
<thead>
<tr>
<th>Application / MVC Component Status</th>
<th>Monitor Health Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVE Starting</td>
<td>NORMAL Resource shows good results</td>
</tr>
<tr>
<td>UP Available</td>
<td>WARNING Resource shows degradation</td>
</tr>
<tr>
<td>MALTED Degraded</td>
<td>MINOR More severe than WARNING</td>
</tr>
<tr>
<td>Terminating Received STOP command</td>
<td>CRITICAL More severe than MINOR</td>
</tr>
<tr>
<td>Terminated Terminated normally</td>
<td>FATAL More severe than CRITICAL</td>
</tr>
<tr>
<td>ABENDING Going to end abnormally</td>
<td>UNKNOWN Healthstate not available</td>
</tr>
<tr>
<td>Abended Ended abnormally</td>
<td>Check Evaluate health status</td>
</tr>
<tr>
<td>BREAKING Non-recoverable abending</td>
<td>Monitor Status</td>
</tr>
<tr>
<td>BROKEN Non-recoverable abend</td>
<td>FAILED Monitor failed</td>
</tr>
<tr>
<td>Capture Capture Message</td>
<td>BROKEN Monitor finally failed</td>
</tr>
</tbody>
</table>

Figure 111. AT Status Specification Panel

You can select one status for an application or monitor resource, or both.

For messages that are defined for a monitor resource only but have an application status and a monitor status specification, the AT entry is generated both for the application status and the monitor status.
Message Automation Definitions for Applications

If you select a monitor status for a message but then remove the message from the MESSAGES/USER DATA policy item of the monitor resource, the monitor status is ignored for the AT build. The monitor status selection is however still shown on this panel until you remove it.

Press PF3 to save your changes or enter the CANCEL (C) command to discard them and then return to the Message Automation Overview panel.

**Specifying an AT Override:** Enter the line command AO on the Message Automation Overview panel to display the ISPF editor where you can specify an AT override, as shown in Figure 112.

If you have defined an AT action or condition, or both, and then invoke the override panel, the AT entry preview is shown on the override editor screen. You can use this as a model for your own AT definition. You can use system symbols and SA z/OS symbols in your specification.

Use the END command (PF3 key) to leave the editor panel and save any changes as an AT override.

Use the CANCEL command to leave the editor panel and discard any changes that you have made.

**Notes:**

1. You can only define one override per message ID and thus this override applies to all applications where the message has been specified. If you define a command against a message ID that already has an override you must ensure that the command is honored via the override.

2. If an AT override is specified when a specific AT action or condition already existed, then this AT action or condition is removed.

**Syntax check of an AT or MRT Override:** On leaving the override panel, a syntax check is attempted. This check is not performed in the TSO/ISPF environment, but by NetView which must be available on the same system. The prerequisites are described in detail in "Enabling and Disabling AT / MRT Override Syntax Checking" on page 362. If no syntax error is detected, a message is issued indicating a successful check. If syntax errors are found, the error messages are displayed on a pop-up panel which allows:

- to return to the override EDIT panel to fix the error, or

---

Figure 112. Editor Panel to Specify an AT Override

If you have defined an AT action or condition, or both, and then invoke the override panel, the AT entry preview is shown on the override editor screen. You can use this as a model for your own AT definition. You can use system symbols and SA z/OS symbols in your specification.

Use the END command (PF3 key) to leave the editor panel and save any changes as an AT override.

Use the CANCEL command to leave the editor panel and discard any changes that you have made.

Notes:

1. You can only define one override per message ID and thus this override applies to all applications where the message has been specified. If you define a command against a message ID that already has an override you must ensure that the command is honored via the override.

2. If an AT override is specified when a specific AT action or condition already existed, then this AT action or condition is removed.

Syntax check of an AT or MRT Override: On leaving the override panel, a syntax check is attempted. This check is not performed in the TSO/ISPF environment, but by NetView which must be available on the same system. The prerequisites are described in detail in "Enabling and Disabling AT / MRT Override Syntax Checking" on page 362. If no syntax error is detected, a message is issued indicating a successful check. If syntax errors are found, the error messages are displayed on a pop-up panel which allows:

- to return to the override EDIT panel to fix the error, or

---

Chapter 4. Entry Type Reference 177
to confirm that the error message is accepted (for example, an unresolved variable) and the override EDIT panel is left.

If the syntax check is not run because the prerequisites are not met or there is a communication error with NetView then an error message is issued.

The syntax check is not run if the override panel is left with CANCEL.

The syntax check can be disabled, for example, if the Customization Dialog is running on a system without NetView. There is an option on the 'Message and Logging Functions' panel in '0 Settings', selection '3 Message'.

**Specifying that an Entry Is Not Built:** Enter the line command IG on the Message Automation Overview panel to display the AT/MRT Ignore Specification panel, as shown in Figure 113.

---

**Commands Help**

AOFGMSGS AT/MRT Ignore Specification

Enter "S" to make a selection for: IEF451I

Message Ignore or Suppress option

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ignore</td>
<td>No AT entry; no MRT entry; no MPFLSTSA entry</td>
</tr>
<tr>
<td>Suppress</td>
<td>No AT entry; no MRT entry; MPFLSTSA entry SUP(YES),AUTO(NO)</td>
</tr>
</tbody>
</table>

Figure 113. AT/MRT Ignore Specification Panel

This panel allows you to specify that an entry is not built for the AT, MRT, or Message Processing Facility (MPF) member MPFLSTSA. Enter S for one of the following options:

**Ignore** This option does not consider the data to be an automatable message. This specification is recommended for user data that is not a message and should not be automated.

No NetView automation table or message revision table entry is generated. No entry in the MPFLSTSA Message Processing Facility (MPF) member is provided.

**Suppress**

This option completely excludes the message from automation processing and suppresses its display on the MCS console.

No NetView automation table or message revision table entry is generated. The message is added to the MPFLSTSA MPF member with SUP(YES),AUTO(NO). Thus the message is stopped in the MPF member and effectively lost for any further processing.

**Specifying an Action for an MRT Entry:** Enter the line command MS on the Message Automation Overview panel to display the Message Revision Table Conditions panel, as shown in Figure 114 on page 179.
This panel allows you to view and modify specifications that are used to generate a NetView message revision table (MRT) entry for the current message ID. The MRT enables you to intercept MVS messages before they are displayed, logged, automated, or routed through your sysplex. Use the options on this panel to make changes to the following aspects of the message:

- Automation.
- System log attributes.
- Message text.
- Color, highlighting, and intensity (if your terminal supports high intensity). Only one selection per column is allowed.

If you select **Delete completely** no other selection is valid.

The options **Automate** and **Do not automate** are mutually exclusive.

You can combine one of the automation specifications and all of the other conditions (except **Delete completely**).

Any MRT specifications that you make are independent of any AT entry data for the message. Thus if you make a definition for the MRT but not for the AT, any predefined data for AT entry is still in effect.

**Specifying an MRT Override:** Enter the line command MO on the Message Automation Overview panel to display the ISPF editor where you can specify a NetView message revision table entry override, as shown in Figure 115 on page 180.
Message Automation Definitions for Applications

Use the END command (PF3 key) to leave the editor panel and save any changes as an MRT override. Use the CANCEL command to leave the editor panel and discard any changes that you have made.

Nested Class Details
CMD (Command), REP (Reply), COD (Code), and USR (User Data) policies are inherited per message ID. Thus if a message ID has a command definition for an instance and the same message ID is defined for a class with reply data, the command and reply data are not merged on the instance, and the class definitions are not inherited at all.

AT and MRT specifications provide instructions for the generation of the NetView automation table and message revision table entry. This data exists only once in the policy database and is not tied to a particular application, monitor resource or MVS Component. It is used to create one AT entry or one MRT entry. Thus the inheritance rules do not apply.

Examples: Consider the following examples that demonstrate how policy values are inherited and AT entries are built:

No overrides
A class has a command and a linked instance has a reply defined for the message ID, ABC123.

Result: The following AT entry is built:

```
IF MSGID = 'ABC123' THEN
EXEC(CMD('ISSUEACT ')ROUTE(ONE %AOFOPSSOPER%));
```

However, the command is not executed because it is not inherited by the instance and so is not loaded during automation agent initialization.

Two applications and an override
One application has a command defined for a message ID, another application has a reply defined for the same message ID, and an override is defined for the message ID.

Result: One AT entry is generated for the override. This override should deal with any commands or replies that should be executed.

Panel Interaction: The Message Processing panel shows message IDs that are defined for the current application and any data that has been inherited from
linked classes. Inherited message IDs are displayed in a different color. AT and
MRT specifications are always indicated by an asterisk * for either an instance or a
class. If the AT or MRT specifications are for predefined messages, these are
indicated with a P in the color that is used for classes.

If a linked class and the current application have definitions for the same message
ID, only the definition for the current application is displayed.

Consider the following examples that illustrate inheritance from class message IDs
that are modified, and how the panel display reflects this:

Modifying commands for an inherited message ID
Consider an inherited message ID with command and reply definitions
and a status definition. If you modify a command definition, this becomes
a definition for the current application along with the reply data, however
the status data is not affected.

Specifying AT data for an inherited message ID
Again, consider an inherited message ID with command and reply
definitions and an AT definition. If you modify the AT data, the message
ID does not become a definition for the current application. The display
shows the message ID with the counts for Cmd and Rep in the color for
inherited data and the * indicator for AT (the A column) in turquoise.

Command and reply for a class and instance
Consider a class definition that has a command defined for message ID,
and a linked instance that has a reply defined for the same message ID,
and there is also a Status definition, for example BROKEN.

One AT entry is generated using TERMMSG, the reply from the instance is
executed, but the command from the class is not inherited (and thus not
executed). The following line is displayed all in turquoise:

<table>
<thead>
<tr>
<th>Cmd Message id</th>
<th>Description</th>
<th>Cmd Rep Cod Usr A M</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC123</td>
<td>Descriptive text</td>
<td>1 *</td>
</tr>
</tbody>
</table>

STARTUP Policy Item
The STARTUP policy item is used to specify the scheduling subsystem, JCL
procedure name (which are also available on the APPLICATION INFO policy
item), the startup parameters and all the commands necessary to start a subsystem.
These commands are issued when SA z/OS is going to start the subsystem.

You can use the STARTUP policy item to define several command sequences that
are executed if an application requires certain start types that depend upon
particular circumstances, such as a cold start, warm start or recovery start. You can
use the INGREQ command to specify which command sequence should be taken.

You can also use this facility for performance improvements. Consider the example
in Figure 116 on page 182 if application A depends on application B, and B in turn
depends on C, and A also has a PREPAVAILABLE relation to C, after C is started,
you can have SA z/OS issue a sequence of prestart commands to A while the
supporting resource B is starting.
If you select the STARTUP policy item from the Policy Selection panel for Applications (Figure 95 on page 146), you come to the Subsystem Startup Processing panel shown in Figure 117.

For each application, you can specify a list of commands (startup procedures) that are used:
- Before the application is started
- To start the application
- After the application has started

All specified commands are 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 is that you can have SA z/OS perform certain actions for the preparation of an application. For example, before starting CICS, you can have the VTAM major node activated.

The fields in the subheader area can be inherited individually from a linked class and if they have been inherited, they are shown in a different color. Any commands that have been inherited for the prestart, start or poststart phases are also shown in this color in the **Cmd** column. You can find out which class a value has been inherited from by typing the command `classname` (or its abbreviation `cn`) on the command line, moving the cursor to the field of interest and pressing Enter.

Command definitions are inherited per start phase. For example, if a PRESTART command is defined for the instance and both PRESTART and STARTUP commands have been defined for the class, the instance inherits the STARTUP commands from the class.
STARTUP Policy Item for Applications

You can fill in the following fields:

**Scheduling Subsystem**
This optional field specifies the name of the subsystem that is to schedule the current application. See "Scheduling Subsystem" on page 145 for more details.

**JCL Procedure Name**
Specifies the name of the JCL procedure that is used to create this instance of the application. See "JCL Procedure Name" on page 145 for more details.

**Startup Parameters**
Specifies the desired subsystem startup parameters. See "Startup Parameters" on page 156 for more details.

**Specifying Startup Commands**
You can specify commands by entering S or C in the **Cmd** field for the following startup phases:

**PRESTART**
Commands entered for this type are issued before the application is started. You can use these to perform preparation work, for example, having SA z/OS issue VTAM commands to activate additional connections.

If a PrepAvailable relationship is defined for the resource, the PRESTART commands are executed when the PrepAvailable condition is satisfied, or as soon as the start request becomes a winning one. Therefore, if the PRESTART commands rely on another resource running, you should define a PrepAvailable relationship.

**STARTUP**
You should enter only one command that is 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 the **Job Name** and **JCL Procedure Name** entry fields).

**POSTSTART**
Commands entered for this type are issued immediately after the UP message has been received.

The **Cmd** column displays the number of commands that have been entered for a particular startup entry.

For each of these phases (PRESTART, STARTUP, and POSTSTART) a Command Processing panel for that phase is displayed (as shown in Figure 118 on page 184 for the POSTSTART), where you can enter the startup commands that are specific to the current application.
The entries on this panel can be inherited from a class, in which case they are displayed in a different color. Because all of the commands are inherited together, if you edit one of them they all change color to indicate that they are now particular to that instance.

You can find out which class a value has been inherited from by typing the command `classname` (or its abbreviation `cn`) on the command line.

You can use line commands in the `Cmd` field to manipulate the entries on this panel as described in “Line Commands on Message Processing Panels” on page 26.

Use the following fields to enter the commands that are particular to this start phase:

**Type**

Enter the condition (for example, COLD) that the command should be executed for. NORM is a predefined type of SA z/OS.

Leaving this field blank means that the command is executed for any condition.

**AutoFn/***

Enter the name of the automated function that the command is to run under. The AT entry that is generated does not reflect the automated function that you enter, however the execution of the command is routed internally to the automated function that you specify. If you leave this field blank, the function defaults to the one that is currently running the command list. SA z/OS automated functions are mapped to NetView automation operators.

**Note:** Entering * enables return-code checking for the specified command. This means that processing continues if the command returns a return code of 0. In all other cases processing is stopped immediately. If you are running in LOG assist mode, return-code checking has no effect because the commands are not executed.

**Command Text**

Enter the command or command list that is to be issued. You must preface a system command with MVS.

---

**Figure 118. Command Processing Panel for Startup Commands**

The entries on this panel can be inherited from a class, in which case they are displayed in a different color. Because all of the commands are inherited together, if you edit one of them they all change color to indicate that they are now particular to that instance.

You can find out which class a value has been inherited from by typing the command `classname` (or its abbreviation `cn`) on the command line.

You can use line commands in the `Cmd` field to manipulate the entries on this panel as described in “Line Commands on Message Processing Panels” on page 26.

Use the following fields to enter the commands that are particular to this start phase:

**Type**

Enter the condition (for example, COLD) that the command should be executed for. NORM is a predefined type of SA z/OS.

Leaving this field blank means that the command is executed for any condition.

**AutoFn/***

Enter the name of the automated function that the command is to run under. The AT entry that is generated does not reflect the automated function that you enter, however the execution of the command is routed internally to the automated function that you specify. If you leave this field blank, the function defaults to the one that is currently running the command list. SA z/OS automated functions are mapped to NetView automation operators.

**Note:** Entering * enables return-code checking for the specified command. This means that processing continues if the command returns a return code of 0. In all other cases processing is stopped immediately. If you are running in LOG assist mode, return-code checking has no effect because the commands are not executed.

**Command Text**

Enter the command or command list that is to be issued. You must preface a system command with MVS.
SHUTDOWN Policy Item

If you select the SHUTDOWN policy item from the Policy Selection panel for Applications, the Subsystem Shutdown Processing panel is displayed, as shown in Figure 119.

Use the SHUTDOWN option to specify all the commands that are necessary to shut down a subsystem and follow-on activities.

Any commands or replies that have been inherited from linked classes for the different shutdown phases are shown in a different color in the **Cmd** and **Rep** columns.

You should bear in mind the following inheritance behavior for shutdown definitions:

- Shutdown specifications are inherited per phase. For example, if you define a SHUTINIT command for an instance that has been linked to a class that has both SHUTINIT and SHUTNORM commands defined for it, the instance inherits only the SHUTNORM commands from the class.

- Command and reply definitions for each phase are inherited together. For example, if you define a command for the SHUTFORCE phase for an instance that is linked to a class that has a reply defined for SHUTFORCE, nothing is inherited by that instance.

- If you modify any inherited data, that definition becomes one for the current application. For example, if command and reply definitions are inherited for a phase and you then modify the commands, both the reply and command definitions become data for the current application. This is also the case if an instance inherits only command definitions for a phase and you then define a reply for that phase.

You can find out which class a value has been inherited from by typing the command `classname` (or its abbreviation `cn`) on the command line, moving the cursor to the field of interest and pressing Enter.

Because shutdown procedures may involve replies to outstanding messages as well as commands, the **Cmd** field is used to indicate whether a command or reply is to be specified for a particular shutdown phase. If you enter either CMD or REP, you go to another panel where the actual commands and replies can be defined.

Figure 119. Subsystem Shutdown Processing Panel
SHUTDOWN Policy Item for Applications

Similar to the startup processing of applications described in the previous section, the following commands or replies can be chosen from this panel:

**INIT**  Commands and replies that you enter for this phase are issued when the shutdown process is initiated (that is, before the subsystem is brought down). Possible entries could be commands to send out warning messages to users or quiescing activities.

If you specified a PREPUNAVAILABLE relationship, 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 is shutting down.
- For the JES2 subsystem, you may wish to stop users on JES2 NJE lines from starting new transmissions or sessions.

**Note:**
You should not use the initial shutdown commands to actually shut down the application because unpredictable results may occur.

**NORM**
If the NORM option is selected for the INGREQ STOP command, all the commands and replies that are entered for this phase are issued. These entries should be the commands and replies that perform a normal shutdown of the subsystem.

**IMMED**
The commands and replies specified here are run if the IMMED option for INGREQ STOP is selected. These entries bring the subsystem down immediately.

**FORCE**
The entries specified here are run if the FORCE option for INGREQ STOP is selected. The commands and replies would be the typical ones a user would enter to force a subsystem off the system without any delay.

**FINAL**
The commands specified here are run after the final termination message has been received. Note that shutdown reply processing is not allowed for FINAL phases, that is, you can only specify commands for this shutdown phase.

The **Cmd** and **Rep** columns display the number of commands or replies that have been entered or inherited from a linked class for a particular shutdown entry.

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** field in the Application Automation Definition panel) the subsystem is still not down, another attempt (or pass) is made. 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...
SHUTDOWN Policy Item for Applications

advisable. Either code secondary replies on subsequent passes or chain them through the automation table and ISSUEREP.

All commands for a given pass are processed before any replies for that pass are processed. After each set of commands and replies is issued, SA z/OS waits 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 or reply, or both, defined for the next pass are issued. If a call to the application monitor routine shows that the application has shut down, any subsequent shutdown passes are not processed. If the final termination message is not received within the time specified in AOFSHUTDELAY, SA z/OS assumes that the application is down.

If SA z/OS processes all the shutdown passes defined for an application and the application monitor routine still shows the application as active, a warning is sent to the operator and the application status is set to STUCK.

The next panel that is displayed depends on the type of shutdown processing that you specify in the Cmd field for the shutdown phases on the Subsystem Shutdown Processing panel:

- Specify command processing for all shutdown types by entering CMD, C or S.
- Specify reply processing for all shutdown types except FINAL by entering REP or R.

How to proceed is described in the following sections:
- “Shutdown Command Processing for INIT or FINAL”
- “Shutdown Command Processing for NORM, IMMED, or FORCE” on page 188
- “Shutdown Reply Processing for INIT” on page 190
- “Shutdown Reply Processing for NORM, IMMED, or FORCE” on page 191

Shutdown Command Processing for INIT or FINAL
To define shutdown commands for INIT or FINAL phases, you enter S or C in the Cmd field on the Subsystem Shutdown Processing panel for those phases. This displays a Command Processing panel, as shown in Figure 120.

![Command Processing Panel for SHUTINIT or SHUTFINAL](image)

Figure 120. Command Processing Panel for SHUTINIT or SHUTFINAL

You can use line commands in the Cmd field to manipulate the entries on this panel as described in “Specifying Startup Commands” on page 183.
**SHUTDOWN Policy Item for Applications**

Use the following fields to enter the commands that are particular to the specified shutdown phase:

- **Type**: Enter the condition that the command should be executed for. The following values (for the corresponding shutdown phases) are allowed:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Shutdown phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>NORM</td>
<td>SHUTNORM</td>
</tr>
<tr>
<td>IMMED</td>
<td>SHUTIMMED</td>
</tr>
<tr>
<td>FORCE</td>
<td>SHUTFORCE</td>
</tr>
</tbody>
</table>

- **AutoFn/***: Enter the name of the automated function that the command is to run under. The AT entry that is generated does not reflect the automated function that you enter, however the execution of the command is routed internally to the automated function that you specify. If you leave this field blank, the function defaults to the one that is currently running the command list. SA z/OS automated functions are mapped to NetView automation operators.

  **Note**: Entering * enables return-code checking for the specified command. This means that processing continues if the command returns a return code of 0. In all other cases processing is stopped immediately. If you are running in LOG assist mode, return-code checking has no effect because the commands are not executed.

- **Command Text**: Enter the command or command list that is to be issued. You must preface a system command with MVS.

**Notes:**

1. You cannot issue multiple passes of initial shutdown commands.
2. Do not invoke MDFYSHUT.

The entries on this panel can be inherited from a class, in which case they are displayed in a different color. Because all of the commands are inherited together, if you edit one of them they all change color to indicate that they are now particular to that instance.

You can find out which class a value has been inherited from by typing the command `classname` (or its abbreviation `cn`) on the command line.

**Shutdown Command Processing for NORM, IMMED, or FORCE**

To define shutdown commands for NORM, IMMED, or FORCE phases, you enter S or C in the **Cmd** field on the Subsystem Shutdown Processing panel for those phases. This displays a Command Processing panel, as shown in Figure 121 on page 189.
You can use line commands in the **Cmd** field to manipulate the entries on this panel as described in "Specifying Startup Commands" on page 183.

Use the following fields to enter the commands that are particular for the specified shutdown phase:

**Ps (Pass)**

You can enter any number between 1 and 99 in this field. Commands or replies with 1 in their corresponding **Pass** field are the entries that are issued during the first attempt at shutting down the subsystem.

If after a specified amount of time (entered in the **Shut Delay** field on the Subsystem Definition panel) the subsystem is still not down, another attempt (or pass) is made. The next pass runs all entries that have 2 in their **Pass** field. This continues until the subsystem is down. If you do not specify a pass for a command, it is issued on every pass.

For example, Figure 121 defines the normal shutdown commands that are issued for the VTAM subsystem. This example defines two passes of shutdown commands. On the first pass, the command \texttt{MVS Z NET,QUICK} is issued. On the second pass, the command \texttt{MVS Z NET,CANCEL} is issued.

In another example, \texttt{MVS P TSO} could be a command for pass 1 and \texttt{MVS C TSO} could be the command for pass 2 (which means that if purging TSO fails, it is canceled).

**AutoFn/**

Enter the name of the automated function that the command is to run under. The AT entry that is generated does not reflect the automated function that you enter, however the execution of the command is routed internally to the automated function that you specify. If you leave this field blank, the function defaults to the one that is currently running the command list. SA z/OS automated functions are mapped to NetView automation operators.

**Note:** Entering * enables return-code checking for the specified command. This means that processing continues if the command returns a return code of 0. In all other cases processing is stopped immediately. If you are running in LOG assist mode, return-code checking has no effect because the commands are not executed.
SHUTDOWN Policy Item for Applications

Command Text
Enter the command or command list to be issued. You must preface any system command with MVS.

If you want to use MDFYSHUT to change the shutdown pass processing, you should specify it as a command on one of the passes.

The entries on this panel can be inherited from a class, in which case they are displayed in a different color. Because all of the commands are inherited together, if you edit one of them they all change color to indicate that they are now particular to that instance.

You can find out which class a value has been inherited from by typing the command \texttt{classname} (or its abbreviation \texttt{cn}) on the command line.

Shutdown Reply Processing for INIT
You enter R in the \texttt{Cmd} field for the INIT phase on the Subsystem Shutdown Processing panel to define shutdown replies for that phase. This displays a Reply Processing panel, as shown in Figure 122.

You can use line commands in the \texttt{Cmd} field to manipulate the entries on this panel as described in "Specifying Startup Commands" on page 183.

Use the following fields to enter the replies that are particular to the specified shutdown phase:

\textbf{Type} Enter the condition that the command should be executed for. The following values (for the corresponding shutdown phases) are allowed:

\begin{center}
\begin{tabular}{|c|c|}
\hline
\textbf{Condition} & \textbf{Shutdown Phase} \\
\hline
NORM & SHUTNORM \\
IMMED & SHUTIMMED \\
FORCE & SHUTFORCE \\
\hline
\end{tabular}
\end{center}

\textbf{Ct (Retry Count)} Enter a number in the range 1 to 99 in this field to define the number of retry attempts that are made before the common routine decides that a reply cannot be issued. A retry occurs when the outstanding reply number
SHUTDOWN Policy Item for Applications

is not available. For the first retry there is a one-second delay, and for all
further retries there is a two-second delay. The default retry count is 0.

Reply Text

Enter the reply that is to be issued when the selected shutdown phase is
invoked for this subsystem. For a null reply enter NULL as the reply.

The entries on this panel can be inherited from a class, in which case they are
displayed in a different color. Because all of the commands are inherited together,
if you edit one of them they all change color to indicate that they are now
particular to that instance.

You can find out which class a value has been inherited from by typing the
command classname (or its abbreviation cn) on the command line.

Shutdown Reply Processing for NORM, IMMED, or FORCE

To define shutdown replies for NORM, IMMED, or FORCE phases, you enter R in
the Cmd field on the Subsystem Shutdown Processing panel for those phases. This
displays a Reply Processing panel, as shown in Figure 123.

You can use line commands in the Cmd field to manipulate the entries on this
panel as described in “Specifying Startup Commands” on page 183.

Use the following fields to enter the replies that are particular to the specified
shutdown phase:

Ps (Pass)

Enter a number between 1 and 99 in this field. Commands or replies with
1 in their corresponding Pass field are the entries that are issued during
the first attempt at shutting down the subsystem.

If, after a specified amount of time (entered in the Shut Delay field on the
Subsystem Definition panel) the subsystem is still not down, another
attempt (or pass) is tried. The next pass runs all entries that have 2 in their
Pass field. This continues until the subsystem is down. A pass number is
required for each shutdown reply.

For example, MVS P TSO could be a command for pass 1 and MVS C TSO
could be the command for pass 2 (which means that if purging TSO fails,
it is canceled).
SHUTDOWN Policy Item for Applications

Ct (Retry Count)
Enter a number in the range 1 through 99 in this field to define the number of retry attempts that are made before the common routine decides that a reply cannot be issued. A retry occurs when the outstanding reply number is not available. For the first retry there is a one-second delay, and for all further retries there is a two-second delay. The default retry count is 0.

Reply Text
Enter the reply to be issued when the selected shutdown phase is invoked for this subsystem. For a null reply enter NULL as the reply.

The entries on this panel can be inherited from a class, in which case they are displayed in a different color. Because all of the commands are inherited together, if you edit one of them they all change color to indicate that they are now particular to that instance.

You can find out which class a value has been inherited from by typing the command classname (or its abbreviation cn) on the command line.

THRESHOLDS Policy Item
The Thresholds Definition panel is displayed if you select policy item THRESHOLDS on the Policy Selection panel for applications, as shown in Figure 124.

<table>
<thead>
<tr>
<th>COMMANDS HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOPG0YNA</td>
</tr>
<tr>
<td>Command ====&gt;</td>
</tr>
</tbody>
</table>

Entry Type : Application    PolicyDB Name : USER_PDB
Entry Name : APPL1           Enterprise Name : USER_ENTERPRISE
Resource : APPL1

Critical Number ... 5 (1 to 50)
Critical Interval ... 01:10:00 (hh:mm:ss or hhmmss, 00:00:01 to 24:00:00)
Frequent Number ... 3 (1 to 50)
Frequent Interval ... 01:00:00 (hh:mm:ss or hhmmss, 00:00:01 to 24:00:00)
Infrequent Number ... 3 (1 to 50)
Infrequent Interval ... 12:00:00 (hh:mm:ss or hhmmss, 00:00:01 to 24:00:00)

Figure 124. Thresholds Definition Panel

The information provided in the THRESHOLDS dialogs is used to determine when an operator should be notified of an error situation. An attempt at recovering an error (such as an abend of a subsystem) is handled by automation, but an operator may want to know if this error is occurring often.

Thresholds can be set that let the operator know if certain errors are occurring infrequently, frequently or have reached a critical stage where the recovery process should be ended (to avoid endless loops). This is done by specifying how many times an error must happen in a certain time period for each error situation.

The entries on this panel can be inherited from a class, in which case they are displayed in a different color. Because all of the thresholds are inherited together, if you edit one of them, all of the thresholds change color to indicate that they are now particular to that instance because they are no longer inherited.
THRESHOLDS Policy Item for Applications

You can find out which class a value has been inherited from by typing the command `classname` (or its abbreviation `cn`) on the command line.

You can define three different threshold levels:

**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.

You can enter data in the following fields:

**Number**
This field is used to define the number of occurrences of an error that creates the particular error level. The range of allowable numbers is 1 to 50.

**Interval**
The number of errors entered in the **Number** field must occur in a certain time period in order to create a specific error level. You specify this time period in the **Interval** field. You can specify it in the form HH:MM:SS or HHMMSS. Valid values are 00:00:01 to 24:00:00.

The values entered in the **Number** and **Interval** fields should be such that the threshold levels are reached in a logical manner (a critical threshold should not be reached before a frequent threshold).

Not all three threshold levels need be specified, for example, if an application needs only a specification for the critical number and interval, you can leave the frequent and infrequent fields empty.

For more information on setting thresholds see the section “How SA z/OS Uses Error Thresholds” in IBM Tivoli System Automation for z/OS User’s Guide.

MINOR RESOURCES Policy Item

The Minor Resource Definitions panel, as shown in Figure 125 on page 194, is displayed if you select the MINOR RESOURCES policy item on the Policy Selection panel for applications.
MINOR RESOURCES Policy Item for Applications

This panel allows you to manage automation flags and thresholds for minor resources.

**Cmd**
Use this field to enter a line command:

- **F or FLG**
  Enter automation flag definitions for the minor resource. See "AUTOMATION FLAGS Policy Item" on page 160 for more details.

- **S, T, or THR**
  Enter threshold definitions for the minor resource. See "THRESHOLDS Policy Item" on page 192 for more details.

- **CN**
  Displays the class name for the column where the cursor is positioned. If the cursor is in the "Cmd" column and data are inherited from one class only then that class is shown. If data are inherited from multiple classes then the name of the "lowest-level" class is shown.

- **D**
  Deletes the minor resource definition.

**Minor Resource**
Use this field to specify the minor resource name. The following rules apply to the minor resource names:

- The name cannot contain any of the following characters: comma, single quotation mark, double quotation mark, left or right parenthesis, equal sign or a blank.
- The name cannot start or end with a period.
- The name cannot contain consecutive periods.

To add a minor resource enter a name in one of the empty entries at the bottom of the list.

To rename a minor resource overtype the current name with a new name and press Enter.

**Automation Flags**
Displays an asterisk if there is user data. If the data is defined on a linked class then the asterisk is shown in the color of the inherited data. The following characters indicate the automation flags set for the minor resource:
MINOR RESOURCES Policy Item for Applications

<table>
<thead>
<tr>
<th>Thres</th>
<th>Threshold - indicates thresholds are defined for the resource.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Automation - is the global automation flag that must be YES for</td>
</tr>
<tr>
<td></td>
<td>the remaining flags to be used. If it is NO, there is no automation</td>
</tr>
<tr>
<td></td>
<td>for the resource.</td>
</tr>
<tr>
<td>I</td>
<td>Initstart - indicates whether a resource can be started when its state</td>
</tr>
<tr>
<td></td>
<td>is DOWN (that is, it controls initial starting of the resource).</td>
</tr>
<tr>
<td>S</td>
<td>Start - indicates whether the start command and actions or</td>
</tr>
<tr>
<td></td>
<td>responses to initialization for the resource are allowed.</td>
</tr>
<tr>
<td>R</td>
<td>Recovery - indicates whether recovery of the resource (actions on</td>
</tr>
<tr>
<td></td>
<td>messages that are neither for the initialization nor the ending of the</td>
</tr>
<tr>
<td></td>
<td>resource) is allowed.</td>
</tr>
<tr>
<td>T</td>
<td>Terminate - indicates whether the shutdown command and actions</td>
</tr>
<tr>
<td></td>
<td>on or responses to ending messages are allowed.</td>
</tr>
<tr>
<td>RS</td>
<td>Restart - indicates whether the restarting of a resource that</td>
</tr>
<tr>
<td></td>
<td>abnormally ended or was automatically shut down is allowed.</td>
</tr>
</tbody>
</table>

SA z/OS provides automation routines for recovery actions that use the automation flags or thresholds that are defined for these minor resources together with commands that are defined for messages with the same name. For more information see the chapter “Automation Routines” in IBM Tivoli System Automation for z/OS Customizing and Programming. See also the sections “Automation Flags and Minor Resource Flags” and “Thresholds for Minor Resources” in IBM Tivoli System Automation for z/OS User’s Guide.

SYSTEM ASSOCIATION Policy Item

The System Association panel, as shown in Figure 126, is displayed if you select the SYSTEM ASSOCIATION policy item on the Policy Selection panel for applications. Note that the specified definitions are not used for automation by SA z/OS. They are for customer-specific purposes only.

Using MOVED and FALLBACK Status

SA z/OS provides two automation statuses, MOVED and FALLBACK, to handle applications that normally run on one system (the primary) but which can run on an alternative system (the secondary) if the primary is unavailable.
SA z/OS does not provide any logic to actually perform a fallback from a primary to a secondary system. If you want cross-system fallback, you must 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.

**GENERATED RESOURCES Policy Item**

The Generated Resources panel appears if you select the GENERATED RESOURCES policy item from the Policy Selection panel for Applications, as shown in Figure 127. This panel shows all resources that are generated by SA z/OS for the application. This panel is for information only. No actions are allowed.

Application resources are generated when a link between an application and a system occurs. Applications can never be linked directly to systems. This link happens indirectly by linking an application to an application group (either in an application's WHERE USED policy item or in an application group's APPLICATIONS policy item.)

<table>
<thead>
<tr>
<th>COMMANDS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFGXRSG</td>
<td>Generated Resources</td>
</tr>
</tbody>
</table>

**Figure 127. Generated Resources Panel**

This panel displays the following information:

**Resource Name**

Shows the resources that have generated for the application.

**Generated by link to**

This says who the owner of a resource is. In this case; owners are application groups because the current entry is an APL.

A link to the owner creates the resource, unlinking from the owner (or deletion of the owner) deletes the resource.

**MEMBER OF Policy Item for Applications**

The Member of panel is displayed if you select policy item MEMBER OF on the Policy Selection panel for applications.
This panel lists all Resources where this Application is a member "resource-wise", that is, it lists all relations from resources that have been derived from the current Application to other resources. This panel is for information only. No data can be changed.

Use the RESOURCES policy of an APG to add or remove it to the list in the ...is a member of column (that is in the RESOURCES policy of the APG Entry Name that the member of Resource was derived from).

In this example, this panel is invoked for application AOR1, which is a member of a Sysplex Application Group. In such a case, you will see all the resources that are generated for the sysplex.

Policy Items for JES2 Subsystems

[Figure 129] is an example of a Policy Selection panel for a JES2 subsystem.

JES2 SPOOLFULL

Provides for recovery from a JES spool full condition. [Figure 130 on page 198] 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. See "JES2 SPOOLSHORT Policy Item" on page 199.
Policy Items for JES2 Subsystems

**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.

**JES2 SPOOLFULL Policy Item**
If you select policy item SPOOLFULL for a JES2 subsystem, SA z/OS displays the SPOOLFULL Processing panel.

<table>
<thead>
<tr>
<th>Command</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0FPEJA</td>
<td>SPOOLFULL Processing</td>
</tr>
</tbody>
</table>

**Entry Type:** Application  **PolicyDB Name:** USER_PDB  
**Entry Name:** JES2  **Enterprise Name:** USER_ENTERPRISE

Enter SPOOLFULL settings.

<table>
<thead>
<tr>
<th>Retry Time</th>
<th>Spool recovery attempt interval (hh:mm:ss)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer</td>
<td>Recovery target below TGWARN (0-&gt;50)</td>
</tr>
<tr>
<td>Reset Time</td>
<td>Recovery reset interval (hh:mm:ss)</td>
</tr>
</tbody>
</table>

**Priority of systems for spool recovery:**

| CPUID | 1 | 2 | 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 YES NO

*Figure 130. SPOOLFULL Processing Panel*

The following parameters can be entered:

**Retry Time**
The time entered in this field defines the time between attempts at spool recovery. If the first set of recovery commands has been issued and the spool is not recovered after this amount of time, the second set of defined commands is issued. Time range 00:00:00 to 23:59:59. The default is 5 minutes.

**Buffer**
The percentage amount of spool that is to be recovered under the threshold (TGWARN for SPOOLSHORT, 100% for SPOOLFULL). The value must be positive and less than 75% of the threshold value. If invalid or not specified a default of 5 is used. For example, if TGWARN is set to 80 and Buffer is set to 10, SPOOLSHORT recovery will continue until spool utilization is less than 70%.

**Reset time**
This is the amount of time that must elapse after the spool has been driven below the recovery target before the spool incident is considered to be closed. If the spool exceeds the trigger threshold again before this interval is up, the recovery will begin again with the next pass, rather than restarting from pass 1.

The default is 15 minutes.

**CPUID 1st - 32nd**
An ordered list of system IDs belonging to the JES2 subsystems sharing the JES spool in a MAS definition. During the recovery the JES2 command D MEMBER,STATUS=ACTIVE,NAME is issued and the first machine in the list that is active on the display response is assumed to be responsible for handling the recovery.
JES2 SPOOLFULL Policy Item for JES2 Subsystems

**Note:**
You must keep the CPUID lists for JES subsystems sharing the spool space in synchronization.

If SA z/OS cannot find an active machine in the CPUID list, it will assume responsibility for the recovery itself. This can lead to multiple machines issuing recovery commands if your lists are incomplete.

**Edit Spoolfull Pass Commands**
Entering YES in this field allows you to edit the recovery commands. Figure 131 shows the SPOOLFULL Processing panel where you can enter commands to be issued during SPOOLFULL processing.

![Figure 131. SPOOLFULL Processing Panel for JES2](image)

Commands are issued by passes, so a command to be issued on every pass should be specified without pass specification. Two substitution variables (EHKVAR1 and EHKVAR2) are available. EHKVAR1 contains the current spool utilization, and EHKVAR2 contains the recovery target.

**JES2 SPOOLSHORT Policy Item**
If you select policy item SPOOLSHORT for a JES2 subsystem, SA z/OS displays the SPOOLSHORT Processing panel, as shown in Figure 132 on page 200.
JES2 SPOOLSHORT Policy Item for JES2 Subsystems

The parameters that can be entered on this panel are similar to those described in "JES2 SPOOLFULL Policy Item" on page 198, and the Spoolshort Processing panel is similar to the Spoolfull Processing panel.

JES2 DRAIN Policy Item

If you select the DRAIN policy item for a JES2 subsystem, SA z/OS displays the JES2 DRAIN Processing panel, as shown in Figure 133.

This panel is used to identify which JES resources are to be drained and how they are to be drained if a JES shutdown is unsuccessful:

Initial drain

Choose the resources that are to be stopped with a JES $P command. An initial drain occurs either because of a JES shutdown or an operator issued...
JES2 DRAIN Policy Item for JES2 Subsystems

DRAIN JES. YES indicates that a stop command should be issued for the resource while a NO (the default) does not drain the resource type.

Force drain

Define the drain command to be entered for each resource if JES has not successfully shut down. Possible JES commands could be Restart ($E), Cancel ($C) or any other appropriate command. If NO is entered (the default), this resource type does not have any commands issued for it.

Policy Items for JES3 Subsystems

Figure 134 shows a Policy Selection panel for a JES subsystem.

---

JES3 SPOOLFULL Policy Item

If you select policy item SPOOLFULL for a JES3 subsystem, SA z/OS displays the SPOOLFULL Processing panel, as shown in Figure 135 on page 202.

---

See "Policy Items for Applications" on page 145 for descriptions of most policy items on this panel. The additional policy items are:

JES3 SPOOLFULL

Allows you to specify recovery from JES spool full condition. For more information see "JES3 SPOOLFULL Policy Item."

JES3 SPOOLSHORT

Allows you to specify recovery from JES spool short condition. For more information see "JES3 SPOOLSHORT Policy Item" on page 202.

---
JES3 SPOOLFULL Policy Item for JES3 Subsystems

Commands are issued by passes, so a command that is to be issued on every pass should be specified with an asterisk * in the Pass field. Two substitution variables are available: EHKVAR1 contains the current spool utilization and EHKVAR2 contains the recovery target.

JES3 SPOOLSHORT Policy Item
If you select policy item SPOOLSHORT for a JES3 subsystem, SA z/OS displays the SPOOLSHORT Processing panel, as shown in Figure 136.

Figure 135. JES3 SPOOLFULL Processing Panel

Commands are issued by passes, so a command that is to be issued on every pass should be specified with an asterisk * in the Pass field. Two substitution variables are available: EHKVAR1 contains the current spool utilization and EHKVAR2 contains the recovery target.

JES3 SPOOLSHORT Policy Item
If you select policy item SPOOLSHORT for a JES3 subsystem, SA z/OS displays the SPOOLSHORT Processing panel, as shown in Figure 136.

Figure 136. JES3 SPOOLSHORT Processing Panel

Commands are issued by passes, so a command that is to be issued on every pass should be specified with an asterisk * in the Pass field. Two substitution variables are available: EHKVAR1 contains the current spool utilization and EHKVAR2 contains the recovery target.
Policy Items for CICS Subsystems

Figure 137 shows the Policy Selection panel for applications of type CICS (see the Application Type entry field in Figure 94 on page 141). This Policy Selection panel contains additional policy items.

The CICS specific policy items are:

**CICS CONTROL**
With this policy item you can define the extended subsystem information required for CICS Automation. For more information, see "CICS CONTROL Policy Item."

**CICS CONNECTION**
With this policy item you can select CICS Link Monitoring definitions required for CICS Automation. For more information, see CICS Connection Policy Item on "CICS CONNECTION Policy Item" on page 250.

**CICS CONTROL Policy Item**
With this policy item you can define the extended subsystem information required for CICS Automation, as shown in Figure 138.

**Subtype**
This parameter indicates the resource type. It can be entered in the Subtype field of the APPLICATION INFO policy item of the application.
Policy Items for CICS Subsystems

instance or application class. The value shown is either for the application instance or the application class that the application is linked to.

**ApplID**

The following settings are available:

**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 ACBNAME 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 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.

**Keypoint**

This specifies whether or not a CICS keypoint is required and whether or not Indoubt UOWs can exist. If YES is specified and a keypoint is missed, warning message EVE677 is issued during shutdown. The status will also be changed to STOPPED if SA z/OS determines that an immediate restart of CICS is unlikely. The default value is NO.

**Reload transaction**

This specifies a reload transaction to reload the CICS message exit policy on configuration refresh (INGAMS REFRESH). The default value is SARL if nothing is specified (and the subtype is not CMAS). Enter NONE if no reload transaction is desired.

**CICSPLEX/SM CMAS Region**

This specifies the CICSPlex/SM CMAS region for shutdown purposes.

The extended CICS subsystem automation that you define here is also described in the CICSCNTL section in IBM Tivoli System Automation for z/OS Product Automation Programmer's Reference and Operator's Guide.

**Policy Items for DB2 Subsystems**

Figure 139 on page 205 shows the Policy Selection panel for applications that have been created with application type DB2 (see the Application Type field in Figure 94 on page 141).
The DB2 specific policy items are:

**DB2 CONTROL**

With this policy item you can specify additional data for a DB2 subsystem of subtype MSTR. For more information see "DB2 CONTROL Policy Item."

**DB2 CONTROL Policy Item**

If you select DB2 CONTROL policy item from the Policy Section panel for DB2 subsystems, and you have specified the subtype MSTR, Figure 140 is displayed where you describe the DB2 master address space:

![Figure 139. Policy Selection Panel for DB2 Subsystems](image)

---

**Figure 140. DB2 Control Entries Panel**

On this panel, either all the fields are inherited from linked classes all together and displayed in a different color, or none of them are.

You can find out which class a value has been inherited from by typing the command `classname` (or its abbreviation `cn`) on the command line.
DB2 CONTROL Policy Item for DB2 Subsystems

You can override any inherited values by overtyping them. If you override any values for a class, all the inherited values become specific to that particular class or instance (and thus change color to reflect this).

**Subtype**
This parameter shows the subtype and is always MSTR. It can be entered in the **Subtype** field of the APPLICATION INFO policy item of the application instance or application class. The value shown is either for the application instance or the application class that the application is linked to.

**Subsystem ID**
This entry should correspond to the DB2 subsystem ID as defined in the system PARMLIB.

You can include system symbols and system automation symbols. See "Assigning System Automation Symbols (AOCCLONE)" on page 8.

**Active Log Dataset**
The name of the active log data set to check against before the command to message DSNJ002I is issued. This value is compared with the data set name in the message text. The specified value is implicitly appended with a wildcard character. You can include system symbols and system automation symbols.

**Log Full Threshold**
How full that the active log data set is allowed to be before the command corresponding to the DSNJ110E message is issued. The default value is 90%.

**STOP Tablespace Delay**
This is the delay between STOP tablespace request cycles. Up to 3 cycles are performed to ensure that the tablespace is shut down. 2 minutes (00:02:00) is the default.

**Active Log Alert**
This is the amount of time after which the DSNJ111E alert is reset. The default value is 5 minutes (00:05:00).

**Log Alert Threshold**
This is the number of DSNJ111E alerts to allow for before issuing any associated commands. 5 alerts is the default.

**Log Offload Interval**
This is the amount of time to allow between DSNJ115I messages before issuing an alert plus any associated commands for the DSNJ115I message entry. 30 minutes is the default.

The following fields are for pre SA z/OS 3.3 releases only:

**Shutdown Indoubt**
This indicates whether indoubt threads should be checked for before shutting down DB2. Enter YES if the shutdown should proceed without checking for indoubt threads. If you specify NO, using precheck on the NetView shutdown panel still allows checking for indoubt threads.

**Connection Monitor Delay**
This is the delay between connection monitoring cycles. Default value is 30 minutes (00:30:00).
DB2 CONTROL Policy Item for DB2 Subsystems

Terminate Thread Delay
This is the amount of time to delay between thread termination cycles.
Default value is 2 minutes (00:02:00).

Terminate Thread Cycles
This is the number of cycles to attempt at terminating threads. Default values is 6 cycles.

Policy Items for IMS Subsystems
Figure 141 shows the Policy Selection panel for applications that have been created with application type IMS (see the Application Type entry field in Figure 94 on page 141). 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 item IMS CONTROL is available.

IMS CONTROL
Use this policy item to define additional IMS information for an IMS subsystem of subtype CTL. This policy item is only defined for an IMS control region. For more information see "IMS CONTROL Policy Item."

For more information see IBM Tivoli System Automation for z/OS Product Automation Programmer's Reference and Operator's Guide.

IMS CONTROL Policy Item
If you select the IMS CONTROL policy item from the Policy Selection panel for IMS subsystems, and you have specified the subtype CTL, Figure 142 on page 208 is displayed where you describe the IMS/DC control region.
The fields that you enter or update have the following meaning:

**Subtype**
This parameter shows the subtype and is always CTL. It can be entered on the APPLICATION INFO policy item of the application, or the application class in the Subtype field. The value shown is either the one specified for the application or the application class that the application is linked to.

**Subsystem ID**
Specifies the subsystem ID (IMSID) specified in the IMS control region parameters or SYSGEN. You can include system symbols and system automation symbols, see “Assigning System Automation Symbols (AOCCCLONE)” on page 8.

**IMSPlex name**
Specifies the IMSPlex name that is defined in the z/OS PROCLIB members. You must not specify an IMSPlex name if you do not have the Common Service Layer (CSL) infrastructure in place.

The following field is for releases earlier than SA z/OS 3.3 only:

**DBCTL control region**
Specify YES to indicate that this is an IMS DBCTL control region, or NO to indicate that it is an IMS DB/DC control region.

### Policy Items for TWS Subsystems
Figure 143 on page 209 shows the Policy Selection panel for applications that have been created with application type OPC (see the Application Type entry field in Figure 94 on page 141). This Policy Selection panel contains additional policy items for an TWS subsystem.

These policy items are only applicable to an TWS subsystem defined as an instance. If a subsystem CLASS of type OPC is selected, no additional policy items are available.
The TWS specific policy item is:

**OPC CONTROL**

Use this policy item to specify TWS extensions to subsystem definitions. For more information, see ["OPC CONTROL Policy Item"](https://example.com).

**OPC CONTROL Policy Item**

If you select the OPC CONTROL policy item on the Policy Selection panel for TWS subsystems, SA z/OS displays the panel in Figure 144 where you can describe additional information for TWS subsystems.

**Figure 143. Policy Selection Panel for TWS Subsystems**

The TWS specific policy item is:

**OPC CONTROL**

Use this policy item to specify TWS extensions to subsystem definitions. For more information, see ["OPC CONTROL Policy Item"](https://example.com).

**OPC CONTROL Policy Item**

If you select the OPC CONTROL policy item on the Policy Selection panel for TWS subsystems, SA z/OS displays the panel in Figure 144 where you can describe additional information for TWS subsystems.

**Figure 144. OPC control specifications Panel**

**Subtype**

This parameter shows the subtype that was specified in the APPLICATION INFO policy item for the application itself or application class that this application is linked to.

Fill in the following fields:

**Subsystem ID**

Enter the TWS subsystem ID.

**Controller ID**

Enter the TWS subsystem ID of the controller that is controlling this tracker. This field is only used for tracker definitions. It allows for the specification of a remote controller. If the controller is not in the same sysplex as the tracker, specify this field.

**Note:** Remote controllers with the hot standby function are not supported.
Controller LU Name

Enter the Logical Unit name of the controller that is controlling this tracker. This field is only used for tracker definitions. It allows for the specification of a remote controller. If the controller is not in the same sysplex as the tracker, specify this field.

You do not need to specify the Controller ID and Controller LU Name fields if the controller is in a remote sysplex or system that can be controlled by SA z/OS with the TARGET= parameter of the commands.

Policy Items for USS Subsystems

Figure 145 shows the Policy Selection panel for applications that have been created with application type USS (see “Application Type” on page 142).

USS Applications must be defined with a HASPARENT relationship to JES.

By using the USS CONTROL policy item you can define UNIX System Services-specific data.

<table>
<thead>
<tr>
<th>Action</th>
<th>Policy Name</th>
<th>Policy Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSS CONTROL</td>
<td>Define USS specific data</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 145. Policy Selection Panel for USS Subsystems

USS CONTROL Policy Item

If you select the USS CONTROL policy item on the Policy Selection panel for USS subsystems, SA z/OS displays the panel shown in Figure 146 and Figure 147 on page 211 where you can enter additional information for UNIX subsystems.

Figure 146. USS Control Specification Panel for Application Classes
If either of the values for the User ID or Monitoring Command has been inherited from linked classes, they are displayed in a different color. You can find out which class a value has been inherited from by typing the command `classname` (or its abbreviation `cn`) on the command line, moving the cursor to the field of interest and pressing Enter. You can override any inherited values by overtyping them. If you override either of the values that have been inherited, they are individually applied to the current application and change color to turquoise to indicate this.

For class objects, only User ID and Monitoring Command may be specified.

For instance objects, additionally either Process, File, or Port monitoring must be specified.

**User ID**

Enter the UNIX user ID that this resource belongs to.

**Monitoring Command**

Enter the UNIX System Services monitoring command that is used to get the status of the resource. This command must be a valid program in the HFS. If the field is left blank, the SA z/OS supplied standard monitoring routine is used. The return codes must be one of the following:

- 0 The resource is available.
- 4 The resource is starting.
- 8 The resource is unavailable.
- 12 An error occurred.

If the first character is a `'/`, the specified command is considered to be an absolute path. Otherwise the monitoring routine must be in the same directory as the SA z/OS routines.

For USS Process monitoring specify either one or both of the following fields:
**USS CONTROL Policy Item for USS Subsystems**

**Process Command/Path**
Enter the path of the command that is running, as shown by the UNIX command `ps -e`.

Sample output of UNIX `ps -e` command:

```
PID ... CMD
  16 ... /usr/sbin/inetd
  50331665 ... /bin/sh
  50331666 ... /usr/sbin/sshd
  83886161 ... /SYSTEM/local/WebSphere/V6R1/AP/AppServer/java/bin/java
  50331736 ... /local/WebSphere/V6R1/AP/AppServer/java/bin/java
```

**Filter**
Enter additional filter criteria to uniquely identify the USS process. This field is optional and can only be specified if a Process Command/Path specification is available.

As shown in the sample output for the Process Command/Path field, the command/path specification, especially for Java processes, is not unique. You can use the UNIX command `ps -ef` to obtain the complete definitions.

Sample output of UNIX `ps -ef` command:

```
PID ... CMD
  83886161 ... java -Djava.util.logging.configureByServer=true
  50331736 ... java -Dcom.ibm.eez.adapter.debug=true
```

Specify `-Djava.util.logging.configureByServer=true` as the filter to identify the first Java process listed above.

For USS File monitoring specify the following field:

**File Name**
Enter the path of a file in the HFS. If the file exists and the user ID that you specified is the owner of the file the resource is available, otherwise the resource is not available.

For USS Port monitoring specify either one or both of the following fields:

**IP Stack**
Enter the TCP/IP jobname to indicate the corresponding IP stack.
Complete this field only if your system is configured for multiple IP stacks.
If nothing is specified, the first active TCP/IP job is used. You may include system symbols and system automation symbols.

**Port Number**
Enter the TCP port number that this resource is supposed to listen to on the local host.

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

**Creating a New Event**
To begin defining a new event, access the Events entry type by entering 7 or EVT for events on the Entry Type Selection panel (Figure 1 on page 5).
Events Entry Type

This displays a panel similar to the one shown in Figure 148. This panel may show a list of existing events.

<table>
<thead>
<tr>
<th>COMMANDS</th>
<th>ACTIONS</th>
<th>VIEW</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFGENAM</td>
<td>Entry Name Selection</td>
<td>Row 1 of 3</td>
<td></td>
</tr>
<tr>
<td>Command ===&gt; new</td>
<td>PolicyDB Name : USER_PDB</td>
<td>Enterprise Name : USER_ENTERPRISE</td>
<td></td>
</tr>
<tr>
<td>Action</td>
<td>Entry Name</td>
<td>C</td>
<td>Short Description</td>
</tr>
<tr>
<td>EVENT1</td>
<td>Startup event, backup completed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EVENT2A</td>
<td>Startup event, AOR1 is ready</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EVENT2B</td>
<td>Startup event, AOR2 is ready</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 148. Entry Name Selection Panel for Events

Enter the command new at the command line. A panel similar to the one shown in Figure 149 is displayed. Specify the event name, the UNSET condition, and a description.

<table>
<thead>
<tr>
<th>COMMANDS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOFQYN2</td>
<td>Define New Entry</td>
</tr>
<tr>
<td>Command ===&gt;</td>
<td></td>
</tr>
</tbody>
</table>

Define new entry of type Event

Entry Name . . . . . . EVENT1

Unset Condition . . . NONE (DOWN NONE START UP)

Short description . . . Backup successfully completed

Long description 1 . . .

Long description 2 . . .

Long description 3 . . .

Long description 4 . . .

Long description 5 . . .

Figure 149. Define New Event Panel

The unset condition is preset to NONE. You can change this input field if desired. For more information on the unset condition, see "CONDITION Policy Item" on page 214.

Policy Items for Events

Figure 150 on page 214 shows the policy items that are displayed when an event is selected from the Entry Name Selection panel for events.
Policy Items for Events

The unique policy item for Events is:

CONDITION

With this policy item you can specify when SA z/OS should unset (forget) a previously set event. See "CONDITION Policy Item."

Note:

On the WHERE USED panel for Events, only action “remove” is available.

CONDITION Policy Item

Selecting the CONDITION policy item will bring up a panel as shown in Figure 151 where you can define the unset condition of the current event.

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 Entry Type
Service periods define the availability of resources. A service period is a set of time intervals (service windows) during which a resource should be active or inactive.

Policy Items for Service Periods
Figure 152 shows the policy items that are displayed when an existing Service Periods entry is selected from the Entry Name Selection panel.

<table>
<thead>
<tr>
<th>ACTION</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>---------------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>ADFGEPOL</td>
<td>Policy Selection Row 1 to 5 of 5</td>
</tr>
<tr>
<td>Command ===&gt;</td>
<td>SCROLL ===&gt; PAGE</td>
</tr>
<tr>
<td>Entry Type : Service Period</td>
<td>PolicyDB Name : USER_PDB</td>
</tr>
<tr>
<td>Entry Name : SERVP1</td>
<td>Enterprise Name : USER_ENTERPRISE</td>
</tr>
<tr>
<td>Action</td>
<td>Policy Name</td>
</tr>
<tr>
<td>Description</td>
<td>Enter description</td>
</tr>
<tr>
<td>Service Window</td>
<td>Define service windows</td>
</tr>
<tr>
<td>Where Used</td>
<td>List APLs, APGs and MTRs linked to this entry</td>
</tr>
<tr>
<td>Copy</td>
<td>Copy from existing entry</td>
</tr>
</tbody>
</table>

Figure 152. 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 (or frames) when an application, application group or monitor resource should be available or unavailable. You need to link a service period with a defined service window to Applications, Application Groups, or Monitor Resources.

SERVICE WINDOW Policy Item
Selecting the SERVICE WINDOW policy item displays a panel as shown in Figure 153 on page 216 where you can define the time duration of the service period.
A service period is a schedule of times (referred to as service windows) when a
resource is expected to be active or inactive. Times that you enter into a line
preceded by the letter U will be used as From/To times for the periods when the
resource should be available (UP). A line preceded by the letter D will contain
From/To times for DOWN periods. Start and Stop times are specified in 24 hour
notation, 00:00-24:00, and in any order. For example, to keep an application
continuously available from Monday through Tuesday around the clock, specify
the times as shown:

<table>
<thead>
<tr>
<th>Day</th>
<th>P From</th>
<th>To From</th>
<th>To From</th>
<th>To From</th>
<th>To From</th>
<th>To From</th>
<th>To From</th>
<th>To From</th>
<th>To From</th>
<th>To From</th>
<th>To From</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday</td>
<td>U _</td>
<td>00:00</td>
<td>24:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>D _</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tuesday</td>
<td>U _</td>
<td>00:00</td>
<td>24:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>D _</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wednesday</td>
<td>U _</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>D _</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thursday</td>
<td>U _</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>D _</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Friday</td>
<td>U _</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>D _</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Saturday</td>
<td>U _</td>
<td>08:00</td>
<td>14:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>D _</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sunday</td>
<td>U _</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>D _</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weekday</td>
<td>U _</td>
<td>07:00</td>
<td>18:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>D _</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weekend</td>
<td>U _</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>D _</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Daily</td>
<td>U _</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>D _</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The schedule is defined by specifying one or more (up to five) service windows
for:
- **Each day of the week**: Monday through Sunday
- **Weekday days**: the weekday schedule (Monday through Friday)
- **Weekend days**: the weekend schedule (Saturday and Sunday)
SERVICES WINDOW Policy Item for Service Periods

- **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, you can specify a priority in the P column:

- **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 conflict. The hierarchy of requests and priorities is implemented as follows:

<table>
<thead>
<tr>
<th>Request from</th>
<th>Wins over request from</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service period high</td>
<td>Service period low, INGREQ low</td>
</tr>
<tr>
<td>INGREQ low</td>
<td>Service period low</td>
</tr>
<tr>
<td>INGREQ high</td>
<td>Service period high, service period low, INGREQ low</td>
</tr>
</tbody>
</table>

**Note:**
There are other methods of generating requests within SA z/OS. For a complete description of these methods and the implemented hierarchy of their priorities see the documentation of the SA z/OS concepts in IBM Tivoli System Automation for z/OS User’s Guide.

When defining a service period keep in mind that the startup time is the time at which SA z/OS is to begin startup processing. It is not the time at which the resource will be available. The time until availability depends on how long the startup of an application or an application group takes.

According to the goal driven concept, a vote to start a resource is generated from a service period that you enter into an in a UP service window. A vote to stop a resource is generated from each service period that you enter into a DOWN service window. These votes have to compete with other votes that have been probably issued (for example, due to operator request). At the end of the service period, these votes are removed from SA z/OS. An end of an UP service window does not mean implicitly the start of a DOWN service window.

For more information about the hierarchy of service windows or the syntax of FROM and TO times, see the online help for the product.

If you later change a service period that is already active, and the service period is activated again, this may lead to conflicts and mismatches with overrides that have been made previously. So if you want to change a service period, check your existing NetView overrides that use this service period beforehand, so that no override is outside the new service period (see also the INGSCHED command, described in IBM Tivoli System Automation for z/OS Operator’s Commands).
Triggers Entry Type

In combination with events, triggers are used to control the starting and stopping of resources beyond what is done with specified goals. You use events and triggers if external conditions must be fulfilled (such as a tape mount, for example).

Policy Items for Triggers

Policy Items for Triggers

Condition Policy Item

With the CONDITION policy item you can:

1. Define up to ten startup conditions for the application that you want to connect to the trigger (1), see “Defining STARTUP and SHUTDOWN Conditions for a Trigger” on page 219.

2. Define up to ten shutdown conditions for the application that you want to connect to the trigger (2), see “Defining STARTUP and SHUTDOWN Conditions for a Trigger” on page 219.

The unique policy item for a trigger is:

CONDITION

This policy item allows you to define the startup and shutdown conditions for this trigger. You need to link the trigger with its defined conditions to Applications, Application Groups, or Monitor Resources.
When you define conditions for a new trigger you get an empty Trigger Conditions panel. The entries in Figure 155 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 enter S in front of a STARTUP condition on the panel shown in Figure 155 (1), SA z/OS displays another panel similar to Figure 156 on page 220.
CONDITION Policy Item for Triggers

Figure 156. STARTUP Condition for Trigger Panel

Figure 156 displays a list of events where you can select one or more events that should trigger the startup of an application. Entering S in the Action field selects an event and sets its status to SELECTED. All selected events are added to the startup condition that is currently selected.

Additionally you can enter a description for the startup condition.

If you enter S in front of a SHUTDOWN condition on the panel shown in Figure 155 on page 219, SA z/OS displays another panel similar to Figure 156. The title of the panel is SHUTDOWN Condition for Trigger and the panel contains a list of events that you can add to or remove from the currently selected shutdown condition.

Processor Entry Type

Use the Processor Entry Type to define and control your processor hardware in the sysplex.

Creating a New Processor

How to create new policy objects in general is described in “How to Create New Policy Objects” on page 19.

Figure 157 shows the panel for defining a new Processor policy object.
**Processor Entry Type**

**Name**  The processor name must have 1 to 8 alphanumeric characters and must begin with an alphabetic character, #, $, or @. The name must be different from the names of all groups, subgroups, systems (including the processor operations names), and processors defined in the current policy database.

**Processor Type**  Specifies the processor machine type of the target hardware. Select a name from the list shown to specify the processor type.

Use the generic type name Mainframe to specify the IBM Mainframe family that includes all IBM System z10, System z9, zSeries and S/390 types of processors. The actual processor type is determined at SA z/OS processor operations run time when a target system associated with the target hardware is initialized. The default is Mainframe.

Use the type PSM to specify a ProcOps Service Machine on VM as virtual hardware for VM guest systems. For more details about the PSM, see IBM Tivoli System Automation for z/OS Customizing and Programming.

**Mode**  Specifies the operating mode of the processor hardware as defined in the Reset Profile that is used for hardware activation. For zSeries, 390-CMOS in logical partition mode, and PSM processors specify LPAR. Note that for processors of type PSM, the mode is forced to LPAR.

**Policy Items for Mainframe Processors**

Figure 158 shows the Policy Selection panel for processors.

![Figure 158. Policy Selection Panel for Processor](image)

The unique policy items for the Processor entry type are:

**PROCESSOR INFO**  Using this policy, you can specify the processor type and mode, the connection protocol and network information. See "PROCESSOR INFO Policy Item" on page 222.

**LPARS AND SYSTEMS**  This policy is displayed for processors with mode LPAR. See "LPARS AND SYSTEMS Policy Item" on page 225.

**SYSTEMS**  This policy is displayed for processors with mode ESA. Use this policy to select a system.
**PROCESSOR INFO Policy Item for Mainframe Processors**

**PROCCESSOR INFO Policy Item**
A panel similar to Figure 159 is displayed if you select the PROCESSOR INFO policy item from the Policy Selection panel for Processors.

**Important Note:**
Processors of type NVC are only supported for SA z/OS releases prior to SA z/OS V3R3.

**Figure 159. Processor Information Panel**
The fields at the top of this panel show the following information that was defined when the processor was created:

**Processor Type**
The processor machine type of the target hardware.

**Processor Mode**
The operating mode of the processor hardware as defined in the Reset Profile used for hardware activation.

**Connection Protocol**
Indicates the protocol used for communication with the processor:
- **INTERNAL**
  Connection via the operating system (BCPII).
- **SNMP**
  Connection via TCP/IP.
- **NVC**
  Connection via SNA network.
**PROCESSOR INFO Policy Item for Mainframe Processors**

**HTTP**
Connection via HTTP and enables management of zEnterprise BladeCenter Extension (zBX) resources attached to the processor using the Unified Resource Manager Web Services API.

**Note:** Do not specify HTTP if the Processor is not a member of an Ensemble defined in the policy or if ProcOps is not intended to manage any zBX resources attached to the processor. Specify HTTP if the zBX resources attached to the CPC should be discovered during initialization of the Ensemble connection to manage them later using extended ProcOps commands.

**Additional Notes:**
1. You can also specify multiple values:
   - INTERNAL SNMP
   - INTERNAL HTTP
   - INTERNAL NVC
   - INTERNAL SNMP HTTP
   - SNMP HTTP
2. Combining NVC with SNMP or HTTP is not allowed.
3. If INTERNAL and SNMP are specified and both connections are active, the "ISQCCMD target_system_name" form for LPAR Management can no longer used; instead the form "ISQCCMD target_hardware_name.LPAR_name" must be used. For more information, see the description of the ISQCCMD in [IBM Tivoli System Automation for z/OS Operator's Commands](https://www.ibm.com). 

On the Processor Information panel you can specify the following definitions so that the automation knows where the system is located to send the commands to the appropriate support element.

**CPC Name / NAU**
Indicates the network addressable unit of the support element within the network. This field is required for all connections that are supported.

**Network Name**
Indicates the name of the SNA network that the support element is located in. This field is required for all connections that are supported. For SNMP connections, the central processor complex name of the support element consists of the **Network Name** and **CPC Name**.

**Site/Location Name**
Specify a descriptive name for the processor's location.

The following specification is for INTERNAL processors:

**Auth Token**
The authorization token of the support element. It is defined at the support element, and is used to validate command authorization.

**Connection Monitor Interval**
Indicates how frequently the support element INTERNAL connection status should be monitored. Frequency must be specified using format mm:ss. The allowed minimum is 1 minute (01:00), the maximum is 99
PROCCESSOR INFO Policy Item for Mainframe Processors

- minutes (99:00). If nothing is specified the default value is 5 minutes
  (05:00). To suspend monitoring specify NONE.

  **Note:** At runtime this connection monitor will not monitor the INTERNAL
  connections of processors specified in the GDPS GEOPLEX
  DOMAIN and/or GEOPLEX OPTIONS definitions. These
  connections are exclusively monitored by the GDPS HMC monitor
  function.

The following specifications are for SNMP or NVC processors:

**Community Name / Auth Token**
For SNMP connections, this indicates the community name of the support
element. For NVC connections, it indicates the authorization token. Both
community name and authorization token are defined at the support
element, and are used to validate command authorization. This field is
required.

**ProcOps Target HW Name**
The name that Processor Operations knows the processor by. For
processors with only SNMP or NVC specified as connection protocol, this
name is defaulted to the processor's entry name. If you specify multiple
connection protocols this field is required and must be different from the
processor's entry name.

The name must be unique among the names of zEnterprise ensembles,
groups, subgroups, systems (including both the entry names and
system/image names) and processors (including both the entry names and
the Processor Operations Target HW names) in the current policy database.

**IP Stack**
The name of the IP stack to be used. If not defined, the default IP stack is
used. You may include system symbols and system automation symbols.

  **Note:** Note that this field can only be specified for processors using the
  SNMP protocol.

**Activation Reset Profile**
Indicates the reset profile name to use when activating the target hardware.
This profile name must match with the reset profile name defined at the
target hardware support element. This field is optional. The reset profile
name specified here will be used by Processor Operations common
command processing if the CNAME keyword is omitted from the
ACTIVATE or LOAD command, and use of a reset profile is appropriate.

If you do not specify a reset profile name, and the CNAME keyword is not
used on the LOAD or ACTIVATE command, Processor Operations omits
the CNAME parameter from the command that is sent to the support
element. In this case, the support element uses its default reset profile.

**Path Poll Frequency**
Enter how often (in minutes) SA z/OS processor operations should poll
the support element for status. Valid values are 0 to 99 minutes. Specify 0
to deactivate polling.

**Path Poll Retries**
Enter the number of attempts to retry path polling that should occur before
SA z/OS processor operations notifies the operator that the connection
failed. Valid values are 0 to 99 retries.
Command Retries
The number of retry steps for calling the hardware interface. If nothing is
specified the default value is 10. This field can only be specified for
processors with the SNMP protocol.

Command Retry Wait Time
The wait time between the retry steps for calling the hardware interface. If
nothing is specified the default value is 1 second. This field can only be
specified for processors with the SNMP protocol.

TCP/IP Address or Hostname for Support Element first Adapter/second Adapter
or a HMC
TCP/IP Address or Hostname indicates the TCP/IP address or symbolic
hostname for communication with the target hardware.
At least one address is required for SNMP connections. If you specify two
addresses, the support element first adapter address has preference and the
second adapter address can be used to recover from adapter failures. Use
the Processor Operations ISQIPSWT focal point command to perform the
switch from the first to the second adapter address. Depending on the
Support Element version, a second adapter with an alternate address may
not be supported.
You can choose to use an HMC address if multiple processors should be
controlled via one single connection. This field can only be specified for
processors with the SNMP protocol.

LPARS AND SYSTEMS Policy Item
If you have created a Processor policy object with LPAR mode, the policy item
LPARS AND SYSTEMS is available, and has the LPAR Definitions panel shown in
Figure 160 associated with it.

Figure 160. LPAR Definitions Panel
You can specify the following:
Action
Action allows you to remove LPAR definitions and to invoke the system
selection panel:
LPARS AND SYSTEMS Policy Item for Mainframe Processors

- DELETE (D) deletes the current LPAR definition. Confirmation on a separate panel is not required.
- SYSTEM (S) displays a selection list of all defined systems allowing a system to link to the current LPAR.

**LPAR Name**
Specifies the name of the logical partition. Duplicate LPAR names without Target System Name are wiped out when the panel is left. However, it is possible to link different systems to one LPAR, so duplicate LPAR names are accepted when they have links to different systems. Linking one system to multiple LPARs is not accepted.

**Target Mode**
Specifies the mode of the system that is to run in this LPAR.

The Target Mode LINUX is used for hardware that supports LPARs for Linux only, where a reduced instruction set is provided and therefore cannot run MVS or VM. For a Linux system specify LINUX if the processor hardware is configured in 'LINUX Only' mode, otherwise specify ESA.

Valid LPAR Target Modes are 370, ESA, CF, and LINUX.
Valid VM GUEST Target Modes are 370, ESA, and LINUX.

**Target System Name**
You can specify a system entry name, which establishes a "link" of this system to the partition. Any name is allowed here, it is not required that the system entry has already been defined. During build, such an undefined system is treated as not specified. If the system is already defined as SYS entry, it is checked that it is a valid selection according to the rules described in the following section for entry type SYS. Otherwise, an error message is displayed.

**Policy Items for PSM Processors**
*Figure 161* shows the Policy Selection panel for PSM processors.

<table>
<thead>
<tr>
<th>Action</th>
<th>Policy Name</th>
<th>Policy Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPY</td>
<td>COPY</td>
<td>Copy data from an existing entry</td>
</tr>
<tr>
<td>GUESTS AND SYSTEMS</td>
<td>Define VM guests and select systems</td>
<td></td>
</tr>
<tr>
<td>PSM INFORMATION</td>
<td>Enter and display PSM processor information</td>
<td></td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>Enter description</td>
<td></td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>Enter description</td>
<td></td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>Enter description</td>
<td></td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>Enter description</td>
<td></td>
</tr>
<tr>
<td>Policy Name</td>
<td>Policy Description</td>
<td></td>
</tr>
<tr>
<td>Enterprise Name</td>
<td>USER_ENTERPRISE</td>
<td></td>
</tr>
<tr>
<td>Entry Type : Processor</td>
<td>PolicyDB Name : USER_PDB</td>
<td></td>
</tr>
<tr>
<td>Entry Name : PSMPROC1</td>
<td>Enterprise Name : USER_ENTERPRISE</td>
<td></td>
</tr>
</tbody>
</table>

*Figure 161. Policy Selection Panel for PSM Processors*

The unique policy items for the Processor entry type are:

**PSM INFORMATION**
Using this policy, you can specify the processor type and mode, and network information. See "PSM INFORMATION Policy Item" on page 227.
## GUESTS AND SYSTEMS
This policy allows you to define VM guests and select systems. For more details, see “GUESTS AND SYSTEMS Policy Item” on page 228.

## PSM INFORMATION Policy Item
A panel similar to Figure 162 is displayed if you select the PSM INFORMATION policy item from the Policy Selection panel for PSM Processors.

### COMMANDS HELP

<table>
<thead>
<tr>
<th>Command</th>
<th>ProcOps Service Machine Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entry Type: Processor</td>
<td>PolicyDB Name: USER_PDB</td>
</tr>
<tr>
<td>Entry Name: PSMPROC1</td>
<td>Enterprise Name: USER_ENTERPRISE</td>
</tr>
<tr>
<td>Processor Type: PSM</td>
<td>Processor Mode: LPAR</td>
</tr>
<tr>
<td>Connection Protocol: TCP/IP</td>
<td></td>
</tr>
<tr>
<td>Security Resource Name</td>
<td></td>
</tr>
<tr>
<td>Security Network Name</td>
<td></td>
</tr>
<tr>
<td>Site/Location Name</td>
<td></td>
</tr>
<tr>
<td>Community Name / Auth Token</td>
<td></td>
</tr>
<tr>
<td>IP Stack</td>
<td></td>
</tr>
<tr>
<td>VM Host Target System Name</td>
<td></td>
</tr>
<tr>
<td>Service Machine Name</td>
<td></td>
</tr>
<tr>
<td>Enforce IPv4 for Hostname: YES NO</td>
<td></td>
</tr>
<tr>
<td>Message Server Port: 0 - 65535</td>
<td></td>
</tr>
<tr>
<td>Command Server Port: 0 - 65535</td>
<td></td>
</tr>
<tr>
<td>Path Poll Frequency: 0 to 99 minutes</td>
<td></td>
</tr>
<tr>
<td>Path Poll Retries: 0 to 99</td>
<td></td>
</tr>
<tr>
<td>TCP/IP Address or Hostname</td>
<td></td>
</tr>
</tbody>
</table>

### Figure 162. ProcOps Service Machine Information Panel

The ProcOps Service Machine Information panel shows the Processor Type, Processor Mode and Connection Protocol that were defined when the processor was created. For ProcOps Service Machines the Processor Mode is always LPAR and the Connection Protocol is always TCPIP, and neither value can be changed.

On this panel you can update the following:

**Security Resource Name**
This field corresponds to the CPC Name / NAU name of other processors and is used for SAF checking. This field is required.

**Security Network Name**
This field corresponds to the Network Name of other processors and is used for SAF checking. This field is required.

**Site/Location Name**
Specify a descriptive name for the processor's location.

**Community Name / Auth Token**
The authorization token for TCP/IP connections (ProcOps Service Machines are connected only via TCPIP). This field is required.

**IP Stack**
Indicates the name of the IP stack on the ProcOps Focal Point system that is to be used for this PSM connection.

If defined, the ProcOps FP TCP/IP must be configured for multiple IP stacks. If blank, the default IP stack is used, regardless of the TCP/IP stack configuration.
PSM INFORMATION Policy Item for PSM Processors

You may include system symbols and system automation symbols.
Leave the field blank if the ProcOps FP system is not configured for multiple TCP/IP stacks.

VM Host Target System Name
The name of the VM host system if it is also defined in ProcOps. This field is optional. The number of PSMs per hosting VM system is restricted to 12.

Note: The host command ISQXPSM can only be used to operate the PSM via the VM host if this information is provided.

Service Machine Name
The name of the PSM as it is known to the VM host system.

Note: The host command ISQXPSM can only be used to operate the PSM via the VM host if this information is provided.

Enforce IPv4 for Hostname
You should set this to YES to indicate that the connection to the PSM will use IPv4 in an IPv6-enabled environment. In IPv4-only environments the flag is ignored.

Note: This flag will only be tested at Processor Operations run time if the Hostname address format is used. If nothing is specified the default value is NO.

Message Server Port
The port that the ProcOps Service Machine Message Server uses for communication with the ProcOps receiver task.

Command Server Port
The port that the ProcOps Service Machine Command Server uses for communication with the ProcOps send program.

Path Poll Frequency
How frequently the support element should be queried for status. The frequency must be specified in whole minutes in the range 0–99. A frequency of 0 means that polling is deactivated. This field is required.

Path Poll Retries
The number of times that the status query should be retried before notifying the operator. Valid values are 0–99 retries. This field is required.

TCP/IP Address or Hostname
The TCP/IP address or symbolic host name for communication with the ProcOps Service Machine. This address is required.

GUESTS AND SYSTEMS Policy Item
If you have created a Processor policy object of type PSM, the GUESTS AND SYSTEMS policy item is available, and has the GUEST Definitions panel shown in Figure 163 on page 229 associated with it.
This panel allows you to define VM GUESTs and to link systems to them. Within ProcOps, VM guest systems are treated like LPARs.

**Action**
- Action allows you to remove VM GUEST definitions and to invoke the system selection panel:
  - DELETE (D) deletes the current VM GUEST definition. Confirmation on a separate panel is not required.
  - SYSTEM (S) displays a selection list of all defined systems allowing a system to link to the current VM GUEST

**GUEST Name**
- Specifies the name of the VM GUEST. Duplicate VM GUEST names without Target System Name are wiped out when the panel is left. However, it is possible to link different systems to one VM GUEST, so duplicate VM GUEST names are accepted when they have links to different systems. Linking one system to multiple VM GUESTs is not accepted.

**Target Mode**
- Specifies the mode of the system that is supposed to run on this VM GUEST. The Target Mode LINUX is used for hardware that supports VM GUESTs for Linux only, where a reduced instruction set is provided and therefore cannot run MVS or VM.

For a Linux system specify LINUX if the processor hardware is configured in 'LINUX Only' mode, otherwise specify ESA. Valid VM GUEST Target Modes are 370, ESA, and LINUX.

**Target System Name**
- You can specify the name of the system that runs on this VM GUEST. Any name is allowed here; it is not required that the system entry has already been defined. During build, such an undefined system is treated as not specified.

If the system is already defined as a SYS entry, it is checked that it is a valid selection according to the rules described in the following section for entry type SYS. Otherwise, an error message is displayed.
The number of guest systems per PSM is restricted to 60. When a new guest system is defined, the number of guests that have already been defined for the associated PSM is checked. If there are already 60 defined, any further guest definition is rejected.

Monitor Resource Entry Type

The Monitor Resource (MTR) entry type allows you to obtain the health state of an object in two different ways:

- Actively, by polling, that is, executing a monitoring command periodically
- Passively, by processing events

Monitor resources are connected to application resources (APLs) or application group resources (APGs). The health status of the monitored object is propagated to the APLs or APGs and results in a health status there.

Selecting Monitor Resources (MTR) from the Entry Type Selection panel brings you to the Entry Name Selection panel that lists your current monitor resources, if you have any. From this panel you can either create new monitor resources, or select an entry and action for one of the existing entries.

Policy Items for Monitor Resources

Figure 164 shows the policy items that are displayed when an existing Monitor Resources entry is selected from the Entry Name Selection panel.

The unique policy items for the Monitor Resource entry type are:

**MONITOR INFO**

This policy item allows you to make definitions for the monitor resource. See "MONITOR INFO Policy Item" on page 231 for more details.

**RELATIONSHIPS**

This policy item has the same functionality for entry types Application, ApplicationGroup, and Monitor Resource. For each application, application group, or monitor resource you may specify several relationships. They will be used to automate the startup or stopping of those applications, application groups, and monitor resources. If you select this policy item, in
Monitor Resource Entry Type

the Relationship Selection List panel, you can view existing relationships or define new relationships for the current application, application group, or monitor resource.

You define relationships between dependent resources and supporting resources. In the customization dialog you define relationships always from the point of view of the dependent resource. So the resource that you have currently selected will become the dependent resource, when you define a new relationship.

See “RELATIONSHIPS Policy Item” on page 126 for more information on viewing existing or defining new dependency relationships.

MESSAGES/USER DATA
This policy item allows you to define actions to be taken in response to particular events. Normally, these events are message IDs, but they can also be OMEGAMON exceptions. See “MESSAGES/USER DATA Policy Item” on page 235 for details.

HEALTHSTATE
This policy item allows you to define the processing that takes place when the monitor resource detects a change of the health state of the monitored resource. See “HEALTHSTATE Policy Item” on page 239 and the section “Event and Trigger Support” in IBM Tivoli System Automation for z/OS User’s Guide for more details.

TRIGGER
This policy item allows you to manage the link from the current Monitor Resource entry to the Trigger entry. Each Monitor Resource entry may be linked to one Trigger entry. See “TRIGGER Policy Item” on page 240 and the section “Event and Trigger Support” in IBM Tivoli System Automation for z/OS User’s Guide for more details.

SERVICE PERIOD
This policy item allows you to link a service period to the monitor resource. Its resources are then automatically started up or shut down in the time frame specified by the service period.

MONITOR INFO Policy Item
If you select the MONITOR INFO policy item from the Policy Selection panel for monitor resources, the Monitor Resource Information panel is displayed, as shown in Figure 165 on page 232.
To create a new Monitor Resource entry, provide the following information:

**Monitored Object**
This is the name of the real object that is being monitored by this monitor resource. It is used to locate the monitor resource that is affected by a real monitor event.

For example, the following values can be used:
- OLDS if the monitored object is the IMS Online Log Data Set
- RECON if the monitored object is the IMS Recovery Control Data Set

For more details, see the section “Health Monitoring” in the chapter “How to Monitor Applications” in IBM Tivoli System Automation for z/OS Customizing and Programming.

You can include system symbols and system automation symbols. See “Assigning System Automation Symbols (AOCCLONE)” on page 8 for more details.

**Note:**
If you define the same message ID for different monitor resources, the name in the Monitored Object field must be identical. This is not required for messages ING150I and ING080I.

**Monitored Jobname**
This can be used to specify the name of the job that this monitor resource accepts events from. This ensures that only events that are issued from a specific job trigger automation actions for this monitor resource.

For example, if the monitor routine should only react for the IMS Online Log Data Set of a specific IMS, specify the job name here.
MONITOR INFO Policy Item for Monitor Resources

You can include system symbols and system automation symbols. See "Assigning System Automation Symbols (AOCCLONE)" on page 8 for more details.

Activate Command
This can be any NetView command. It is executed when the monitor resource is started.

You can include system symbols and system automation symbols but they must be entered in uppercase letters (for example, &AOCCLONE1). See "Assigning System Automation Symbols (AOCCLONE)" on page 8 for details on using system automation symbols.

Deactivate Command
This can be any NetView command. It is executed when the monitor resource is stopped.

You can include system symbols and system automation symbols but they must be entered in uppercase letters (for example, &AOCCLONE1). See "Assigning System Automation Symbols (AOCCLONE)" on page 8 for details on using system automation symbols.

Monitor Command
This can be any NetView command. If a command and an interval is specified, the command will be executed periodically while the monitor is active.

If a command is specified and no interval is specified, the command will be executed once during the startup phase.

If no command is specified the monitor is a passive monitor. In this case, the health status must be set via the INGMON command (see IBM Tivoli System Automation for z/OS Programmer’s Reference for more details).

You can include system symbols and system automation symbols but they must be entered in uppercase letters (for example &AOCCLONE1). See "Assigning System Automation Symbols (AOCCLONE)" on page 8 for more details.

Monitoring Interval
Specifies an interval time for periodic execution of the monitor command. If you leave the field blank, no periodic monitoring is performed. If you specify an interval, you must also specify a monitor command.

Captured Messages Limit
Specifies the maximum number of history messages that will be saved for the SA z/OS DISPMT command. If the field is left blank the value can be inherited from the system defaults definition. If it is also not specified there, a default value of 0 will be used.

Desired Available
Lets you specify the default desired status of the resource. The desired state of each resource is either Available or Unavailable, which is the goal that automation tries to achieve.

You can specify the following values:

ALWAYS
The desired state of the resource is set to Available, unless it is dependent on a resource that has a Desired Available setting of ONDEMAND. In this case the resource behaves as if it had a Desired Available setting of ONDEMAND itself.
This is the same behavior as in earlier releases of SA z/OS.

**ONDEMAND**

If there is *demand* for the resource to be available, its desired status is set to Available, otherwise its desired status is set to Unavailable. Demand arises either from propagated MakeAvailable votes or implicitly through membership of a non-passive basic application group (APG) that has a desired status of Available. Demand does not arise from dependent resources with a Desired Available setting of ALWAYS.

A MakeAvailable vote that is propagated to the resource overrides any demand considerations.

An active ONDEMAND member of a move or server group is always sent a vote that sets its desired status, thus overriding any demand considerations.

**ASIS**

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

If you leave the field blank the value can be inherited from the system defaults definition. Refer to "System Defaults Entry Type" on page 268 for further details.

**Note:** If nothing is specified or inherited the default value is ALWAYS.

**Inform List**

This field allows you to specify:

- Where the application is registered to
- Where all status changes are propagated to

Valid receivers are SDF, NMC, IOM, SMF, EIF, TTT and USR. You can specify more than one receiver. The list of values can be blank or comma separated.

If the field is left blank the value is inherited from the systems default definition (see "System Defaults Entry Type" on page 268).

If the resource should not be registered at all, specify NONE. This will prevent inheritance from the systems default definitions.

**Owner**

Specifies information for the operator about who to contact in case of error.

**Info Link**

This field can be used to specify a location (for example, a URL) where additional information about the application can be found.

**Runtokens**

This field is used to specify one or more blank delimited tokens (each up to 20 characters long) to define the runmode, for which the monitor resources qualifies.

**Note:** There is no verification process to confirm whether the specified runtokens are in use by a specific runmode.
MESSAGES/USER DATA Policy Item for Monitor Resources

MESSAGES/USER DATA Policy Item

In entry type MTR, the MESSAGES/USER DATA policy item allows you to specify OMEGAMON exceptions and messages in the Message ID field, as shown in Figure 166. Note that the ‘+’ sign and the imbedded blank must be specified to distinguish the exception from regular messages.

With an OMEGAMON exception in the Message id field, you can use the line command Cmd to specify commands, Cod for codes, Usr for user data, and A for health states. This results in a default automation table entry that is processed whenever message ING080I is issued on behalf of the subject monitor and exception. If the defaults provided by the customization dialog do not meet your installation’s needs, you can select an override action to modify the AT entry.

You can use the FIND command on the Message Processing panel to search the displayed data for a specific string. For more details, see “How to Find Data in a Policy Item” on page 23.

On the Command Processing panel (accessed via the Cmd line command), you can specify a set of one or more commands to react to the exception represented by ING080I.

You can also define generic key/value pairs (USER DATA) that are available to the called command. SA z/OS provides a predefined key, DISABLETIME, which you can use to determine the time that processing for this particular exception will be disabled. The commands that are specified will be issued, but will only be re-issued if the exception still exists after the time specified with DISABLETIME. This is useful if the recovery for an exception takes a while and no subsequent parallel recovery actions should be taken while the initial recovery is still running.

The A line command allows you to select the health state of the monitor whenever this exception trips and message ING080I is issued. When A is specified, a panel similar to Figure 167 on page 236 is displayed.

Figure 166. Message Processing Panel with OMEGAMON Exception

With an OMEGAMON exception in the Message id field, you can use the line command Cmd to specify commands, Cod for codes, Usr for user data, and A for health states. This results in a default automation table entry that is processed whenever message ING080I is issued on behalf of the subject monitor and exception. If the defaults provided by the customization dialog do not meet your installation’s needs, you can select an override action to modify the AT entry.

You can use the FIND command on the Message Processing panel to search the displayed data for a specific string. For more details, see “How to Find Data in a Policy Item” on page 23.

On the Command Processing panel (accessed via the Cmd line command), you can specify a set of one or more commands to react to the exception represented by ING080I.

You can also define generic key/value pairs (USER DATA) that are available to the called command. SA z/OS provides a predefined key, DISABLETIME, which you can use to determine the time that processing for this particular exception will be disabled. The commands that are specified will be issued, but will only be re-issued if the exception still exists after the time specified with DISABLETIME. This is useful if the recovery for an exception takes a while and no subsequent parallel recovery actions should be taken while the initial recovery is still running.

The A line command allows you to select the health state of the monitor whenever this exception trips and message ING080I is issued. When A is specified, a panel similar to Figure 167 on page 236 is displayed.
Enter AS on this panel to display the AT Status Specification panel, as shown in Figure 168. Select the WARNING monitor status.

Pressing PF3 returns you to Message Automation Overview panel, where you can now see that the AT entry that will be generated for the + XCHN message is:

```
Generated AT entry:
IF TOKEN(5) = 'XCHN' THEN
  EXEC(CMD('INGMON 'MTRNM',STATUS=WARNING') ROUTE(ONE %AOFOPGSSOPER%));
```

Note:
The MTRNM variable is available for all exceptions that cause an ING080I message to be created.

The following series of panels shows a sample specification with different sets of commands based on attributes contained within the exception message.

As an example, the XCHN exception is used. A code entry is specified to return the value CONFIG whenever channel path 26 is found to be not available. The appropriate reaction is to issue the command MYCMD to fix it. In addition, and for all other channels that are not available, the administrator should be notified.
Figure 169 shows the input for the Cmd line command.

With action COD on the Message Processing panel of Figure 166 on page 235, a table of codes can be specified that will be evaluated by INGMON to select the commands to issue based on the actual setting of one or more codes. Figure 170 shows an example where the value CONFIG is returned whenever code1 has a value of 26. In this case, it is channel path 26 because the XCHN exception is being processed.

A row is matched whenever the conjunction of code1, code2, and code3 evaluates to true. Otherwise, the next row is evaluated for a match. If no match is found, the value returned will be ALWAYS.

With action USR on the Message Processing panel of Figure 166 on page 235, a key/value pair can be specified that denotes the administrator to be notified, as shown in Figure 171 on page 238. (There must be a REXX exec that performs this function.)
You can also use the Command Processing panel to set up Pass Processing to make multiple attempts to recover from or resolve the exception. PASS counts from 1 to 99 and is incremented for each instance of the message or exception, unless recovery is disabled. In the example in Figure 172 when the number of outstanding replies exceeds the OMEGAMON threshold:

- The first time, issue MYCMD and pass parameter 'ORANGE'
- The second time, do nothing
- The third time, issue MYCMD and pass parameter 'RED'

Note that a blank pass means that the command is always executed.

The USR action can also be used to disable further processing within INGMON using the DISABLETIME keyword, as in Figure 173 on page 239. The data value is the time, which can be specified in the normal NetView format: hh:mm:ss, mm:ss, :ss, or mm..

A DISABLETIME of 0 means that the message processing for this particular combination of monitor and exception is disabled forever. If message processing should be enabled again or prematurely, an INGMON CLEARING request must be issued. It is best issued from within the command that handles the resolution.
MESSAGES/USER DATA Policy Item for Monitor Resources

Action OVR on the Message Processing panel of Figure 166 on page 235 needs to be used to manually refine the NetView AT-entry in order to exploit the code entry. SA z/OS generates a default condition and action statement that can be modified by the installation. In the example, the condition statement would be modified to set a variable named MISSCHAN:

```
IF TOKEN(2)='MONITORS/MTR/SYS1' &
   TOKEN(5)='XCHN' &
   TOKEN(10)=MISSCHAN
```

The action statement would be modified to say:

```
EXEC(CMD 'INGMON MONITORS/MTR/SYS1 CODE1='MISSCHAN)
```

OVR can also be used to add the NetView THRESHOLD condition item to the condition statement suggested by SA z/OS. The THRESHOLD condition item counts the number of occurrences within a given time interval and returns a value of true (1) only if the count is equal to or higher than the specified threshold.

**HEALTHSTATE Policy Item**

If you select the HEALTHSTATE policy item from the Policy Selection panel for monitor resources, the Command Processing panel is displayed, as shown in Figure 174.
HEALTHSTATE Policy Item for Monitor Resources

You can use this panel to enter commands to be executed when the monitor resource detects a change in the healthstate of the monitored resource. The following fields are available:

**State**  Defines the health state that the monitored resource must change to in order to initiate the specified commands. Valid values are:
- NORMAL
- WARNING
- MINOR
- CRITICAL
- FATAL
- blank

Leaving this field blank means that the specified commands will be executed in any health state change.

**AutoFn/**
Enter the name of the automated function that the command is to run under. The AT entry that is generated does not reflect the automated function that you enter, however the execution of the command is routed internally to the automated function that you specify. If you leave this field blank, the function defaults to the one that is currently running the command list. SA z/OS automated functions are mapped to NetView automation operators.

**Note:** Entering * enables return-code checking for the specified command. This means that processing continues if the command returns a return code of 0. In all other cases processing is stopped immediately. If you are running in LOG assist mode, return code checking has no effect because the commands are not executed.

**Command Text**
Enter the command or command list to be issued. You must preface a system command with MVS.

You can use line commands in the **Cmd** field to manipulate the entries on this panel as described in "Line Commands on Message Processing Panels" on page 26.

You can use the FIND command on the Command Processing panel to search the displayed data for a specific string. For more details, see "How to Find Data in a Policy Item" on page 23.

A healthstate can be specified on the call to ACFCMD, which will issue the corresponding command or Command Group defined in the CMD entry.

A healthstate of blank can be specified to ensure that the corresponding command is issued regardless of the SEL= parameter included on the ACFCMD call. Thus, for any given healthstate processing, the blank entries will be issued in addition to the entries coded for that particular healthstate.

**TRIGGER Policy Item**
The Trigger for Monitor Resource panel Figure 175 on page 241 is displayed if you select policy item TRIGGER from the Policy Selection panel for monitor resources. For more details about triggers, see the section “Event and Trigger Support” in IBM Tivoli System Automation for z/OS User’s Guide.
TRIGGER Policy Item for Monitor Resources

This panel allows you to manage the link (only one is allowed) between the current Monitor Resource entry and a Trigger entry.

zEnterprise Ensembles Entry Type

Use the zEnterprise Ensembles (ENS) Entry Type to define and control your groups of processing power for the zEnterprise zBX compute systems in your enterprise.

Creating a New zEnterprise Ensemble

How to create new policy objects in general is described in “How to Create New Policy Objects” on page 19.

Figure 176 shows the panel for defining a new zEnterprise Ensemble policy object.

<table>
<thead>
<tr>
<th>COMMANDS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>------------</td>
<td>------</td>
</tr>
<tr>
<td>Define New Entry</td>
<td></td>
</tr>
<tr>
<td>Command ==&gt;</td>
<td></td>
</tr>
</tbody>
</table>

Define new entry of type zEnterprise Ensemble

Entry name .......
Ensemble Name .......
TCP/IP Address or Hostname of primary HMC .

User ID ........
Password ........

Short Description . . .
Long Description 1 ...
Long Description 2 ...
Long Description 3 ...

Figure 176. Defining a New zEnterprise Ensemble Panel

Entry Name

Specifies the name for the Entry Type in the SA z/OS Customization Dialogs. Each ensemble must be assigned a unique, non-blank name. The name must start with an alphabetic character. It must be unique among the names of all zEnterprise ensembles, groups, subgroups, systems (including both the entry names and the ProcOps Target HW names) in the current Policy Database.
**zEnterprise Ensembles Entry Type**

**Ensemble Name**
Specifies the 16 character ensemble name as defined on the ensemble HMC. Any character is valid.

**TCP/IP Address or Hostname of Primary HMC**
Specifies the TCP/IP address or symbolic hostname of the primary ensemble HMC where the zManager Web Service API is hosted.

**User ID**
Specifies the name of the HMC user to be associated with the new HMC session. It must be defined on the HMC. It is used to logon and to initialize the connection to the HMC. The minimum length is 4 characters.

**Password**
Specifies the password used to authenticate the HMC user identified by the User ID field. It is required. The predefined value SAFPW may be used to allow z/OS to maintain the password of the user ID. In this case, the Ensembles entry type is used to set up the SAF resource identifier in the VSAM file representing the NetView password dataset. Be aware of this when changing, adding or deleting ensemble entries. For more information, refer to the appendix "Security and Authorization" in IBM *Tivoli System Automation for z/OS Planning and Installation*. The minimum length is 4 characters.

**Policy Items for zEnterprise Ensembles**

Figure 177 shows Policy Selection panel for zEnterprise Ensembles.

![Figure 177. zEnterprise Ensemble Policy Items Panel](image)

The unique policy items for the zEnterprise Ensemble Entry Type are:

**ENSEMBLE INFO**
Using this policy, you can specify and display zEnterprise ensemble information. See "ENSEMBLE INFO Policy Item."

**PROCESSORS**
This Policy Item allows you to manage links to Processor entries. See "Processor Entry Type" on page 220 for further information.
**zEnterprise Ensembles Entry Type**

<table>
<thead>
<tr>
<th>COMMANDS HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>--------------------</td>
</tr>
<tr>
<td>zEnterprise Ensemble Information</td>
</tr>
<tr>
<td>Command ===&gt;</td>
</tr>
<tr>
<td>Entry Type : zEnterprise Ensemble</td>
</tr>
<tr>
<td>Entry Name : NEWZGRP</td>
</tr>
</tbody>
</table>

Ensemble Name ........ ensemble01

TCP/IP Address or Hostname of primary HMC
123.456.789.000

TCP/IP Address or Hostname of alternate HMC

User ID............ OPER1
Password........... (Password or SAFPW) ABCDEF

IP Stack...........
Connection Logon....... (YES NO BYPASS)
Path Poll Frequency..... (0 to 359 minutes)
Path Poll Retries...... (0 to 99)
Command Retries....... (0 to 99)
Command Retry Wait Time... (0 to 600 seconds)

**Figure 178. zEnterprise Ensemble Information Panel**

**IP Stack**

Specifies the jobname of the TCPIP (stack) to indicate the corresponding IP stack. Specify this field only if your system is configured for multiple IP stacks. If nothing is specified the first active TCPIP (stack) is used. You may include system symbols and system automation symbols.

**Connection Logon**

Defines how the Ensemble Connection is handled when processor operations is started.

**YES**
If set to YES, an automatic logon with the specified user ID and password is completed, and a connection initialization is attempted, when processor operations is started. Polling will retry until the status is ACTIVE.

**NO**
If set to NO, no logon is done and the connection is not established when processor operations is started. The status is set to UNKNOWN/DORMANT. In this status polling will try to logon and initialize the connection.

**BYPASS**
If set to BYPASS, the connection status is set to CLOSED and polling will ignore the connection.

**Path Poll Frequency**

Defines how frequently the ensemble HMC should be queried for status. It must be specified in whole minutes. Valid values are 0 to 359 minutes. If nothing is specified, the default value is 0 which means that polling is deactivated.

**Path Poll Retries**

Defines the number of query retries before notifying the operator that the connection could not be established. Valid values are 0 to 99 retries. If nothing is specified, the default value is 0, which means no retry is attempted.
**Command Retries**

Defines the number of retry steps for calling the hardware interface. Valid values are 0 to 99 retries. If nothing is specified the default value is 10.

**Command Retry Wait Time**

Defines the wait time in whole seconds between the retry steps for calling the hardware interface. Valid values are 1 to 600 seconds. If nothing is specified the default value is 1 second.

**OPC System Detail Entry Type**

Figure 179 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.

The unique policy item for OPC system details is:

**OPCAO ENVIRON**

This policy item checks the status on START and STOP requests. If you select this policy item, SA z/OS displays the OPC System details panel, which is described in more detail in "OPCAO ENVIRON Policy Item."

**OPCAO ENVIRON Policy Item**

The OPCAO ENVIRON policy item checks the status on START and STOP requests. If you select this policy item from the Policy Selection panel for OPC System Details, SA z/OS displays the panel shown in Figure 180.

The following fields can be filled in:

**Status check**

This entry controls the checking of subsystem status on TWS Automation
OPCA ENVIRON Policy Item for OPC System Details

START, STOP, and RECYCLE requests. If this is not coded, TWS Automation enforces the rule that START operations are ended in error if the subsystem is already UP, and likewise for STOP operations if the subsystem is down. Code REQUEST=NO if you want to have TWS Automation proceed as normal in this case. The default value is YES.

Operation reset delay
This specifies how long the NetView interface to TWS may be unavailable before TWS Automation will not reset operations that ended in error while it was down.

The NetView interface consists of both the NetView SA z/OS TWS Automation address space and its associated NetView SSI address space, over which requests flow from TWS to TWS Automation. When the interface is down, TWS Automation sets any operations destined for a NetView workstation to an error status with an error code of UNTV. If you do not code a value, no operations are reset when the interface becomes available again.

The default value is NEVER.

Batch Interface Server
This specifies the PPI name of the server that accepts commands from batch jobs and executes them on NetView. The default value is EVJCMDRV.

For more information see IBM Tivoli System Automation for z/OS TWS Automation Programmer’s Reference and Operator’s Guide.

OPC Controller Details Entry Type

Figure 181 shows the policy items that are displayed when you create a new OPC controller detail or when you select an existing OPC controller detail from the related Entry Name Selection panel.

Note that if multiple controllers are running on the same system they must share the same Controller Details (OCS) definitions. If more than one OCS policy object is linked to the same system, the last is used.

The unique policy items for OPC controller details are:

OPCA PCS
With this policy item you either specify the NetView domain that the OPC
controller resides on or specify that the local sysplex is to be searched for
the active controller when required. You must also specify the subsystem
name for the OPC controller. For more information, see “OPCA PCS Policy
Item.”

SPECIAL RESOURCES
Use this policy item to select one or more special resources as defined with
the (OSR) Special Resources entry type for the current controller. These
selected special resources for TWS are used on the systems that the current
controller details entry is linked to. (You can view the systems that this
entry is linked to, and link or unlink this entry to systems using the
WHERE USED policy item).

OPCA PCS Policy Item
The panel shown in Figure 182 appears if you select the OPCA PCS policy item
from the Policy Selection panel for TWS controller details.

The OPCA PCS automation definition defines the NetView domain that the TWS
Controller resides on and specifies the MVS subsystem name for the TWS
Controller system.

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 TWS Controller always runs on the specified domain.
The keyword SYSPLEX indicates that the TWS Controller may be run on
any one of the systems in the local sysplex.

OPC controller subsystem
The MVS subsystem name of the TWS Controller as defined in the
IEFSSNxx member.

Note: When defining the TWS Controller to SA z/OS, this name must be
used as the subsystem name and job name.

OPC Special resources
The following entries are valid:

YES  The OPC special resources will be set for any SA z/OS-monitored
subsystem that is specified in the OPC special resource definitions
panel (see Figure 184 on page 248). Ensure that OPC Special
Resources are defined if using this option.
### OPC Special Resources Entry Type

**Figure 183** shows the policy items that are displayed when you select an OPC special resource from the related Entry Name Selection panel.

The unique policy item for an OPC special resource is:

**SPECIAL RESOURCE**

With this policy item you can specify resources as special resources for TWS. SA z/OS will display the OPC special resource definitions panel. For a detailed description of this panel, see "SPECIAL RESOURCE Policy Item."

**SPECIAL RESOURCE Policy Item**

With this policy item SA z/OS is able to globally control the creation and setting of OPC Special resources based on the status of SA z/OS monitored subsystems. This allows TWS to resolve job scheduling dependencies based on the status of SA z/OS automated resources. A group or groups of special resources can be linked to a controller definition defined with option 31 (see "Defining Automation for TWS Components" on page 76).
SPECIAL RESOURCE Policy Item for OPC Special Resources

Fill in your special resource names with up to 24 characters. You can enter wildcard characters and patterns.

OPC Workstation Domains Entry Type

Figure 185 shows the policy items that are displayed when you select an OPC workstation domain from the related Entry Name Selection panel.

The unique policy item for an OPC workstation domain is:

OPCA DOMAINID

If you select this policy item, it leads to the Code Processing panel where you specify how SA z/OS should react if TWS issues the specified message.

For a detailed description of this panel and how to define code processing policy with it, see “Code Processing for OPC Workstation Domains” on page 249.

For more information you should also see OPCA DOMAINID in IBM Tivoli System Automation for z/OS TWS Automation Programmer’s Reference and Operator’s Guide.
You can use this policy item to define the system or sysplex that SA z/OS routes the request that is associated with the message to (see Figure 186).

Use the entry in the Code 1 field, NVxx, to specify a NetView automatic general workstation with a NetView domain index of xx. This is resolved in the Controller NetView into the target NetView system through the definitions in the Value Returned field. These can be:

- The NetView domain ID
- The name of a remote sysplex. SA z/OS determines one of the systems in the remote sysplex that has a Gateway connection with the local system. This system is then used to route the command to.
- SYSPLEX. All SA z/OS systems in the local sysplex are queried for the status of the application that is associated with the job name of the request.
- *LOCAL. The request is processed on the same system that the TWS controller runs on.

Note:
A sysplex name should only be used for commands that operate sysplex-wide.

You can use the FIND command on the Code Processing panel to search the displayed data for a specific string. For more details, see “How to Find Data in a Policy Item” on page 23.

You can use line commands to manipulate the entries. You can also use operators in your code definitions to make comparisons with the values that are provided by the automation procedure. For more details see the online help.

Figure 186. Code Processing Panel for TWS Workstation Domain IDs

You can use the FIND command on the Code Processing panel to search the displayed data for a specific string. For more details, see “How to Find Data in a Policy Item” on page 23.

You can use line commands to manipulate the entries. You can also use operators in your code definitions to make comparisons with the values that are provided by the automation procedure. For more details see the online help.

CICS Connections Entry Type

Figure 187 on page 250 shows the policy items that are displayed when you create or modify a link definition for CICS from the related Entry Name Selection panel.
The unique policy items for CICS connections are:

CICS CONNECTION

This policy item lets you describe the interregion and intersystem communication (IRC and ISC) links so that they can be monitored by SA z/OS CICS Automation. For more information on how to do this, see "CICS CONNECTION Policy Item." See also the section CONNECTION in IBM Tivoli System Automation for z/OS Product Automation Programmer's Reference and Operator's Guide.

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, see IBM Tivoli System Automation for z/OS Product Automation Programmer's Reference and Operator's Guide.

CICS CONNECTION Policy Item

If you select policy item CICS CONNECTION from the Policy Selection panel for CICS links, SA z/OS displays the following panel where you describe the inter-region and inter-system communication (IRC and ISC) links to be monitored by SA z/OS CICS Automation.

CICS Connections Entry Type

![Policy Selection Panel for CICS Connections](image1)

Figure 187. Policy Selection Panel for CICS Connections

Fill in the following fields:
CICS Connections Entry Type

Connection id
The 4 character symbolic name by which this link is known to this CICS subsystem. This is defined on the CICS subsystem itself (SYSID).

Type
Indicates the type of subsystem this link is attached to, valid values are "CICS" or "IMS", this is used only when you want echoplexing to occur.

Echo
depends on the entry in the Type field:

  CICS  The transaction name used to call the echoplex program on the target subsystem.

  IMS   Any non-blank character.

The time parameter is used to define the time delay in seconds after which an echo response must have been received from the remote system before a message is issued. Valid values are from 1 to 99, default 3.

Time zone offset
This parameter specifies the remote systems time offset relative to the UTC (Greenwich mean time). This is applied to the times specified in the DAY keyword to determine when monitoring is active for the remote system. Default value is 0.

Check interval
The time interval between 2 consecutive checks of a connection. Valid values are from 00:05 to 99:59, the default value is 05:00.

Max repair tries
Specifies the number of automatic actions that are to take place after the detection of a connection failure. Valid values are 0 to 9, default value is 3.

Release delay
Specifies the time delay in seconds needed by the system to release a connection. Valid values are 4 to 99, default value is 5.

Acquire delay
Specifies the time delay in seconds needed by the system to acquire a connection. Valid values are 4 to 99, default value is 5.

Critical link
Specifies whether a link failure for this link will change the severity shown on the SDF panels, default value is NO.

**CCN MONITOR PERIOD Entry Type**

Figure 189 on page 252 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.
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 “MONITORING PERIOD Policy Item.”

**MONITORING PERIOD Policy Item**

If you select policy item MONITORING PERIOD from the Policy Selection panel for CICS monitoring periods, SA z/OS displays the following panel where you define the time frames for when the CICS connection link related to the (CCN) Link definition should be active.

For information about valid values and the syntax of the time frames, see the online help.
Messages Entry Type

The Messages entry type shows a list of all messages defined in the policy database together with the messages from MVC entry '+SA_PREDEFINED_MSGS'. This list offers alternative access to the messages definitions with the advantage that it is not necessary to know the entry defined for the message. This policy is the main location for all AT, MRT and MPF related changes.

Messages Policy Items

Figure 191 shows the Policy Items that are displayed when Messages is selected from the entry type Selection panel.

<table>
<thead>
<tr>
<th>ACTIONS HELP</th>
<th>Policy Selection</th>
<th>Row 1 to 3 of 3</th>
<th>SCROLL===&gt; PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command ===&gt;</td>
<td>Policy Selection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Entry Type : Messages</td>
<td>PolicyDB Name : USER_PDB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Entry Name : USER_ENTERPRISE</td>
<td>Enterprise Name : USER_ENTERPRISE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Action</td>
<td>Policy Name</td>
<td>Policy Description</td>
<td></td>
</tr>
<tr>
<td>MESSAGE AUTOMATION</td>
<td>Define AT and MRT data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MPF DEFINITIONS</td>
<td>Define MPF header and trailer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROCESSING OPTIONS</td>
<td>Define Build Options</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 191. Messages Entry Type Policy Selection Panel

The Policy Items for Messages are:

Message Automation

This policy shows a list of all messages defined in the Policy Database together with the messages from special MVC entry "+SA_PREDEFINED_MSGS" which contains the predefined messages that are used for AT build. See "MESSAGE AUTOMATION Policy Item" for further information.

MPF Definitions

This policy allows you to enter or update specifications for MPF header and trailer lines which will be included in the MPFLSTSA member during MPF build. See "MPF DEFINITIONS Policy Item" on page 254 for further information.

Processing Options

This policy allows you to specify whether a default entry in the Message Revision Table is generated for all messages. See "PROCESSING OPTIONS Policy Item" on page 257 for further information.

MESSAGE AUTOMATION Policy Item

This policy item shows a panel similar to Figure 192 on page 254 with all messages in the policy database together with the short descriptions, some status data and a line command area. The list is sorted alphabetically by message.
MESSAGE AUTOMATION Policy Item for Messages

This policy shows a list of all the messages that are defined in the policy database together with the messages from the special MVC entry, +SA_PREDEFINED_MSGS, which contains predefined messages that are used for the NetView automation table (AT) and message revision table (MRT) build.

The messages are sorted alphabetically and shown together with their short description and status data. The AT and MRT status columns show whether there are specifications for the NetView automation table and message revision table build:

- **I** Ignore or Suppress is specified for this message.
- **C** An AT condition is specified for this message.
- **A** A definition for this message causes a specific AT or MRT action. For the AT this is a status selection, for the MRT this is a selection resulting in a REVISE statement.
- **O** An AT or MRT override is specified for this message.
- **P** The message has user AT data and predefined AT data, and the specifications are not checked in detail to find out which AT data combination would be used for AT build.

This panel allows you to select one entry from the list displayed or manage its message definitions with the following command codes:

**WHU** Shows a list of the Applications, MVS Components, and Monitors where the current message is defined.

If you enter the WHU command for a message ID, this displays the Entries linked to Message panel. It shows you all the entries of type APL, MTR, and MVC where the message has been defined, as shown in Figure 193 on page 255.
MESSAGE AUTOMATION Policy Item for Messages

AC Allows you to define the AT entry conditions for the message.
AS Allows you to define the status specifications for the message action.
AO Allows you to specify your own AT entry.
IG Allows you to manage the options to ignore or suppress the message.
MS Allows you to define the specifications that are used to generate the NetView message revision table.
MO Allows you to define your own MRT entry.

You can use the LOCATE (L) command to locate an entry in the list.

Line command (S) leads to the Message Automation Overview panel for that entry, as shown in Figure 194.

The scrollable AT/MRT preview area shows the AT and MRT entries that would be generated based on the current specifications. If, for example, you modify the AT conditions on a subsequent panel and return, the AT preview is updated to reflect the new specifications.

If a generated AT entry is based on an SA z/OS definition (that is, it is defined in the MVC entry +SA_PREFERRED_MSGS), it is shown in the color that has been selected for class data.
MESSAGE AUTOMATION Policy Item for Messages

Note that if a message is only predefined by SA z/OS (that is, it is only provided in MVC entry +SA_PREDEFINED_MSGS), but not specified for any application or MVS component in the policy database, you cannot modify it.

For more details about message automation definitions, see “Message Automation Definitions” on page 174.

For more details about the NetView automation table and message revision table see IBM Tivoli NetView for z/OS Automation Guide.

MPF DEFINITIONS Policy Item

If you select policy item MPF DEFINITIONS from the Policy Selection panel for messages, a panel similar to Figure 195 is shown.

This panel allows you to enter or update definitions that will be included in the MPFLSTSA member during message processing facility (MPF) build.

The message processing facility (MPF) is a z/OS table that screens all messages that are sent to the z/OS console. The MPF compares these messages with a user-defined list of messages on which to automate, suppress from the z/OS console display, or both, and marks messages to automate or suppress.

MPF Header Lines

Specify up to 5 MPF statements that will be included at the top of the MPFLSTSA member, for example:

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

or

```
.DEFAULT,SUP(YES),RETAIN(1,CE),AUTO(YES)
```

You can specify comment lines by including /* as the first 2 characters.

MPF Trailer Lines

Specify up to 3 MPF statements that will be included at the bottom of the MPFLSTSA member.
Note:
Changes in MPF header and trailer lines are taken into account during the 
next build of NetView automation tables when the MPF member is also 
rebuilt.

PROCESSING OPTIONS Policy Item
If you select Policy Item PROCESSING OPTIONS from the Policy Selection panel 
for messages, the following panel is shown.

| Command | HELP |
|-----------------------------------------------|
| MRT Build Default Specification |

PolicyDB Name : USER_PDB   Enterprise Name : USER_ENTERPRISE

The MRT default option controls whether an MRT entry REVISE ('Y' AUTOMATE) 
is generated for messages that will be in the AT but have no user MRT 
specification:

MRT default AUTOMATE . . . . . . . . . . . . . . . . (YES NO)

If nothing is specified here no default MRT entry is generated.

Figure 196. MRT Build Default Specification

This policy item allows you to specify whether a default MRT entry is generated. 
The entry is generated for a message if:

- an AT entry is created in some AT (independent from AT/MRT scope)
- there is no user specification for an MRT entry.

The generated MRT entry looks like the following:

```
UPON ( MSGID = '.........' )
REVISE('Y' AUTOMATE)
```

Timers Entry Type

The Timers entry type allows you to define sets of timers that run commands or 
automation procedures at scheduled times.

Figure 197 on page 258 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.
Timers Entry Type

The unique policy item for Timers is:

**TIMERS**

You define a timer in the Timer specification panel. For more details see "TIMERS Policy Item."

**TIMERS Policy Item**

SA z/OS displays the following panel if you select the TIMERS policy item from the Policy Selection panel for Timers.

![Policy Selection Panel for Timers](image1)

You should note that, at run time, the commands that you define here are converted to the NetView CHRON format.

A timer definition consists of:

**Timer ID**

When a user timer is defined with the SAVE-option outside the control of SA z/OS, it is recommended that a unique timer ID is specified to avoid duplicate timers being created at automation control file (ACF) reload.

**Note:** The uniqueness of the timer ID applies to the system that uses this entry and all other timer entries that are used by the same system.

**Task** Specify the automated function or NetView task name where the command should run. You may also specify a NetView operator group by using a
TIMERS Policy Item for Timers

plus sign (+) as the first character (for example, +OPERGRP). To run under the primary programmed task (PPT) specify PPT.

Note:
Not all commands can run under the PPT.

Clock
Specifies whether the command is to run on a Greenwich Mean Time schedule or the local system time. You can enter either GMT or LOCAL.

Catchup
Specify whether a saved timer should be caught up (that is, it should still be run) after a system outage.

Note:
This parameter will not work for every timer.

Command text
The actual timer-driven command or automation procedure that SA z/OS issues or calls.

The Execution Time field cannot be filled in from this panel, but it shows the time at which the timer has been scheduled. This is the result of completing the Timer Definition panel (selected with an S in the Action column).

Use the S command code to define when the timer executes. This leads to the second Timer Definitions panel, as shown in Figure 199.
You can use only one of the available options to define a timer, and additionally any valid NetView CHRON timer parameters. Note that entering the colons is optional for all the time fields (that is, you can enter, for example, 011000, which is then converted to 01:10:00).

The time and elapsed time can be entered as:
1. Hours, minutes and seconds
2. Hours and minutes
3. Minutes only (hours then defaults to 00)

Valid values are 00-23 (hours) and 00-59 (minutes and seconds). Specifying 24:00:00 is also valid for elapsed time.

Additionally, you can specify Xs for the time to indicate that the command should run at the next increment of time, for example, XX:00:00. This causes the command to run at the beginning of the next hour.

You can define timers in the various sections of the panel, based on the time interval, as follows:

<table>
<thead>
<tr>
<th>Time interval</th>
<th>When the timer should expire</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Number of Days</strong>, based on the elapsed time.</td>
<td>After After a certain number of days and specified time interval.</td>
</tr>
<tr>
<td></td>
<td>Every Periodically, after a specified number of days and specified time interval.</td>
</tr>
<tr>
<td><strong>Day of Week</strong>, based on the day of the week.</td>
<td>On Only once at a certain time within the next seven days.</td>
</tr>
<tr>
<td></td>
<td>Every Once a week at a certain time.</td>
</tr>
<tr>
<td></td>
<td><strong>Every Weekday</strong> At the same time every weekday (Monday through Friday).</td>
</tr>
<tr>
<td></td>
<td><strong>Every Weekend</strong> At the same time every weekend (Saturday or Sunday).</td>
</tr>
<tr>
<td></td>
<td><strong>Every Day</strong> At the same time every day.</td>
</tr>
<tr>
<td><strong>Date</strong>, if the timer is to expire on a particular day.</td>
<td>On On a particular day.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: Only future dates are allowed. If the year is 20 or less it is assumed to be 20xx.</td>
</tr>
<tr>
<td><strong>Time Only</strong>, based on a specified time.</td>
<td>At At a certain time within the next 24 hours.</td>
</tr>
<tr>
<td></td>
<td><strong>After</strong> A certain period of time after the timer has been set.</td>
</tr>
<tr>
<td></td>
<td><strong>Every</strong> At regular intervals. Optionally you may specify a time frame (using the from and end fields) where the timer expires at the regular interval.</td>
</tr>
</tbody>
</table>

The CHRON parameter can be any combination of NetView CHRON timer parameters, such as RECOVERY, NOTIFY, etc. However, the following parameters must not be specified because they are automatically created from the definitions that you make on this panel and the preceding panel:
Tape Attendance Entry Type

This entry type allows you to specify whether you want to monitor tape mounts on this system. If you do want to monitor tape mounts, you must enter warning and alert thresholds (in minutes and seconds) indicating how long a wait is allowed in mounting a tape before the response is considered unsatisfactory.

You can customize SA z/OS to respond to the IEF238D message while tape drives are attended by operators. This can be achieved by specifying an entry with code definitions for the IEF238D message in the MESSAGES/USER DATA policy item of the MVS Component entry type.

The codes that are passed when checking for a code match with this entry are one of the following, depending on which of these terms is included in the IEF238D message:

- The job name as the first parameter, and either DEVICE or WAIT as the second parameter
- The job name as the first parameter, and DEVICE and WAIT as the second and third parameters

Therefore the coded entries should be entered as follows:

**CODE1**
This should contain a pattern match for the job name that needs the response to the IEF238D message.

**CODE2**
This should contain DEVICE or WAIT or an asterisk (*) to allow for both terms.

**CODE3**
This should contain WAIT (if CODE2=DEVICE), an asterisk (*), or can be left blank.

**Value Returned**
This should contain a selection name that has a reply defined for the IEF238D message, depending on the results of the code matching process.

SA z/OS provides a NetView automation table entry that extracts the values from the message that are to be checked for a code match and passes them as CODEEx parameters to the ISSUEACT generic routine. ISSUEACT searches for a matching code definition and issues a reply, depending on the Value Returned of the matching code definition. For further information about message automation see the chapter “How to Add a Message to Automation” in **IBM Tivoli System Automation for z/OS Customizing and Programming**. For more details on the format of the IEF238D message, see the system message documentation for your particular operating system level and release.

This entry type also allows you to set a schedule for unattended tape drives. If a job requests a tape mount at a time defined as unattended, the job will be canceled. **Figure 200 on page 262** shows the policy items that are displayed when a
new Tape Attendance entry is defined or when an existing Tape Attendance entry is selected from the Entry Name Selection panel.

The unique policy item for Tape Attendance is:

ATTEND TIME

This policy item allows you to specify when tapes are attended and whether 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. See ATTEND TIME Policy Item for more details.

ATTEND TIME Policy Item

Figure 201 shows the Environment Definition: TAPES panel.

Fill in the following fields:

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.
ATTEND TIME Policy Item for Tape Attendance

Monitor Tape Mounts
Specify whether tape mounts should be monitored on the SA z/OS workstation. If you enter YES in this field, SA z/OS will automatically create a Tape Mounts object for the NMC workstation. A Tape Mounts object is an icon that represents an unsatisfied tape mount request by a job. When SA z/OS detects a tape mount request, it creates an icon under the Tape_Mount group icon. The icon is green, initially, indicating a satisfactory condition. If the tape mount request is not satisfied before the Warning interval has expired, the icon changes color to white. If the tape mount request is still not satisfied after the Alert interval has expired, the icon changes color from white to red.

The icon is deleted automatically when the requested tape is mounted or the job requesting the tape mount is canceled.

Note: The operator can specify whether the NMC workstation view that contains the tape mount icons is automatically refreshed. It is advisable to set this to refresh automatically so that tape mounts can be monitored without manual intervention.

The name of the tape mount object is of the form:
SSSSS.VVVVVV_DDD(D)
Where:
SSSSS Is the NetView domain name where the mount request was issued. This may be a remote system.
VVVVVV Is the tape volume name
DDD(D) Is the device number that the tape should be mounted on. This device number is either 3 or 4 digits depending on the level of z/OS.

Warning Delay
The delay period, in minutes and seconds, after which the tape mount icon changes to the warning color of white if the tape mount has not been completed.

Alert Delay
The delay period, in minutes and seconds, between the time that the warning delay is reached and the time that the tape mount icon changes to the alert color of red, if the tape mount has not been completed.

If tapes are attended, enter days and times (24 hour clock) when unattended.
These fields are used to enter days and times when tape drives are unattended. A “*” character can be used in the Day field to specify all days of the week.

MVS Component Entry Type
This entry type allows you to apply automation to various z/OS data sets, resources and facilities, which are referred to as MVS components.

You can set error thresholds and automation flags for the following MVS components:
• MVSDUMP (SYS1.DUMP00–xx)
MVS Component Entry Type

- SMFDUMP (SYS1.MANx)
- LOGREC (SYS1.LOGREC)
- SYSLOG (SYS1.SYSLOG.INDX)

In addition, you can customize the recovery procedure that is used for a WTOBUF overflow and automate responses to messages that are issued for MVS components.

Figure 202 is displayed when you select MVS Component from the Entry Type Selection menu.

![Figure 202. Panel for MVS Component Selection](image)

On top of the list there is a special entry +SA_PREDEFINED_MSGS, followed by the user defined entries. This special entry is described in detail in "Predefined SA z/OS Messages" on page 330 of Chapter 8, "The NetView Automation Table and Message Revision Table," on page 327.

Policy Items for MVS Component

Figure 203 shows the policy items that are displayed when a new MVS Component entry is selected or when an existing MVS Component entry other than the special entry +SA_PREDEFINED_MSGS is selected from the Entry Name Selection panel.

![Figure 203. Policy Selection Panel for MVS Components](image)

The unique policy items for MVS Component entry types are:

PREDEFINED MESSAGES

This policy item allows you to specify a display level for the MESSAGES/USER DATA policy and offers a component selection for the
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 "MESSAGES/USER DATA Policy Item" on page 266.

MINOR RESOURCES
This policy item allows you to define flags and thresholds 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 "MINOR RESOURCES Policy Item" on page 267. See also sections “Automation Flags and Minor Resource Flags” and “Thresholds for Minor Resources” in IBM Tivoli System Automation for z/OS User’s Guide.

PREDEFINED MESSAGES Policy Item
If you select the PREDEFINED MESSAGES policy item from the Policy Selection panel for MVS Components (Figure 203 on page 264), the Component Selection panel is displayed, as shown in Figure 204.

This policy item allows you to specify how many messages are displayed in the MESSAGES/USER DATA policy item of this MVC entry, and to control the number of predefined messages that are used for the AT build.
PREDEFINED MESSAGES Policy Item for MVS Components

Display level
Specifies whether the messages that have AT specifications predefined by SA z/OS are displayed in the MESSAGES/USER DATA policy of this MVC entry.

0 Displays only messages defined for this entry. This is the default.
1 Displays the messages that are defined for this entry together with the messages predefined by SA z/OS. For better readability, the predefined messages are displayed in the same color as the inherited data in application policies.

Note that this option only controls the messages that are displayed in the MESSAGES/USER DATA policy. It has no effect on the AT build.

Note
The predefined messages can only be shown if the special MVC entry, +SA_PREDEFINED_MSGS, is linked to the same systems that the current MVC entry is linked to.

Deselect components
Shows a list of those MVS components that predefined messages exist for.

You can select or deselect individual components. Initially all components are selected. If you deselect a component, it is not displayed in the MESSAGES/USER DATA policy item and it is not included in the AT build for the systems that this MVC entry is linked to.

Note that some of the predefined messages do not belong to any of the listed components. These messages that do not have a component relation are always displayed in the policy and used for the AT build. With a display level of 0, deselecting a component has no effect on the messages that are displayed in the MESSAGES/USER DATA policy item, but it does affect the AT build.

You can select or remove single components with the line commands SELECT or REMOVE, or all components with the primary commands SELECT ALL or REMOVE ALL.

MESSAGES/USER DATA Policy Item

If you select policy item MESSAGES/USER DATA from the Policy Selection panel for MVS Components (Figure 203 on page 264), the Message Processing panel is displayed, as shown in Figure 205 on page 267.
MESSAGES/USER DATA Policy Item for MVS Components

On this panel, you can define how SA z/OS reacts to an issued message. Message processing provided here is the same as provided for Applications. For more information, see "MESSAGES/USER DATA Policy Item" on page 165 and IBM Tivoli System Automation for z/OS Customizing and Programming.

MINOR RESOURCES Policy Item

The Minor Resource Definitions panel is displayed if you select the MINOR RESOURCES policy item in the Policy Selection panel for the MVS Component entry type. Refer to "MINOR RESOURCES Policy Item" on page 193 for Applications. Only two flags are supported: (A)utomation and (R)ecovery.

You can use special minor resource names for MVS Component processing, such as:

- **MVSDUMP** for system messages that report that an SVC dump has been made (IEA611E and IEA911E)
- **SMFDUMP** for system messages that report that the smfdump data set is full or has been switched (IEE362A, IEE362I, IEE391A, and IEE392I)
- **LOGREC** for system messages that report that the logrec data set is full or nearly full (IFB040I, IFB060E, IFB080E, IFB081I, and IFC001I)
- **SYSLOG** for system messages that report that the syslog has been queued (IEE043I)
- **LOG** for system messages that report that the syslog has become inactive (IEE037D—followed by IEE043I, IEE533E or IEE769E—and IEE041I).

**MVSCOMP Defaults Entry Type**

MVSCOMP Defaults allow you to set default values for MVS Component sets.

**Policy Items for MVSCOMP Defaults**

Figure 206 on page 268 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.
The unique policy items for MVSCOMP Defaults are:

**MVSESA INFO**
This option allows you to specify a captured messages limit as the maximum number of messages that will be captured for each MVSESA resource while SA z/OS is running. These messages will show up in the SA z/OS command DISPSYS. If the field is left blank the value can be inherited from the system defaults definition. If neither of them is specified, a default value of 0 is used.

You can use the Inform List field to specify where each MVSESA resource is registered to and who all status changes are propagated to. Valid receivers are SDF, NMC, IOM, EIF, TTT, and USR. You can specify one or more values. If the field is left blank the value can be inherited from the systems default definition. If the resource should not be registered at all, specify NONE. This will prevent inheritance from systems default definitions.

**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 the section “Automation Flags” in IBM Tivoli System Automation for z/OS User’s Guide for more information about setting automation flags.

**THRESHOLDS**
This policy item is used to set default error thresholds for MVS Components. The settings specified here will be used if you do not provide specific thresholds for MVS components. See the section “How SA z/OS Uses Error Thresholds” in IBM Tivoli System Automation for z/OS User’s Guide for more information about error thresholds.

More than one MVSCOMP Default may be attached to a system, but it is recommended that you attach only one to avoid data being overwritten during SA z/OS initialization or an INGAMS REFRESH.

---

**System Defaults Entry Type**

This entry type 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.
Policy Items for System Defaults

Figure 207 shows the policy items that are displayed when a new System Defaults entry is defined or when an existing System Defaults entry is selected from the Entry Name Selection panel.

The unique policy items for System Defaults entry type are:

AUTOMATION OPTIONS
This policy item is used to set system-wide options. See “AUTOMATION OPTIONS Policy Item” for more information.

AUTOMATION FLAGS
This policy item allows you to specify automation settings that are used as defaults for the system. The settings specified here are used if you do not provide specific automation settings for those entry types that allow automation. See the section “Automation Flags” in IBM Tivoli System Automation for z/OS User’s Guide for more information on setting automation flags.

THRESHOLDS
This policy item is used to set system-wide error thresholds. The settings specified here are used if you do not provide object-specific threshold settings. See the section “How SA z/OS Uses Error Thresholds” in IBM Tivoli System Automation for z/OS User’s Guide for more information on error thresholds.

More than one set of System Defaults may be attached to a system, but it is recommended that you attach only one to avoid data being overwritten during SA z/OS initialization or an INGAMS REFRESH.

AUTOMATION OPTIONS Policy Item
The System Automation Options panel, as shown in Figure 208 on page 270 is displayed if you select the AUTOMATION OPTIONS policy item on the Policy Selection panel for the System Defaults entry type.
AUTOMATION OPTIONS Policy Item for System Defaults

The following parameters can be specified for system-wide defaults. If the corresponding entry is not specified for individual resources, this value will prevail.

Captured Messages Limit
This specifies the maximum number of messages that will be captured for each resource while SA z/OS is running. These messages will show up in the DISPSYS, DISPINFO, or DISPMTR command output, or the SA z/OS Graphic Interface.

Exceptional Messages Limit
This defines the maximum number of exceptional messages (that is, those with a severity of UNUSUAL, IMPORTANT or CRITICAL) that are captured for the entire system while SA z/OS is running.

These messages are displayed with the SA z/OS INGMSGS command and in SDF, NMC, and TEP.

If you do not specify a value, the default is 300. The maximum is 1020.

Desired Available
Lets you specify the default desired status of the resource. The desired state of each resource is either Available or Unavailable, which is the goal that automation tries to achieve.

You can specify the following values:

ALWAYS
The desired state of the resource is set to Available, unless it is dependent on a resource that has a Desired Available setting of ONDEMAND. In this case the resource behaves as if it had a Desired Available setting of ONDEMAND itself.

This is the same behavior as in earlier releases of SA z/OS.

ONDEMAND
If there is demand for the resource to be available, its desired status is set to Available, otherwise its desired status is set to Unavailable.

Demand arises either from propagated MakeAvailable votes or implicitly through membership of a non-passive basic application group (APG) that has a desired status of Available. Demand does not arise from dependent resources with a Desired Available setting of ALWAYS.

A MakeAvailable vote that is propagated to the resource overrides any demand considerations.
AUTOMATION OPTIONS Policy Item for System Defaults

An active ONDEMAND member of a move or server group is always sent a vote that sets its desired status, thus overriding any demand considerations.

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

**Note:** If nothing is specified or inherited the default value is ALWAYS.

**Prepare Move**
Specifies whether the downtime during the move or switch of a resource should be kept as short as possible. For more details see “Prepare Move” on page 120.

**Move Mode**
Specifies how a move is made from an old member to a new one. For more details see “Move Mode” on page 120.

**Inform List**
This field allows you to specify:
- Where the application is registered to
- Where all status changes are propagated to
- Whether SMF records are written

Valid values are SDF, NMC, IOM, SMF, EIF, TTT, and USR. You can specify more than one value. The list of values can be blank or comma separated. If you do specify a value, no status changes are propagated.

System automation symbols (AOCCLONEs) or system symbols are not allowed.

Application Defaults Entry Type

The Application Defaults 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 that the Application Defaults set is attached to. If the application belongs to an application class with automation flags or error thresholds specified, the class policy takes precedence over the defaults.

**Policy Items for Application Defaults**

Figure 209 on page 272 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.
Application Defaults Entry Type

The unique policy items for Application Defaults policy are:

**APPLICATION INFO**
This policy item allows you to set various defaults for applications that you define to the system. The values entered in the Application Information Defaults panel are used as default values for Application objects that are attached to this system and do not have this information specified. For more details about these parameters, see "Application Entry Type" on page 140.

**AUTOMATION FLAGS**
This policy item allows you to specify automation flag settings that are used as defaults for all applications on the system. The settings specified here are used if you do not provide application-specific automation settings. See the section “Automation Flags” in IBM Tivoli System Automation for z/OS User’s Guide for more information on setting automation flags.

**THRESHOLDS**
This policy item allows you to specify threshold settings that are used as defaults for all applications on the system. The settings specified here override system default settings and are used if you do not provide application-specific threshold settings. See the section “How SA z/OS Uses Error Thresholds” in IBM Tivoli System Automation for z/OS User’s Guide for more information on setting thresholds.

More than one set of Application Defaults may be attached to a system, but it is recommended that you attach only one to avoid data being overwritten during SA z/OS initialization or an INGAMS REFRESH.

Automation Operators Entry Type

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 Entry Type

Automation operators are assigned specific messages to act on. During automated operations, the messages are routed to the appropriate operator using the NetView ASSIGN command.

Policy Items for Automation Operators

Figure 210 shows the policy items that are displayed when a new Automation Operators policy object is created or Automation Operators is selected from the Entry Type Selection panel.

The unique policy item for Automation Operators is:

OPERATORS

This policy item allows you to specify Automation operators. Refer to the section "Automation Operators" in Chapter 2 - Concepts of IBM Tivoli System Automation for z/OS User’s Guide.

Operators Policy Item

Figure 211 shows the Automation Operator panel that you can use to select operators, specify specific messages and link it to console names and NetView operator IDs.

Messages are entered in the Messages for this Operator field. You can specify:
Automation Operators Entry Type

- 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

The order of the automation operators in the list is irrelevant for the execution. NetView ensures that the messages are assigned correctly with the most specific first, followed by the more general. You can use the following action codes to change the order of automation operators in the list:

(M) MOVE
   Moves the automation operator to another position in the list that is marked with BEFORE and AFTER.

(B) BEFORE
   Is used with MOVE. The automation operator is placed before the line marked with the BEFORE action.

(A) AFTER
   Is used with MOVE. The automation operator is placed after the line marked with the AFTER action.

(I) INSERT
   Inserts a new line directly underneath the current one. You can add another automation operator or message on this line.

To continue the specification of message IDs across several lines, insert a line where required and leave the automated function blank.

The following automated functions should not have messages assigned to them on the Automation Operator Definition panel:
- 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.

After you have entered the automated function and assigned messages, a new panel is displayed where you can enter the automation operators and NetView operators that are to receive messages. Figure 212 on page 275 provides an example.
The fields are:

**MVS Console Name**

This is a 1-8 alphanumeric field as defined in the CONSOLnn member of the MVS PARMLIB. The system console name "MASTER" or "ANY" can also be used. Otherwise, the first character of the name must be alphabetic. If console name is specified, the autotask is considered as an attended task. It will be applied to the Primary Automation Operator only.

The name may include any system symbol (for example, &DOMAIN.). System automation symbols (&A0CCLONEx.) are not allowed.

**Automation Operators**

Each automation operator relates to a specific NetView operator ID. Each message is assigned to a list of operators. This list includes the primary NetView ID, the secondary NetView ID (if specified), the primary NetView ID for SYSOPER, and the primary NetView ID for BASEOPER. The message is processed on the first operator in the list that is logged on when the message is received. If none of the operators are logged on, it will be processed on the task that started the system NetView Subsystem Interface Router task (SIR task), if it was started by a NetView task. In the NetView default initialization sequence, AUTO1 starts the SIR task. If the SIR task was not started by an autotask (that is, it has INIT=Y coded in DSIDMN) or the task that started it is no longer active, the message will not be automated. Similar lists of autotasks are defined in AOFMSGSY and are used to route commands to tasks from the NetView automation table.

If a group is specified (the ID starts with a "+"), all the IDs that are assigned to the group are inserted into the list at that point.

You may also use SYSCLONE variables and the &DOMAIN. variable for the automation operators and NetView operators specification.

**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.

---

**Figure 212. Automation Operator Definitions Panel**

The panel shows the definition of an automation operator named AOP_OBJ1 with the following details:

- **Automated Function**: BASEOPER
- **Messages assigned**: Console for NetView cmds
- **Automation Operators**:
  - Primary: AUTBASE
  - Backup: Id1.. Id6.. (Id1 to Id6 are NetView operator IDs)
- **NetView Operators**:
  - Id 1.. Id 6.. (Id1 to Id6 are NetView operator IDs)
Notify Operators Entry Type

The Notify Operators entry type allows you to define notification messages for certain operators.

Policy Items for Notify Operators

Figure 213 shows the Entry Selection panel to define these operators.

NOTIFY OPERATORS

Notification operators are human NetView operators who receive SA z/OS notification messages, which are messages about events SA z/OS detects or actions it takes. These notification messages are routed to one or more logged-on human operators, using the message forwarding path you set up in the Network policy object. The messages can optionally be held until the operator clears them. See "NOTIFY OPERATORS Policy Item."

Figure 213. Policy Selection Panel for Notify Operators

NOTIFY OPERATORS Policy Item

Figure 214 on page 277 shows the Operator Notification panel that you can use to select notification operators or define new notification operators that you want to be notified about selected message classes from applications on a system.
NOTIFY OPERATORS Policy Item for Notify Operators

When you specify Yes in the Assign User message classes field on the Operator Notification Message Classes panel (Figure 215), to have the operator notified about user-defined message classes, SA z/OS displays existing message classes that have been defined with the AUTO MSG CLASSES policy item of the Enterprise entry type (ENT).

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

When defining notification operators, you can customize:
- Which operators receive notification messages
- The types of message held on notification operators' console screens:
  - Immediate Action
  - Immediate Decision
  - Possible Action
  - Information
Notify Operators Policy Item for Notify Operators

- System Wait
- The classes of messages sent to the operators. (See "Auto Msg Classes Policy Item" on page 82 for information about defining your own message classes.) Each SA z/OS message can have one or more message classes associated with it. Similarly, each notification operator can have one or more message classes assigned. When messages are processed during automated or manual operations, the class of each message is compared to the classes defined for each notification operator. If a match occurs, the notification operator receives the message. All one and two digit numeric message classes are reserved for IBM use. You may use message classes made up of one alphabetic and one numeric character (or one numeric and one alphabetic).

Table 7 shows the message classes defined by SA z/OS. Message classes from 80 to 90 are relevant for monitoring the automation manager. These message classes are discussed in more detail in IBM Tivoli System Automation for z/OS Messages and Codes.

Table 7. SA z/OS Message Classes

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

For example, any operator defined with a notification operator class of all SA z/OS messages (class number 40 and 80) receives all SA z/OS messages. Operators responsible for system operations should receive all SA z/OS messages.
NOTIFY OPERATORS Policy Item for Notify Operators

messages or a subset of those messages. As another example, you may want to have your lead operator receive all debug messages.

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

SA z/OS uses notification messages to update the status display facility (SDF).

Network Entry Type

When multiple z/OS systems are interconnected and require consolidated operations at one focal point system, you can use focal point services to monitor and control the systems. These services allow an operator to view the status of multiple systems from a single system acting as a focal point and to send messages, commands, and responses from a focal point system to other systems. The Network entry type allows you to:

- Define paths for sending messages, commands, or responses from one system to another.
- Define primary and backup focal point systems.
- Define user Terminal Access Facility (TAF) fullscreen sessions and specify whether each of these should be initiated automatically when the operator logs on to the SA z/OS NetView Interface. If a TAF session is not defined to start automatically, you can initialize it using the OPER dialog.
- Define gateway sessions connecting systems.

A gateway session uses autotasks both inbound to and outbound from a system to send data between systems.

Communication with the focal point system is performed using the NetView RMTCMD command or XCF. XCF is used when commands are sent to systems within the same sysplex (that is, with the same XCF group ID). Otherwise, RMTCMD is used. In both cases, you can see the responses on your operator task.

- Define OMEGAMON sessions.

OMEGAMON sessions can be used to retrieve exception information or general performance and availability data for health monitoring and proactive automation.

- Manage access keys to the NetView password data set, where passwords can be stored that are needed when OMEGAMON sessions have to be established.
- Define Simple Object Access Protocol (SOAP) servers and their attributes to be able to communicate with IBM Tivoli Monitoring (ITM).

The chapter “Automating Networks” in IBM Tivoli System Automation for z/OS Customizing and Programming provides more information on the defining automation networks.

Policy Items for Network

Figure 216 on page 280 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 NameSelection panel.
The unique policy items for Network are:

**SDF FOCALPOINT**
This policy item is used to define the domain that is to receive the notification messages to be forwarded from the current domain. See "SDF FOCALPOINT Policy Item" for more details.

**FULL SESSIONS**
With this policy item you can define applications that TAF fullscreen sessions can be established to. See "FULL SESSIONS Policy Item" on page 281 for more details.

**GATEWAY**
Each domain that is to receive commands, replies, or notification messages must have a gateway defined for it. Communication with the focal point system is performed via XCF (within the same sysplex) or the NetView RMTCMD command. See "GATEWAY Policy Item" on page 282 for more details.

**OMEGAMON SESSIONS**
This policy item is used to define the attributes used to establish and run a communication session with OMEGAMON based on the NetView Terminal Access Facility (TAF). See "OMEGAMON Policy Item" on page 283 for more details.

**AUTHENTICATION**
This policy item is used to define access keys for particular OMEGAMON application IDs in order to let SA z/OS retain passwords in the NetView password data set. See "AUTHENTICATION Policy Item" on page 286 for more details.

**SOAP SERVER**
This policy item is used to define SOAP servers and their attributes. See "SOAP SERVER Policy Item" on page 286 for more details.

**SDF FOCALPOINT Policy Item**
If you select the SDF FOCALPOINT policy item from the Network Policy Selection panel, the panel shown in Figure 217 on page 281 is displayed.
This option is used to define the domain that is to receive the notification messages to be forwarded from the current domain. Two domains can be specified in the fields of the panel:

**Primary Domain**
This is the domain that serves as the primary focal point for the Status Display Facility (SDF) of SA z/OS.

The name must consist of up to 5 alphanumeric characters or can be &DOMAIN.

*Note:* If nothing specified the default value is &DOMAIN.

**Backup Domain**
This is the domain that serves as the backup focal point for the Status Display Facility (SDF) of SA z/OS and is used if the primary domain is not active.

The name must consist of up to 5 alphanumeric characters or can be &DOMAIN.

*Note:* If nothing specified the default value is &DOMAIN.

### FULL SESSIONS Policy Item
If you select the FULL SESSIONS policy item from the Network Policy Selection panel, the panel shown in Figure 218 on page 282 is displayed.
With this option you can define applications that TAF fullscreen sessions can be established to. 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 Policy Item
If you select the GATEWAY policy item from the Network Policy Selection panel, the panel shown in Figure 219 on page 283 is displayed.
GATEWAY Policy Item for Networks

Each domain that is to receive commands, replies or notification messages must have a gateway defined for it. This includes the domain name for the outbound gateway autotask. To define the gateway, provide the following information:

**Domain**
The name of all the domains that should receive anything forwarded from the current domain.

**Description**
The description of the domain that is to be communicated with through the gateway.

For more details, see the section “Automating Networks” in IBM Tivoli System Automation for z/OS Customizing and Programming.

**OMEGAMON Policy Item**
If you select the OMEGAMON policy item from the Network Policy Selection panel, the panel shown in Figure 220 on page 284 is displayed.

![Figure 219. GATEWAY Definitions Panel](image-url)

Each domain that is to receive commands, replies or notification messages must have a gateway defined for it. This includes the domain name for the outbound gateway autotask. To define the gateway, provide the following information:

**Domain**
The name of all the domains that should receive anything forwarded from the current domain.

**Description**
The description of the domain that is to be communicated with through the gateway.
OMEGAMON Policy Item for Networks

This panel allows you to create, select or delete OMEGAMON sessions.

**Action**
Can be used to select (S) a session for further processing or delete (D) a session.

**Session Name**
A unique, eight character name for the session that you are defining. Any alphabetic, numeric, #, @ and $ character can be used. The first character must be alphabetic.

**Description**
This is optional arbitrary text that describes the purpose of this session.

Selecting a session from the OMEGAMON Session Definitions panel leads to the OMEGAMON Session Attributes panel, as shown in **Figure 221**.

---

**Figure 220. OMEGAMON Sessions Definitions Panel**

**Figure 221. OMEGAMON Session Attributes Panel**
OMEGAMON Policy Item for Networks

This panel allows you to define OMEGAMON sessions together with their attributes.

**VTAM Applid**
This is the name of the OMEGAMON VTAM application as defined by the installation during customization of the OMEGAMON product. The name is 1 to 8 characters long and may contain system and automation symbols.

**VTAM Logmode**
This is the logon mode name designating the device properties for the session setup to the OMEGAMON application. Because the NetView command BGNSESS is used with the ATTACH option, it must be compatible with a 24x80, non-queryable screen. If you do not specify the logmode, the default BIND image table as defined to VTAM is used.

**TAF SRCLU**
This is the name of the source LU that is assigned to this session by the installation. Specify this field if the default names that are created and assigned to a TAF session by NetView do not conform with your network naming conventions. If you leave the field blank, NetView assigns a name.

You can include system symbols and system automation symbols (see "Assigning System Automation Symbols (AOCCLONE)" on page 8).

**Type**
This is the type of OMEGAMON monitor that a session is established with. Valid types are OMIIICICS, OMIIIDB2, OMIIIMS, and OMIIIMVS; they represent the OMEGAMON classic for CICS, DB2, IMS, and MVS, respectively.

**User ID**
This is the ID of the user that logs on to OMEGAMON when a session is established. To establish such a session, the user ID must be authorized to log on to OMEGAMON either within OMEGAMON itself or using an SAF-product.

To issue OMEGAMON commands, the user ID must also be authorized to issue commands that are protected by OMEGAMON internal command security or by the security profiles that are managed by an SAF product. For more details about user authorization, see the security chapter in IBM Tivoli System Automation for z/OS Planning and Installation.

The user ID has a maximum length of 8 characters and may contain system and automation symbols. It can be omitted if no OMEGAMON product level security is required.

**Password**
The password of the user who logs on to OMEGAMON. The password has a maximum length of 8 characters and must be specified if a user ID was specified.

The predefined keyword SAFPW may be used to allow SA z/OS to retain the password in the NetView password data set for the user ID. See also "AUTHENTICATION Policy Item" on page 286.

**Timeout**
This is the maximum time in seconds that SA z/OS waits for a response from OMEGAMON. If the specified timeout expires, the user is logged off and the session is terminated.

Any value between 1 and 999 seconds can be specified. The default value is derived from the setting of the WAITTIME variable (default: 29 seconds) that can be set in CNMSTYLE.
**Session Data**

This is an optional 60-character field of data that is passed unchanged to OMEGAMON during logon. The data field must be used to specify the job name of a particular CICS region or the name of a particular DB2 subsystem that should be monitored by automation, for example:

- `CICS=PRDAOR1` where `PRDAOR1` is the job name of the CICS region
- `DB2=S664` where `S664` denotes the DB2 subsystem

**AUTHENTICATION Policy Item**

If you select the AUTHENTICATION policy item from the Network Policy Selection panel, the panel shown in **Figure 222** is displayed.

![Authentication Definitions Panel](image)

When `SAFPW` is specified as the password for a given OMEGAMON session, SA z/OS retrieves the actual password from the password data set using the user ID that is specified for this session and the owner that is specified in this policy item. For more details about the use of the NetView password data set, see the security chapter in [*IBM Tivoli System Automation for z/OS Planning and Installation*](#).

**Owner**

Custom value of up to 5 characters that is used by SA z/OS to look up the password in the NetView password data set when an OMEGAMON session with any of the VTAM application IDs listed under `Share` is established.

**Share**

This is the list of VTAM application IDs representing OMEGAMON monitors that passwords are kept for in the NetView password data set under the same owner. The application IDs may be delimited by blank or comma.

**SOAP SERVER Policy Item**

If you select the SOAP SERVER policy item from the Network Policy Selection panel, the panel shown in **Figure 223** on page 287 is displayed.
Use this panel to manage your IBM Tivoli Monitoring (ITM) SOAP server definitions.

**Action**
Use this field to select (S) a server for further processing or to delete (D) a server.

**Server Name**
Enter a unique 8-character name for the server. Any alphabetic, numeric, #, @ and $ character can be used. The first character must be alphabetic.

**Description**
Enter descriptive text for the server.

Selecting a server leads to the SOAP-Server Attributes panel, as shown in Figure 224.

![Figure 223. SOAP Server Definitions Panel](image)

Use this panel to manage your IBM Tivoli Monitoring (ITM) SOAP server definitions.

**Action**
Use this field to select (S) a server for further processing or to delete (D) a server.

**Server Name**
Enter a unique 8-character name for the server. Any alphabetic, numeric, #, @ and $ character can be used. The first character must be alphabetic.

**Description**
Enter descriptive text for the server.

Selecting a server leads to the SOAP-Server Attributes panel, as shown in Figure 224.

![Figure 224. SOAP-Server Attributes Panel](image)

**SOAP Server Policy Item for Networks**

**Figure 223. SOAP Server Definitions Panel**

**Figure 224. SOAP-Server Attributes Panel**
SOAP SERVER Policy Item for Networks

Use this panel to define the SOAP server and its attributes. The following fields can be specified:

Host Name
This is the required address of the host where the SOAP server is running. It can be a symbolic host name (which consists of alphanumeric characters and dots) or an IP address. Both IPv4 and IPv6 address formats are allowed.

An IPv6 address consists of 8 hexadecimal qualifiers, each in the range 0–X'FFFF' separated by a colon. You can include system symbols and system automation symbols but they must be entered in uppercase.

An IPv4 address consists of 4 decimal qualifiers, each in the range 0–255, separated by a period. System symbols and system automation symbols are not supported in IPv4 format.

Port Number
This is the port number that the SOAP server is listening to. The default for the ITM SOAP server is 1920.

Protocol
This defines how data is exchanged between SA z/OS and the SOAP server. SOAP-requests can be issued using the HyperText Transfer Protocol (HTTP) or via HTTPS secured through Secure Socket Layer or Transaction Level Security protocols. The default protocol is HTTP.

Absolute Path
This is the absolute path of the SOAP service on the host that completes the address of a SOAP service. If there is no path, specify a forward slash (/) to denote the root directory as the SOAP server path. The default path that is used for ITM SOAP services is ///cms/soap.

For more details about how SA z/OS uses SOAP messages and how to use them see [IBM Tivoli System Automation for z/OS Customizing and Programming](#).

Sysplex Defaults Entry Type

The Sysplex Defaults entry type allows you to define sysplex resource information defaults.

Policy Items for Sysplex Defaults

[Figure 225 on page 289](#) shows the Entry Selection panel to define sysplex resource information defaults.
The unique policy item for the Sysplex Defaults entry type is:

**RESOURCE INFO**
Use this policy to specify where sysplex resources are registered to and how the desired status of the resource is determined in the absence of any propagated DesiredStatus votes on the resource. See “RESOURCE INFO Policy Item.”

**RESOURCE INFO Policy Item**

Figure 226 shows the Sysplex Resource Information Defaults panel that you can use to define where sysplex resources are registered.

**Desired Available**
Lets you specify the default desired status of the resource. The desired state of each resource is either Available or Unavailable, which is the goal that automation tries to achieve.

For further details, refer to “APPLGROUP INFO Policy Item” on page 122.

**Prepare Move**
Specifies whether the downtime during the move or switch of a resource should be kept as short as possible. For more details see “Prepare Move” on page 120.

**Move Mode**
Specifies how a move is made from an old member to a new one. For more details see “Move Mode” on page 120.
RESOURCE INFO Policy Item for Sysplex Defaults

Inform List
This field allows you to specify:
- Where the application is registered to
- Where all status changes are propagated to
- Whether SMF records are written

Valid receivers are SDF, NMC, IOM, SMF, EIF, TTT and USR. You can specify more than one receiver. The list of values can be blank or comma separated.

If nothing is specified, no status changes are propagated.

Resident CLISTs Entry Type

Because NetView automatically loads frequently used REXX scripts (CLISTs) into storage, you do not need to define your own resident REXX scripts. However, if you want to do this, you can define resident CLISTS that will then be automatically loaded during SA z/OS initialization.

A resident CLIST is a REXX script that is loaded into and kept in NetView storage for the first and all subsequent calls.

Resident CLISTS can improve performance by removing the overhead required to access REXX script that are stored on a direct access storage device (DASD) each time the REXX script is called. Resident CLISTS are automatically loaded into main storage, using the NetView LOADCL command, during initialization of the automation environment. See Tivoli NetView for z/OS Automated Operations Network User’s Guide for more information on using the LOADCL command for REXX scripts.

If a new automation control file is loaded after SA z/OS is up and initialized, and that automation control file contains fewer automation procedures in the resident portion than the original automation control file loaded at initialization, 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.

Policy Items for Resident CLISTS

[Figure 227 on page 291] shows the policy items that are displayed when a new Resident CLISTS entry is defined or when an existing Resident CLISTS entry is selected from the Entry Name Selection panel.
Resident CLISTs Entry Type

The unique policy item for Resident CLISTs sets is:

**CLISTS**

This option allows you to enter the names of the REXX scripts (CLISTs) that you want to be made resident during initialization of SA z/OS.

---

**Status Display Entry Type**

The status display facility (SDF) provides a display of automated systems and resources using assigned status colors. An operator monitors the status of automated systems and resources by viewing the SDF main panel. Typically, for example, an application shown in green on an SDF status panel indicates the application is up, while red indicates the application is stopped or in a problem state. Operators can use the SDF to monitor the system and decide which actions to take when problems occur. This section describes the entries used to define the SDF in the customization dialog. You only need to change these entries if you use values other than the SA z/OS-provided defaults. The entries specified here define statuses and the priorities assigned to those statuses. These entries are used by SA z/OS common routines to gather data for requests to add status descriptors to status components. Additional parameters manage existing status descriptors and eliminate redundancy.

If no definitions exist the SA z/OS defaults are in effect. For more details, see [IBM Tivoli System Automation for z/OS User’s Guide](#).

**Policy Items for Status Display**

[Figure 228 on page 292](#) shows the policy items that are displayed when a new Status Display policy object is defined or when an existing Status Display object is selected from the Entry Name Selection panel.
The unique policy item for Status Display is:

**SDF DETAILS**

You can use this policy item to define colors and highlights associated with each status.

See “SDF DETAILS Policy Item” for more information about this policy item.

For more information on SDF customization, see [IBM Tivoli System Automation for z/OS User's Guide](#).

**SDF DETAILS Policy Item**

A panel similar to Figure 229 is displayed if you select the SDF DETAILS policy item from the Policy Selection panel for Status Display.

These entries are show as an example but are not required in the definition. Please refer to the Table in the section “Subsystem Colors and Priorities” of [IBM Tivoli System Automation for z/OS User's Guide](#) for a list of SDF default settings.
SDF DETAILS Policy Item for Status Display

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 that is defined in the AOINIT member. Although the color of the status descriptor is changed, the priority is still determined by the order that the status descriptor is chained in.

There is no default color associated with an SA z/OS status. Any color specifications defined using this parameter override the color defined for the range that the priority is in.

Clear
The CLEAR parameter helps manage redundant status descriptors. With SA z/OS, the reference value on a status descriptor is usually the status component name. This option is only available for statuses defined in the automation control file and accessed by SA z/OS common routines described in IBM Tivoli System Automation for z/OS Programmer’s Reference.

The following values are allowed:

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.

(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.

(Y,RV) Performs the same function as CLEAR=(Y,RV*) but the reference value must be the same (not a wildcard).

(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.)

(Y,RV2*,CO) Performs the same function as CLEAR=(Y,RV*) with the following differences:
  • The color of the existing status descriptor must be the same as that specified in the color entry before it is deleted.
  • Status descriptors that have reference values prefixed with the new reference value followed by the number 2 are deleted, provided that the color matches the status descriptor that is being added.

N Indicates that redundant status descriptors are not deleted.

Chapter 4. Entry Type Reference 293
SDF DETAILS Policy Item for Status Display

**Srv**
Determines the format of the reference value that is generated for this entry.

**Y**
Generates Status_component/Status.

**N or blank**
Generates Status_component.

**Req (noadd)**
Specifies whether the component with this status is to be added to the SDF chain:

**blank**
The entry is to be added. This is the default value.

**NOADD**
The entry is not to be added.

**Note:** Combining NOADD with Clear=Y causes a component to be removed from the SDF display when this status is reached.

User E-T Pairs Entry Type

This entry type allows you to create your own entry-type pairs and associated keywords in the same format as those used by SA z/OS.

You can use User E-T Pairs to store data for your own automation procedures in the automation control file. User E-T Pairs are provided for situations where there is no suitable, predefined SA z/OS entry type. Using the automation control file to store policy-specific information helps keep your automation procedures generic.

Policy Items for User E-T Pairs

Figure 230 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.

---

**Figure 230. 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. See “E-T DATA Policy Item” on page 295 for more information on this policy item.
E-T DATA Policy Item

Figure 231 shows the UET Entry-Type Selection panel.

The Entry Name and Type fields define a unique set of key-data pairs. In general, the Entry Name field can contain any name except subsystem names that already exist in the policy database. The Type field can be used to identify groups of information for the entry name. You can define wildcards and types by appending the asterisk (*) character to the type value.

The following actions are possible:

S(ELECT)
To select the entry-type pair for further processing. Another panel, similar to Figure 232 on page 296 is displayed where the keyword-data pairs can be entered.

E(DIT)
To select the entry-type pair to change the value of any of entry, type, short and long description.

D(ELETE)
To delete the complete entry-type pair with all associated keyword-data pairs.

The following VIEWs are possible:

- Entry-type pairs only
- Entry-type pairs with their short description
- Entry-type pairs with their long description

When you enter an E (for Edit) in the Action column, another panel, similar to Figure 232 on page 296 is displayed where 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 a keyword does not have to be unique; that is, each keyword may have multiple sets of data assigned to it.

The following rules apply to 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.
The UET Keyword-Data Specification panel allows you to manage the list of keyword-data pairs for one entry-type pair of one User E-T pair entry. You can define multiple keyword-data pairs for each entry-type pair. By default, the first 20 characters of the keyword value is shown. The length of the data value that is displayed depends on your screen width (for a 24x80 screen, it is 55 characters).

From this panel you can:
- Specify whether the value entered in the Data field remains unchanged (Mixed case=YES) or is automatically translated to uppercase characters (Mixed case=NO)
- Change the length of the field that the keyword value is displayed in using the Keyword length field

If necessary, use F11 to scroll to the right to view more of the data field.

For a description of further commands on this panel, see “Command and User Data Policies” on page 25.

From within a REXX exec, 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 67.

More than one set of User E-T Pairs may be attached to a system, but it is recommended that you attach only one to avoid data being overwritten during initialization of SA z/OS.
Chapter 5. Policy Database Maintenance

Option 4 (Policies) of the Customization Dialog Primary Menu panel allows you to maintain the list of your policy databases and to access various panels for selected policy databases.

Policy Database Selection List

If you have already created one or more policy databases, they are listed in the Policy Database Selection panel, as shown in Figure 233.

Figure 233. Policy Database Selection Panel

Databases created with the NEW command automatically appear in your policy database list. You can add other databases if you know the name of their data set and have the appropriate authority to update them. You can only 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. Enter s in the Action column on that line.

The selected policy database is opened, and the Entry Type Selection panel for that database is displayed. At this point you are ready to start defining your enterprise.

The commands and actions available from the Policy Database 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 1 on page 5.

(N) NEW
This option lets you create a new policy database as described in "Creating a New Policy Database" on page 299. The Create a New Policy Database panel is displayed. See "Creating a New Policy Database" on page 299 for more details.

(E) EDIT
This option lets you change the Enterprise name and data set name of your policy database for the build, report, and log output. To change the data within the database, you must OPEN it. Using the EDIT option, you can also start and stop the logging of user activity. For more information, see "Specifying Settings for Policy Databases" on page 304.
Policy Database Maintenance

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

SA z/OS does not provide any extended security to prevent you from deleting policy database data sets. This must be provided by a System Authorization Facility (SAF) product such as RACF.

Note:
SA z/OS does not allow you to delete more than one policy database at a time. If you enter multiple delete requests in the **Action** column, the first request is processed and the remainder are ignored.

(A) ADD
This option lets you add a policy database to the displayed list. You enter a policy database name, an enterprise name and the name of a policy database data set that already exists.

(M) REMOVE
This option lets you remove the policy database details from the displayed list. The data set itself is not affected.

Note:
If you remove a policy database from all display lists, the policy database still exists as a z/OS data set.

(B) BUILD
This option displays the Build Functions Menu. You can build various configuration files. This function is explained in Chapter 7, "Building and Distributing Configuration Files," on page 317.

(G) MANAGE
This option will take you to the Data Management Menu panel shown in Figure 250 on page 337 which allows you to import policies into a policy database of the current release of SA z/OS.

You can find more information about this option in Chapter 9, "Data Management," on page 337.

(R) REPORT
This option displays the Report Selection Menu for the selected policy database. You can create a report of all automation data in the policy database. The report offers options to report either the full automation policy, or to select a specific entry type and name to be reported.

You can either create an unformatted file to be viewed in a TSO session, or an HTML file to be viewed by an Internet Browser.

The following reports can be created:

- A report for all objects in the policy database, that will not be used for building the system operations configuration files (unlinked entries and relationships).
- A report for all automatically generated resources, which are the resources of type APL, APG, and MTR.
Policy Database Maintenance

- A report of all members in the system operations build output data set, that are not referenced by the actual configuration.
- A status message report.

For more information, see Chapter 6, “Creating Reports about Policy Databases,” on page 309.

(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
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, the next higher (alphabetically) entry is the second in the list. This is the same as the ISPF LOCATE function.

Creating a New Policy Database

A policy database is a partitioned data set (PDS) that contains policy data in its members. To create a new policy database, follow these steps:

Step 1. Select option 4 (Policies) on the Customization Dialog Primary Menu.
   This takes you to the Policy Database Selection panel, shown in Figure 234 which lists all policy databases that are currently available. You can toggle the display of the data set and enterprise name in the VIEW pull-down menu.

   MENU COMMANDS ACTIONS VIEW HELP
   -------------------------------------------
   AOFPGDB Policy Data Base Selection Row 1 to 2 of 2
   Command ===>
   Action PolicyDB Name Enterprise Name/Data Set Name
   ___________ _______________ _____________________
   USER_PDB USER.ENTERPRISE ‘USER.DOCU.PDB’
   RESOURCE_PDB MY.ENTERPRISE ‘USER.RESOURCE.PDB’
   
   Figure 234. Policy Database Selection Panel

Step 2. Enter N or NEW on the command line.
This takes you to the Create a New Policy Database panel, shown in Figure 235 on page 300

   MENU COMMANDS ACTIONS VIEW HELP
   -------------------------------------------
   AOFPGDB Policy Data Base Selection Row 1 to 2 of 2
   Command ===>
   Action PolicyDB Name Enterprise Name/Data Set Name
   ___________ _______________ _____________________
   USER_PDB USER.ENTERPRISE ‘USER.DOCU.PDB’
   RESOURCE_PDB MY.ENTERPRISE ‘USER.RESOURCE.PDB’
   
   Figure 234. Policy Database Selection Panel

   MENU COMMANDS ACTIONS VIEW HELP
   -------------------------------------------
   AOFPGDB Policy Data Base Selection Row 1 to 2 of 2
   Command ===>
   Action PolicyDB Name Enterprise Name/Data Set Name
   ___________ _______________ _____________________
   USER_PDB USER.ENTERPRISE ‘USER.DOCU.PDB’
   RESOURCE_PDB MY.ENTERPRISE ‘USER.RESOURCE.PDB’
   
   Figure 234. Policy Database Selection Panel
Creating a New Policy Database

Step 3. Fill in the fields on this panel:

- The **PolicyDB Name** field is mandatory. Enter the name of the new policy database; this can be up to twenty characters in length.
- The **Enterprise Name** field is mandatory. This can be up to twenty characters in length. You can use the same value as in the **PolicyDB Name** field.
- The **Data Set Name** field is mandatory. This can be any valid z/OS data set name. Standard rules for quoted and unquoted data set names apply.

If you enter the name of an existing data set, a warning is issued. If you enter a name that has not been given to an existing data set, a new data set will be created. The policy database that you are creating will be stored in that new data set.

- The **Model PolicyDB name** field is mandatory. This must be an existing policy database name. The new policy database that you are creating will contain a *copy* of the model policy database.

You can use either of the SA z/OS sample policy databases *BASE or *EMPTY or a database that is currently in your list of policy databases.

If you do not want your new policy database to contain data that is copied from an existing policy database, use *EMPTY in this field.

If you want to see what model databases are available, type “?” in this field and press Enter. This will display the Select Model Policy Database panel that you can select a database from.

Note

If a user PDB is selected (that is, one without a “*” as the first character), no add-on PDBs can be selected.

- Selecting one or more of the sample add-on policies from the list on the panel is optional. If you do make a selection, the new policy database that you are creating will contain a copy of each add-on that you have selected. If an add-on policy is marked as customizable, this means that
it contains components that can be separately selected (for example, *ITM contains definitions for OMEGAMON II and OMEGAMON XE monitors).

**Note:**

Each of the sample add-on policies that are shipped with SA z/OS contain definitions that fully complement those in *BASE and *EMPTY.

**Step 4.** Press Enter to submit your input.

This takes you to the New Policy Database Data Set Information panel, shown in Figure 236.

![Figure 236. New Policy Database Data Set Information Panel](image)

**Step 5.** The fields that you must enter data in are listed below. For fields that are not explicitly explained, see the appropriate z/OS literature.

**Managed Storage**

This indicates whether the data set should be allocated on managed storage. If this field is set to YES, the fields marked with an * are used to allocate the data set. If NO is specified then these fields may be specified but are not used.

**Management class**

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

**Storage class**

This specifies the storage class that should be used to obtain the storage related information for the allocation of the data set. The following data, related to storage class, can be specified if you are not using managed storage or you have not specified a storage class or if you wish to override the default storage class values.
Creating a New Policy Database

Volume serial

The volume that the database is to be stored on. If this field is left blank, it defaults to your local system default.

Data class

This specifies the data class that should be used to obtain the data related information for the allocation of the data set. The following data, related to data class, can be specified if you are not using managed storage or you have not specified a data class or if you wish to override the default data class values:

Space Units

Valid allocation units for the data set are:
- CYLINDERS
- TRACKS
- BLOCKS
- MEGABYTES
- KILOBYTES

Primary quantity

This specifies the primary space allocation in space units. Specify a value according to the size of your enterprise.

Secondary quantity

This specifies the secondary space allocation in space units. Specify a value according to the size of your enterprise.

Directory blocks

Number of blocks to allocate for the directory. This field is only used when allocating a PDS. Specify a value according to the size of your enterprise.

Record format

This is FB and cannot be changed.

Record length

This is 80 and cannot be changed.

Block size

This defines the block size for the data set. It must be a multiple of 80. The default is 32720.

Data Set Name type

This indicates whether the data set is to be allocated as a partitioned data set (PDS) or partitioned data set extended (LIBRARY).
Creating a New Policy Database

Device Type
The type of storage device. This field defaults to SYSDA.

Note:
Many installations regularly delete data sets that are 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.

Step 6. Press Enter to submit your input.
This creates the new policy database. SA z/OS displays a Command Progress Display panel. When database creation is finished, the Entry Type Selection panel is displayed for the new database.

Using SA z/OS Sample Policies
SA z/OS provides sample enterprise organizations implemented as SA z/OS policies. The sample policies that are shipped with SA z/OS are installed when the SA z/OS customization dialog is installed.

To see what sample policies there are, navigate to the Create a New Policy Database panel as described in "Creating a New Policy Database" on page 299. The asterisk character (*) at the start of a policy name on that panel indicates that it is a sample.

Diagrams of the sample policies are also provided as PDF files that are located in the USS installation path. The default for this path is: /usr/lpp/ing/doc/policies/.

You can never modify the sample policies themselves. You can only modify copies of the samples. To create a copy of a sample policy, follow the steps in "Creating a New Policy Database" on page 299 and select the sample that you want to copy in step 3 on page 300.

For information on how to import a sample policy into an existing policy database, see "Importing Sample Add-On Policies" on page 341.

There are two kinds of sample policies: basic and add-on.

The basic policies are:
*EMPTY
This sample is simply an empty policy database.

*BASE
This sample policy simplifies the exploitation of basic system automation using SA z/OS. This allows users to startup and shutdown a basic z/OS computing environment within a sysplex.

The *BASE policy contains the main z/OS, networking and security components that are required in most client shops:
1. z/OS Link List Look-Aside
Creating a New Policy Database

1. z/OS Base components
2. z/OS Base applications
3. Networking components
4. Security components
5. SA z/OS components

The computing environment consists of three systems running within a sysplex. This environment is considered to be an appropriate base for high availability solutions, delivered as add-on model policies. Depending on your needs, these models can easily be tailored to less or more than three systems, just by unlinking or removing, or adding or linking systems.

A single system needs to be defined as a sysplex with only one system.

The add-on policies are:
- *CICS
- *DB2
- *E2E
- *GDPS
- *HYPERSWAP
- *IBMCOMP
- *IMS
- *ITM
- *NMC
- *PROCOPS
- *SAP
- *TBSM
- *TWS
- *USS
- *WEBSPHERE

For information on what is in the sample add-on policies, use the view (v) action on the Create a New Policy Database panel (shown in Figure 235 on page 300).

Additionally, there are diagrams available for all add-on policies. The default HFS path where they can be found is usr/lpp/ing/doc/policies. For more information about the installation settings, refer to Step 1 in Chapter 8 - Installing SA z/OS on Host Systems in the IBM Tivoli System Automation for z/OS Planning and Installation.

Specifying Settings for Policy Databases

Specifying the Edit option from the Policy Database Selection panel for a policy database leads to the Settings for Policy Database panel, as shown in Figure 237 on page 305.
The first set of fields on this panel allows you to specify **Enterprise and data set names**, as follows:

**Enterprise Name**
This allows you to re-specify the enterprise name. If this policy database has also been added to the Policy Data Base Selection list of other users they need to REMOVE or ADD the policy database to get the new name.

**PolicyDB data set name**
This allows you to re-specify the enterprise data set name.

**Build output data set**
This allows you to specify where you want the configuration files that are created by the System Operations build to be stored.

**Report output data set**
This allows you to specify where you want the output, created by the REPORT function, to be stored. It is not recommended that you store the output as a member of the policy database itself due to record length problems.

If you leave the field empty, SA z/OS allocates a data set with a default name (the policy database data set name extended with .REP as the low-level qualifier) and a default size.

If you do specify a data set name, SA z/OS allocates a data set with that name and a default size.

If you preallocate a data set and specify the name here, SA z/OS uses that data set. For the complete policy database report, it is recommended that you use format VB and a record length of 1000 to avoid truncations.

For other reports, a record length of 132 is usually sufficient (see Figure 239 on page 309 for a list of other reports that are available).

**Report output file system**
This allows you to specify the UNIX system where the report output should be stored. No upper case translation is performed. If a directory is specified that does not exist, it is created.
Specifying Settings for Policy Databases

Output file name
This specifies where you want the output created by the file update function to be stored. It needs to be a sequential file. You can leave the output file name field empty. If you do, SA z/OS will allocate a data set with a default name (policy database data set name extended with a low level qualifier of .UPD) and a default size. If you specify a sequential file name that has not been preallocated, SA z/OS will allocate a data set with that name and a default size. If you preallocate a sequential file and specify the name here, SA z/OS will use that file. It is recommended to use format VB and a record length of 255 to avoid truncations.

Input file name
This specifies where you want the input created by the file update function to be stored, or the input for the policy database update to be read from. It needs to be a sequential file.

The second set of fields allows you to set PolicyDB activity log options for this policy database. See "Logging Policy Database Modifications" for more details.

Logging Policy Database Modifications
With SA z/OS you can write a log entry whenever a modification has been made for a specific Policy Database by using the policy database activity log option on the Settings for Policy Data Base panel (see Figure 237 on page 305). Fill in the following fields:

Write log entries
Turns logging ON or OFF for this Policy Database. The default value is OFF.

Log data set name
This allows you to specify two data sets for the activity log. Whenever a data set runs out of space or cannot be accessed for some reason, the dialog switches automatically to the alternate data set and continues writing the log records there.

The alternate log data set is cleared and exit routine INGEX09 is called to allow user-specific actions, such as backing up the data set.

If you leave this field blank, SA z/OS allocates data sets with a default name (the policy database data set name extended with .LOG1 or .LOG2 as low level qualifier) and a default size.

If you specify a data set name, SA z/OS allocates a data set with this name and a default size.

If you preallocate a data set and specify the name here, SA z/OS uses this data set.

It is recommended that you use format VB and a record length of 250 to avoid truncations.

If the Settings for Policy Data Base panel is invoked and the specified policy database data set cannot be accessed, the log option fields are initially empty.
Specifying Settings for Policy Databases

Whenever the policy database data set name is changed, the current log options are read from the policy database and displayed on the panel.

The following information is logged:
- Editing a policy item (for example, changing an automation flag or a command)
- Changing a link, which includes:
  - Entry type links, for example, a link between an application and an application group
  - Links of instances and classes
  - Resource links
- Adding events to a trigger
- Invoking the DATA MANAGEMENT functions (these commands change the policy database content)
- Invoking control file build (it writes build statistics into the policy database)
- Creating and deleting objects
- Copying objects
- Errors (as long as a policy database is open)
- Renaming objects

The following information is not logged:
- Option changes (using selection 0 Settings)
- Data set name changes (EDIT)
- REPORT function invocations
- VIEW functions (for example, view build report)
- Initial conversion of the policy database (when invoked the first time with SA z/OS)
- Invocation of batch functions, for example, batch build

A log entry includes the user ID, timestamp, and the policy item that was changed, or the command name and parameters if available. An example of an activity log is shown in Figure 238 on page 308.
Figure 238. Example of an Activity Log
Chapter 6. Creating Reports about Policy Databases

If you select option 3 Report of the Customization Dialog Primary Menu shown in "How to Start the Customization Dialog" on page 16, the panel shown in Figure 239 is displayed, where you can create reports about various SA z/OS objects.

<table>
<thead>
<tr>
<th>OPTION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Report</td>
<td>Create Policy Database Report</td>
</tr>
<tr>
<td>2 Unlinked</td>
<td>Create List of entries not linked in Policy Database</td>
</tr>
<tr>
<td>3 ViewUnlinked</td>
<td>View List of unlinked entries</td>
</tr>
<tr>
<td>4 Resources</td>
<td>Create List of generated Resources</td>
</tr>
<tr>
<td>5 ViewResources</td>
<td>View List of generated Resources</td>
</tr>
<tr>
<td>6 MemberList</td>
<td>Create List of unused members in configuration data set</td>
</tr>
<tr>
<td>7 ViewMemberList</td>
<td>View List of unused members</td>
</tr>
<tr>
<td>8 StatusMsgs</td>
<td>Create Status Messages Report</td>
</tr>
<tr>
<td>9 ViewStatusMsgs</td>
<td>View Status Messages Report</td>
</tr>
<tr>
<td>A Symbols</td>
<td>Create List of Symbols (AOCCLONE) in Policy Database</td>
</tr>
<tr>
<td>B ViewSymbols</td>
<td>View List of Symbols</td>
</tr>
</tbody>
</table>

Figure 239. SA z/OS Report Selection Menu

**1 Report**

Use this option to create a report that includes all the automation data of the selected policy database. SA z/OS displays the Policy Data Base Report Parameters panel, as shown in Figure 240 on page 310, where you can choose to create either the full automation policy report, or to create a report about a specific entry type and name. You can also run this function in batch mode.
Creating Reports about Policy Databases

Use the **Report options** section of the panel to specify how the report is created and saved.

**Data set type**

Specify the output type for the report:

- **PDS** The report is stored as a partitioned data set.
- **FS** The report is written to a UNIX file system (HFS or ZFS).

The default is PDS.

**PDS name**

If the data set type is PDS, this field can remain empty. A default data set name is assumed and a default allocation is performed. Any user input is translated to upper case.

**File system directory**

If the data set type is FS, you need to enter the file system directory. There is no default. No uppercase translation is performed. If you specify a directory that does not exist, it is created.

**Member / file name**

Specify the member or file name for the report. The default is $RPTHTML or $RPTFLAT. Any name that you specify is saved across sessions.

For a PDS it must be a valid TSO member name. When it is downloaded to a PC, you must add the file extension .html. Note also that the file name for FS is *case-sensitive*.

If you choose to split an HTML file that is targeted to a PDS, the last 1 or 2 characters of the file name are replaced with a serial number. This is 1–n if you specify a single figure split value, or 01–nn for a 2-digit value. Thus, $RPTHTML becomes either $RPTHTM1–$RPTHTMn or $RPTHT01–$RPTHTnn.

If split files are directed to an FS director, the running number is appended to the specified member name, and a .html suffix is added as the files are written.
Creating Reports about Policy Databases

If a member with the specified or generated split-file name already exists, you are asked to confirm the report generation before it is overwritten.

Mode
Specify whether the report should be created in the foreground (ONLINE) or background (BATCH). The default is ONLINE.

Format
Specify the format of the report:
- FLAT creates a plain text file to be viewed in a TSO session.
- HTML creates an HTML file that you can view with your Internet browser. This version of the report creates hyperlinks for all entry names and resource names that take you to the start of the data for the selected entry or resource. These links also work if the HTML report has been split into several files. Note however that a hyperlink is not created if the application or application group name is not unique (for example, there are identical subsystem names for different applications) or it does not exist in the policy database.

The default is FLAT.

HTML file split
If you have large policy databases that produce large reports you may find that your Internet browser has difficulties handling a single large file. You can use this field to have the HTML report split into several smaller files, as follows:

1 One file is generated. This is the default.

nn (>1)
   The report is split into nn files that are all roughly the same size.
   The files are split between policy entries.

* The report output is split with a default size of 10,000 lines per file.

The Command Progress Display panel informs you about the progress of report generation.

You can run a job for batch report creation. The SINGSAMP sample library contains a sample job in member INGEBRPT. The output data set or file system must exist before you run this job.

The example in Figure 241 on page 312 shows a small part of the data listed in the report output format FLAT for the application JES2.
Creating Reports about Policy Databases

```
JES2
Short description : JES2 policy definition
Long description : Defines automation and monitoring characteristics for the JES2 job.
Application Type : JES2
Subsystem Name : JES2
MVS job name : JES2
Scheduling subsystem : MSTR
Command Prefix : $
Restart after IPL : START
Subsystem Startup Parameters : ,PARM=WARM,REL=710,SYS=&SYSNAME.

========== JES2 Drain Specifications ==========
Drain lines : YES
Drain VTAM interface : YES
Drain printers : YES
Force drain lines : $E
Force drain VTAM interface : $E
Force drain printers : $I

========== Application Messages and User Data ==========
<table>
<thead>
<tr>
<th>Message id</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$HASP095</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Code 1</th>
<th>Code 2</th>
<th>Code 3</th>
<th>Value Returned</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>$PJ* STOPPING</td>
</tr>
<tr>
<td>ERROR*</td>
<td>$K03</td>
<td></td>
<td>STOPPING</td>
</tr>
<tr>
<td>ABEND*</td>
<td>SA22</td>
<td></td>
<td>STOPPING</td>
</tr>
<tr>
<td>*</td>
<td>*</td>
<td></td>
<td>ABENDING</td>
</tr>
</tbody>
</table>

| $HASP098 |

<table>
<thead>
<tr>
<th>Pass/Selection</th>
<th>Reply Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DUMP</td>
</tr>
<tr>
<td>2</td>
<td>END</td>
</tr>
</tbody>
</table>

POSTSTART | Executed after startup has been initiated |

<table>
<thead>
<tr>
<th>Pass</th>
<th>Automated Function</th>
<th>Command Text</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>JESOPER</td>
<td>MVS F BPXINIT,RESTART=FORKS</td>
</tr>
</tbody>
</table>

========== Relationships ========== | Relationship Type | Supporting Resource | Automation | Chaining | Satisfy condition |
| PREPUNAVAILABLE | VTAM/APL/= | PASSIVE | WEAK | WhenDown |

Figure 241. Example of Report Output Format FLAT
```
Creating Reports about Policy Databases

2 Unlinked
Use this option to 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
- Relationships that are not used during a build

3 ViewUnlinked
Use this option to view the report of all unreferenced objects created by option 2 Unlinked.

4 Resources
Use this option to 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
Use this option to view a list that is generated with option 4 Resources for each existing system.

6 MemberList
Use this option to create a report of all members in the system operations build output data set that are not referenced by the actual configuration anchored in the ACF map member AOFACFMP. This helps to identify members that can be deleted when you perform a data set cleanup. The report is divided in two sections:
1. The first section lists all unreferenced members that have been identified as valid configuration files, together with some additional information such as entry name, or build time.
Creating Reports about Policy Databases

2. The second section lists all other unreferenced members, where either no build statistics could be found, or which have not been identified as valid configuration files.

Report members created by the customization dialog itself such as build report member $BLDRPT are not included in the report. To run this report, a successful build is required. Otherwise the control file fragments may be damaged and incomplete, and the content of the report is unpredictable.

7 ViewMemberList

Use this option to read the report of all unreferenced members that was created by option 6 MemberList.

8 StatusMsgs

Select this option to create a Status Messages Report. This includes status messages that are predefined by SA z/OS as well as user-defined status messages. The report shows the message ID, message description, and resource status indicated by this message and, if it is defined in the policy database, the entry name and the entry type that the message has been defined for. An example of the start of the Status Messages Report is shown in Figure 244 on page 315.

Pressing Enter returns you to the Report Selection panel, where you can view the report by selecting option 9 ViewStatusMsgs.
Creating Reports about Policy Databases

9 ViewStatusMsgs
Select this option to view the Status Message Report that was created by using option 8. An example of the start of the Status Messages Report is shown in Figure 244.

In the report, user-defined messages are listed with their associated entry name and entry type, whereas those messages listed without these columns are predefined SA z/OS messages.

<table>
<thead>
<tr>
<th>Message id</th>
<th>Status</th>
<th>Description</th>
<th>Entry Name</th>
<th>Typ</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANR0990I</td>
<td>ACTIVE</td>
<td>TSM is coming up</td>
<td>+SA_PREDEFINED_MSGS</td>
<td>MVC</td>
</tr>
<tr>
<td>ANR0991I</td>
<td>TERMINATED</td>
<td>TSM has ended</td>
<td>+SA_PREDEFINED_MSGS</td>
<td>MVC</td>
</tr>
<tr>
<td>ANR0993I</td>
<td>UP</td>
<td>TSM is available</td>
<td>+SA_PREDEFINED_MSGS</td>
<td>MVC</td>
</tr>
<tr>
<td>ARC0002I</td>
<td>TERMINATED</td>
<td>DFHSM has terminated</td>
<td>+SA_PREDEFINED_MSGS</td>
<td>MVC</td>
</tr>
<tr>
<td>ARC0008I</td>
<td>UP</td>
<td>DFHSM is available</td>
<td>+SA_PREDEFINED_MSGS</td>
<td>MVC</td>
</tr>
<tr>
<td>ARC0011I</td>
<td>TERMINATED</td>
<td>DFHSM is stopping</td>
<td>+SA_PREDEFINED_MSGS</td>
<td>MVC</td>
</tr>
<tr>
<td>ARC0015E</td>
<td>HALTED</td>
<td>DFHSM is degraded</td>
<td>+SA_PREDEFINED_MSGS</td>
<td>MVC</td>
</tr>
<tr>
<td>ASA2011I</td>
<td>UP</td>
<td>RRS is available</td>
<td>+SA_PREDEFINED_MSGS</td>
<td>MVC</td>
</tr>
<tr>
<td>ASA2013I</td>
<td>ABENDING</td>
<td>RRS is abending</td>
<td>+SA_PREDEFINED_MSGS</td>
<td>MVC</td>
</tr>
<tr>
<td>CQS0008W</td>
<td>CAPTURE</td>
<td>CRITICAL message captured</td>
<td>IMSCQS</td>
<td>APL</td>
</tr>
<tr>
<td>CQS0009W</td>
<td>CAPTURE</td>
<td>CRITICAL message captured</td>
<td>IMSCQS</td>
<td>APL</td>
</tr>
<tr>
<td>CQS0008O</td>
<td>UP</td>
<td>CQS up message</td>
<td>+SA_PREDEFINED_MSGS</td>
<td>MVC</td>
</tr>
<tr>
<td>TEST001I</td>
<td>UP</td>
<td>Test application has started</td>
<td>APPL1</td>
<td>APL</td>
</tr>
<tr>
<td>TEST002I</td>
<td>TERMINATED</td>
<td>Test application has ended</td>
<td>APPL1</td>
<td>APL</td>
</tr>
</tbody>
</table>

Figure 244. Example Status Messages Report (Excerpt)

A Symbols
This option allows you to create a report with information about all the system automation symbols (AOCCLOENx) in the policy database.

The report file is written to the report output data set (default name is the policy database data set name with .REP as the low level qualifier). The member name is RTPSYM.

The report includes:
- The Symbolname and Description, as defined in the Enterprise (ENT) SYMBOL DEFINITION policy
- For each Symbolname it shows the System name, Value and Specific Description as defined in the System (SYS) AUTOMATION SYMBOLS policy for each System entry where either the Value or Specific Description have been defined for that Symbolname

B ViewSymbols
Select this option to view the Symbols Report that was created by using option A. An example of the start of the Symbols Report is shown in Figure 245 on page 316.
Creating Reports about Policy Databases

<table>
<thead>
<tr>
<th>Symbolname</th>
<th>Value</th>
<th>General Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOCCLONE</td>
<td>2</td>
<td>for use by AddOn PDB Entries</td>
</tr>
<tr>
<td>AOCCLONE1</td>
<td>16</td>
<td>VTAM subarea</td>
</tr>
<tr>
<td>AOC1</td>
<td>17</td>
<td>VTAM subarea</td>
</tr>
</tbody>
</table>

Figure 245. Example Symbols Report (Excerpt)
Chapter 7. Building and Distributing Configuration Files

After you have completed the SA z/OS customization process, you can build the configuration files that are needed for automation.

**Note:**
You must use the customization dialog to create the configuration files. Do not edit any of the configuration files manually.

### How to Define Data Sets for Build Processing

To use the batch options of the build facility for automation control files, you must first specify the names of the data sets required for the batch processing. These names form the DD name concatenations to be included in the JCL of the batch processing job.

The data set names are saved in an ISPF table in the partitioned data set that is referenced by the AOFTABL DD name, for example, ING.CUSTOM.AOFTABL (allocated during installation of SA z/OS).

On the Customization Dialog Primary Menu panel, select the Settings option. This displays the Customization Dialog Settings Menu (see Figure 257 on page 361). On this panel, choose option 11, DSNAMES. The Data Set Customization Details panel is displayed (see Figure 261 on page 367).

For more information, see "Defining Data Sets for Batch Processing" on page 367.

### Building the Configuration Files

Building the configuration files causes enterprise automation data for SA z/OS to be built. This data consists of:

- The automation control file (ACF), which is built into a set of file fragments and a master include file
- The automation manager configuration file (AMC file)
- The NetView automation table (AT)
- The message revision table (MRT)
- The MPFLSTSA member
- Optionally, a processor operations control file (only for systems that are running earlier releases of SA z/OS)

The enterprise-wide policy database for your sysplex and the standalone system is created and maintained with the customization dialog on any suitable system. This is called the *configuration focal point*.

The configuration build is performed with the customization dialog and the configuration files are distributed from there to the locations where needed.
Building the Configuration Files

To build a new enterprise automation configuration you invoke the BUILD process, as follows:

- From the Policy Database Selection panel, enter the action code B in the Action column of the policy database that you want to invoke the BUILD process for.
- From the Entry Type Selection panel, you can invoke the build function as follows:
  - Choose the BUILD (B) command from the Command menu.
  - If you want to process a certain entry type, enter the BUILDF (F) command at the command line and specify the identifier for the entry type that you want to build. For example, to build all system objects that you have defined, enter
    BUILD SYS or F SYS. You go to the Build Parameters panel (see Figure 246 on page 319) with the entry fields filled in according to the parameters that you entered with the BUILDF command.

Note that the build function builds all systems irrespective of whether they have been modified or not.

- From the Entry Name Selection panel, you can invoke the build function as follows:
  - 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.
  - Enter the BUILDF (F) command at the command line for the object you want to build, for example, F JES2.

You then go to the Build Parameters panel (see Figure 246 on page 319) with the entry fields filled in according to parameters that you entered with the BUILDF command.

A Command Progress Display panel shows the progress of the build. If errors are encountered during the build process, the Command Progress Display displays a message. The automation configuration files are placed in the data sets that you defined in the Output Data Set field on the Build Parameters panel.

Every new build generates an automation configuration map AOFACFMP that contains:

- For each system, one ACF<nnn> system include member that is used by the automation agents
- For each sysplex, one HSA<nnn> sysplex include member that describes the automation manager configuration

The ACF, AMC, AT, and MPF build is always invoked together. This is because the configuration data always needs to be synchronized so that the automation agents and the automation manager operate on the same data.

Each automation control file fragment contains information about a specific resource that is defined through the customization dialogs. Each z/OS system in the enterprise has its own master include file. The master include file references the automation members that are appropriate for that system. The name of the master include file for each system is created by SA z/OS. It starts with prefix ACF followed by four alphanumeric characters.

**Configuration Files Build Options**

The Build Parameters panel allows you to specify the information required to build the configuration files that are required for automation. You can build these
Building the Configuration Files

configuration files for standalone systems, individual sysplexes or your complete enterprise, using the options shown in Figure 246.

You can enter one of the following build processes:

1 **Build a complete enterprise**
   Builds all automation configuration files for a complete enterprise.

   **Note:**
   This option must be successfully completed at least once before you can use options 2, 3, or 9.

2 **Build sysplex group or stand alone system**
   Allows the selection of a single sysplex group or a standalone system.

   This build option performs a complete build of all objects linked to the selected sysplex or standalone system. For a large policy database, this option helps to reduce build time for small changes, while creating a complete set of configuration files that are ready for automation.

3 **Build entry type or entry name**
   Allows you to specify an entry type and name to be rebuilt. Only the ACF fragments for that entry will be updated.

   If for example, a timer object is modified and linked to another system, a build of this object with option 3 would only build that fragment and update the timestamp in the system include member, but not build the link changes for the system.

   The same is true if an object is renamed. A build with option 3 of the new name would not add this new object to the system include member.

   Instead a build with option 2 is required.

   For entry type System, this has the consequence that a selective build is rejected if there is also a link change pending.
Building the Configuration Files

However, if this option is used with entries of type Events, Triggers, Service Periods, Application, ApplicationGroup, System, or Group, a check is made as to whether data that affect the automation manager has been changed.

For certain entry types (for example, Service Periods, Events, Triggers) this option cannot be used because they do not have data included that needs to be built into an ACF fragment. In this case, building only the ACF fragment for the requested entry would mean that your previously generated AMC files would no longer be compatible with your current ACF files. To avoid this, a complete build (using build type MOD) will automatically be invoked. Before the build actually starts a confirmation panel is displayed explaining this situation, and allowing you to defer the build.

You will see the same behavior with automation tables if this option is used with entries of type APL or MVC. A check is made as to whether data has been changed that affect the NetView automation table. If so, a confirmation panel is displayed allowing you to defer or to start a complete build (using build option MOD).

If you enter a '?' in this field, SA z/OS displays a list of either entry types or a list of available entry names for the selected entry type, or both.

4 View build report
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.

9 Build Processor Operations
Allows you to use SA z/OS 3.4 configuration file to build a processor operations control file for a focal point system that is running a release of SA z/OS earlier than V3.3.

The data that is created during an automation configuration build in the Output Data Set is read, processor operations definitions are created and the generated output is placed in the ProcOps Control File. You must allocate the data set as a sequential file before the build can be started. You can use sample job INGEDLGA to allocate it (see IBM Tivoli System Automation for z/OS Planning and Installation).

Notes:

1. If a policy database is new, or the build output data set is empty, a full build is required to produce a complete set of configuration files. This means you need to issue a build with option 1 and Type = ALL. For all subsequent builds, option 2 or 3 and Type = MODIFIED can be used.

2. While build option 2 only updates a single system or sysplex, deleted standalone systems and sysplexes still remain in the configuration files. For example, a deleted sysplex or standalone system is not removed from the ACF map AOFCFMP. So, you may consider issuing a build with option 1 whenever a large number of parts has been deleted, to clean up unnecessary entries from the configuration files.

3. Fragments are not built for all entry types (for example, not for SVP or TRG).

In the Build options section, you can enter values for the following fields:
Building the Configuration Files

Output Data Set
Use this field to specify where you want to store the build output. The data set must be an existing partitioned data set.

If the data set is not empty, but already contains a control file, the policy database name and the configuration name within the data set are compared with those of the requested build. If they do not match, a message is issued and the build is not started.

Mode
Use this field to specify how to process the BUILD request:
ONLINE Indicates that the BUILD should run in the foreground.
BATCH Indicates that the BUILD should run in the background.

The default value is ONLINE (if this field is set to blank).

Type
Use this field to specify which entries are to be processed.
MODIFIED Indicates that only those entries that have been changed since the last BUILD request should be processed.

With this option, the build distinguishes between:
• Data changes in the ACF
• Link changes in the ACF
• Data changes in the AMC file
• Link changes in the AMC file

Depending on the type of change, the affected objects are identified, and only the necessary parts of the configuration file are rebuilt.

For example:
• If a new timer is linked to one system, only the system include member for that system is rebuilt. For all other system include members only the header timestamps are updated, and the build of the AMC file, AT, MRT, and MPFLSTSA member is skipped, because timer data does not go to the automation manager configuration file, AT, MRT, or MPFLSTSA member.
• If a new application is linked to a Sysplex Application Group, the build determines which systems are linked to that sysplex, and the include members for those systems are rebuilt automatically, together with the AMC, AT, MRT, and MPFLSTSA member for that sysplex.
• If the AT/MRT Scope is changed, only the AT, MRT, and MPFLSTSA member are built.

ALL Indicates that all entries are to be processed unconditionally, even if there has been no change to those entries since the last BUILD request.

If a build type of ALL is selected together with build option 2, only the configuration files that belong to the selected sysplex or standalone system are rebuilt.

The default value is MODIFIED (if this field is set to blank).

The very first build after installing a new release must be a build of type ALL in order to get a complete set of configuration files for the automation manager and automation agent structure.

Configuration
This field allows you to select between alternative configurations:
Building the Configuration Files

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 built into different data sets. The default value is NORMAL (if this field is set to blanks).

In the Job statement information section you must enter a valid JOB statement for the batch job, if you specified that the BUILD request should run in BATCH mode.

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 MVS job names linked to the same system. Build errors that cause termination 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.

If an error is shown, no usable set of configuration files was created. It is necessary to correct the cause of the problem and run the build again.

If a warning is shown, the configuration files can be used for automation. However it is recommended that you fix the problem and run the build again.

A possible cause for a warning during ACF fragment build may be that there are policy database objects with missing or conflicting definitions, that are not critical for automation, for example, an application default entry (type ADF) without any definitions. In this case, the build function continues, but issues a warning message. This message is displayed whenever a full enterprise build is issued, and does not require that the application be linked to a system.

There is a summary message at the end of the AT build report, and specific messages that show, for example, the object name somewhere within the build progress messages. The specific message reads:

```
.......WARNING. System MIGR_TEST includes the following entries with an unsatisfactory build status in main member ACFZ995:
.......-- SYSTEM_DEFAULTS (NUL)
```

The summary message at the bottom of the build message log reads:

```
.......WARNING. Missing or conflicting definitions found for entry types: SDF. The entry names are listed in the build report.
```

Possible causes for a warning during system main include member build are:
Building the Configuration Files

- Conflicting type definitions for an application instance that is linked to an application class.
- Applications with duplicate MVS job names 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, 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 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 these application groups.
- One or more supporting application resources are not linked to the system. The relationships are not added to the automation manager configuration file.

You can tailor the level of detail of the warning messages. When you invoke the **Settings** option, selecting **3 MESSAGE** allows the message level to be set, as shown in Figure 247.

**Figure 247. Message and Logging Functions Panel**

With level 1 (the default), in the above situation just one summary message is displayed per manager configuration, whilst level 2 displays all details of the definitions causing the problem.
Building the Configuration Files

You can also specify whether SA z/OS should check for Processor Operations-related messages during configuration build and write them to the build report $BLDRPT. The default value is OFF. If you use processor operations you should set this option to ON.

Independent of the message logging switch, error messages are always written to the ISPF LOG. You cannot turn this off.

If the AOFPROF user profile for batch build is not allocated, the default message level is 2 to write all possible messages to the batch report and the ISPF LOG.

Distributing the Configuration Files

The SA z/OS configuration files consists of the following components:

- The automation manager's automation manager configuration file (AMC)
- The automation agent's automation control files (ACF)
- The NetView automation table (AT)
- The NetView message revision table (MRT)
- The MPFLSTSA member

SA z/OS runs on a sysplex basis, that is, the same configuration files must be available to all of the automation agents and automation managers in the sysplex.

It is recommended that you perform the build into a fixed partitioned data set, which allows a build with a type of MOD. Do not use this data set for the automation agents and automation managers. Instead you should copy it into a configuration file GDG. This helps to avoid token mismatches due to changes in the configuration data set, for example, a new build into the data set that is used by the automation agent and automation manager. Additionally, there are always backup versions of the configuration file available so that the configuration can fall back to a previous generation if there is a problem.

An automation agent needs to run on every system that is automated by SA z/OS and at least one automation manager needs to run per sysplex and one per standalone system. Automation managers can manage systems sysplex-wide. Figure 248 on page 325 shows a sample automation configuration and presents the build process.

The MPFLSTSA member must be copied to a z/OS parmlib data set.
Figure 248 shows a sample enterprise consisting of one sysplex Sysplex1 and one standalone system SYS5. The sysplex consists of four systems SYS1, SYS2, SYS3, SYS4 and has a primary and a secondary automation manager (PAM and SAM). The sysplex shares a copy of the system operations configuration files and the standalone system has its own copy of the same files.

The sysplex in this example has one primary and optionally one secondary automation manager and one automation agent on each system. The standalone system must have one automation manager (AM) and one automation agent.

You could share one common automation configuration file in your enterprise, or you can copy the automation configuration file to where it is needed as shown in our scenario. Both automation managers feed the enterprise-wide RODM data base dynamically with information that is available from both AMCs and from information that becomes available during automation processing (like status changes). This feeding is done automatically by the SA z/OS topology manager after you have started it. The generated data is used by the NMC workstation to display predefined graphical views.

The automation manager's AMC file and the automation agent's ACF files must match. Therefore the automation manager and an automation agent exchange a token to verify that the automation configuration matches. You can share one common data set containing all automation manager configuration files and all automation control files within a sysplex.
Building the Configuration Files

Note:
It is mandatory to create and maintain policy databases with the customization dialog.

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

However, it is highly recommended that you use a shared data set in a sysplex, and it is preferable that you use a shared data set for the whole enterprise.

Generating a Job for Batch Build

You can run a job for batch build that performs the same processing as the BUILD function of the customization dialog. The SINGSAMP sample library contains a sample job for this task in the member INGEBBLD. Before submitting this job, you need to edit the jobcard appropriately as described in the sample itself.

The build can only run if the output data set exists and is allocated. If the data sets are not SMS-managed, allocating the data set within the same job step does not work because the cataloging is done by z/OS at end of the job (or job step). Therefore the allocation and cataloging of the build output data set must be done in a separate job (or job step) prior to the build job (or job step).

The batch build produces the following return codes:

0  Successful.
4  Build complete, however, some warning messages are issued. Also issued if user exit INGEX02 ended with a RC > 0.
8  User exit INGEX01 ended with RC > 0. Build processing not started. No configuration files built.
16 Build not started. The policy database is from a previous version of SA z/OS. The policy database must be converted first before a batch build will be successful.
20  Severe error: build terminated without usable configuration.

SA z/OS provides user exits at invocation and termination of the build function. For more details, see “User Exits for BUILD Processing” in IBM Tivoli System Automation for z/OS Customizing and Programming.
Chapter 8. The NetView Automation Table and Message Revision Table

This chapter introduces the NetView automation table (AT) and the message revision table (MRT), which SA z/OS uses when processing messages, and how you can use the customization dialog to maintain your AT and MRT entries.

The NetView Automation Table

The NetView automation table (AT) enables you to respond automatically to messages. It contains statements that define actions that NetView takes when it receives specific messages. For example, you can issue a response in the form of a command, command list, or command processor. You can also set attributes and processing options. For example, you can suppress, log, or route messages.

You can specify any number of actions for SA z/OS to take when an incoming message matches your conditions. Actions can be commands, command lists, and command processors. For simple responses, a single command might be sufficient, such as a NetView command, a VTAM command, or a system or subsystem command. For more complex responses, you can write command lists or command processors. The AT specifies the task that the action is performed under, enabling you to run automation procedures under an autotask.

Actions also include setting message-processing options. For any particular message, you can use message-processing options to specify such things as:
• Suppressing the message (and which operator it should be displayed for)
• Triggering the execution of a command
• Setting the status of a resource
• Setting the health state of a monitor resource
• Running user-specific processing, such as performing alerting by creating a trouble ticket
• Taking a recovery action in case of a failure

For an introduction to NetView and NetView automation tables, see IBM Tivoli NetView for z/OS Automation Guide and IBM Tivoli NetView for z/OS User’s Guide.

The NetView Message Revision Table

The message revision table (MRT) enables user-defined modification of attributes such as color, route code, descriptor code, display and syslog settings, and text of original z/OS messages (rather than copies). You can make decisions about the message based on its message ID, job name, and many other properties. For example, you can take the following actions:
• Revise messages before they are presented to the system log, console or automation
• Treat a message differently depending on its source
• Suppress messages entirely
• Automate only
The message revision table can override actions taken by the z/OS message processing facility (MPF) and can generally replace the MPF. It can also remain active even while NetView is not, but the subsystem interface (SSI) address space is required. However, loading or querying the MRT, or gathering statistics, depends on the functional NetView address space being active.

Note that you can have only one MRT for your enterprise.

For more details, see *IBM Tivoli NetView for z/OS Automation Guide*.

### Enabling AT Updates

You use the AOFSMARTMAT advanced automation option (AAO) in your CNMSTGEN stylesheet to configure how the SA z/OS automation agent performs refreshes of your ATs.

If AOFSMARTMAT is set to 0, the SA z/OS automation agent is disabled from refreshing ATs and the MRT and any changes that you make to your AT and MRT entries with the customization dialog are not loaded by the automation agent. You therefore need to set AOFSMARTMAT to 2 or 3 to enable the changes that you make to your AT or MRT entries to be propagated to your SA z/OS systems. Note that the default setting for AOFSMARTMAT is 3.

For more details, see the description of AOFSMARTMAT in the table “Global Variables to Enable Advanced Automation (CGLOBALS)” in *IBM Tivoli System Automation for z/OS Customizing and Programming*.

### Setting AT/MRT Scope

At the foot of Figure 237 on page 305 is the field **NetView Automation option AT/MRT Scope** which allows you to define the scope of the NetView Automation Table and Message Revision Table that are created by the configuration file build. Valid definitions are:

**NONE**
- No NetView automation table or message revision table is built.

**ENTERPRISE**
- One NetView automation table and one message revision table are built for the entire policy database.

**SYSPLEX**
- One NetView automation table and one message revision table are built per Sysplex Group.
- One NetView automation table and one message revision table are built for each system that is not linked to a sysplex group.

**SYSTEM**
- One NetView automation table and one message revision table are built per system (default).

If the AT/MRT scope is changed to any value except NONE, the tables are rebuilt at the next configuration file build.
Within an override, you can use the keywords *BEGINAPL and *ENDAPL to mark a section for selected applications only. Usually this would be a class, but the specification of instances is also supported. Each marked sequence is contained only in the AT where an instance is linked. Likewise you use the keywords *BEGINMVC and *ENDMVC to mark a section for selected MVC components only. Furthermore you can use the keywords *BEGINSYS and *ENDSYS to mark a section for selected systems only. Each marked system is contained only in the AT of the corresponding system.

The keywords are accepted in upper case and mixed case. The Begin keywords allow the specification of one or more entry names, separated by blanks or commas. A Begin keyword without a entry name is ignored and the current selection (if any) remains in effect. If you need to specify more entry names than fit onto one line, use a continuation line by adding another line with the Begin keyword. The name specifications in multiple consecutive lines with Begin keywords are evaluated together.

There is no syntax check for the specified names and it is not verified whether the names exist as application or system entries in the policy database. If a Begin keyword is specified with a name that does not exist in the policy database, then that portion of the override is excluded from all ATs being built.

The End keywords indicate the end of the current selection and have the following properties:
- The End keyword cannot be qualified with specific entry names. Any text that follows the End keyword on the same line is ignored.
- An End keyword that does not have a matching Begin keyword is ignored.
- If there is no matching end for a Begin keyword, the current selection remains in effect until either the override ends or another specification of the same Begin keyword is found.

Whenever a *BEGINAPL is found, a full system evaluation is run replacing any selections from a previous *BEGINAPL. The same is true for *BEGINMVC and *BEGINSYS. However, a *BEGINSYS specification remains in effect if a *BEGINAPL or *BEGINMVC specification is found, and vice versa.

During initial conversion from a release prior to SA z/OS 3.3 the *BEGINAPL and *ENDDAPL keywords or *BEGINMVC and *ENDMVC are generated automatically if different overrides are found for different applications or MVS components. Here is an example of an override section with the keywords added:

*BEGINAPL C_TWS_CTRL
IF (GROUP:IGNOPC) MSGID = 'EQQW065I' & TOKEN(3) = 'WRITER' THEN
EXEC(CMD('ACTIVMSG UP=YES') ROUTE (ONE %AOFOPGSSOPER%));
*ENDAPL

If the variable "&JOBNAME." is used in an override it is resolved for all jobnames across all classes for which the message is specified. This can be limited to a subset of classes or instances with the keyword *BEGINAPL. If no linked instances can be found then the variable "&JOBNAME." is resolved with a placeholder/period(.).

It is recommended to use these keywords as little as possible because the evaluation may increase the build time depending on the specified names (for
example, with a lot of instances) and to avoid convoluted overrides. Especially it is recommended to avoid combinations of *BEGINSYS with *BEGINAPL or *BEGINMVC within one override.

If an application, MVS component, or system is deleted, a scan for the corresponding *BEGINAPL, *BEGINMVC or *BEGINSYS keyword is run across all message overrides. If the deleted entry name is found within a list of names then it is removed from the list. If there is a Begin keyword where only the deleted name is specified then the complete section of the override is deleted.

If an application, MVS component or system is renamed, a scan for the corresponding *BEGINAPL, *BEGINMVC or *BEGINSYS keyword is run across all message overrides. Wherever the name is found it is replaced with the new name.

If data are imported into a target policy database, and there is a message with an override among the imported data while the same message exists already in the target policy database with an override, then *BEGINAPL or *BEGINMVC keywords are inserted with the corresponding entry names.

If application or MVS component data are copied, and a message with an override is contained in the copied data, then a check is done to determine whether the "copy-from" application is specified in a *BEGINxxx keyword within the copied override. If this is the case then the "copy-to" application name is added to that *BEGINxxx keyword.

---

### The INGMSGSA Automation Table

SA z/OS provides an automation table, called INGMSGSA, that contains all the statements that are required for SA z/OS to work properly.

You must not modify the INGMSGSA automation table. If you need your own automation for messages that are processed in INGMSGSA, you can achieve that by using a separate AT that runs parallel with INGMSGSA. SA z/OS does not stop any message in INGMSGSA and prevent its processing in other ATs.

INGMSGSA is loaded unconditionally as the first AT ahead of any other ATs that you declare in the customization dialog. It must not be specified in the customization dialog.

For more details about SA z/OS and ATs, see “A Guide to SA z/OS Automation Tables” in [IBM Tivoli System Automation for z/OS Customizing and Programming](https://www.ibm.com/software/sysmgmt/products/support/zosAutomationGuide/).  

---

### Predefined SA z/OS Messages

Messages that are predefined by SA z/OS are provided in:

- A NetView automation table called INGMSGSA
- An MVS component (MVC) entry called +SA_PREDEFINED_MSGS

---

### The +SA_PREDEFINED_MSGS MVS Component Entry

The +SA_PREDEFINED_MSGS MVS component entry contains all the messages that are predefined by SA z/OS except the essential messages that are already provided in the INGMSGSA automation table. It is copied into a new policy database when it is created or copied into an existing policy database during conversion from a downlevel release.
The +SA_PREDEFINED_MSGS MVS component entry allows you view all that SA z/OS would automatically generate AT entries for. It thus also allows you to check a particular AT entry that is predefined by SA z/OS to verify that it is appropriate for your environment.

You cannot modify, delete or rename the +SA_PREDEFINED_MSGS MVS component entry. It is updated via an automatic APAR application process (see “How to Apply Service Updates” on page 332). You can, however, view the contents of the entry by selecting it from the list of MVS components.

Policy Items
The +SA_PREDEFINED_MSGS MVS component entry has the following policy items:

DESCRIPTION
The non-editable description that explains that the entry contains predefined messages for SA z/OS.

MESSAGES/USER DATA
This displays all the predefined SA z/OS messages. You cannot edit the entries in this policy item.

WHERE USED
This allows you to link the entry to systems, as for other MVC entries.

Modifying Predefined Messages
There are different ways that you can modify predefined messages:

- Unlinking systems from the +SA_PREDEFINED_MSGS MVS component entry with its WHERE USED policy item. The predefined messages are then not used for the AT of that system.

- Deselecting components in a user-defined MVC entry through the PREDEFINED MESSAGES policy item. The MVC entry must be linked to those systems where the deselection should take effect. Note that if multiple MVC entries are linked to a system, all components that are selected in at least one of the MVC members are added to the AT build. That is, only those components that are deselected in all linked MVC members are excluded from the AT build.

- Overwriting individual messages through the MESSAGES/USER DATA policy item of an MVC entry. How you do this depends on the display level that is set in the PREDEFINED MESSAGES policy item:
  - If it is set to 1, all predefined messages that have not been deselected are shown in the MESSAGES/USER DATA policy item. They have the same color as inherited messages for applications.
    - If you now add a command or just modify the description of a message, it becomes a message specification for the current MVC entry. You can now modify the AT or MRT specification, or both, according to your own needs.
    - This new specification superimposes the predefined SA z/OS data.
  - If it is set to 0, you can enter a predefined message ID in an empty row in the list. The existence of predefined data for an AT or MRT specification is indicated with a P in the corresponding column. You can now modify this predefined data.

- Overwriting individual messages through the MESSAGES/USER DATA policy item of an APL entry. As with predefined messages of an MVC entry, if the display level in the PREDEFINED MESSAGES policy item is set to 0, you can simply add the message ID and modify the predefined AT or MRT data according to your needs.
AT/MRT specifications are processed by NetView per system. And the system specific AT contains the entries for the messages from all instances linked to that system. So if a message is issued by multiple applications, an AT specification is not required once per application, but only once per AT. That is why an AT specification in SA z/OS applies to the message wherever it is defined across all applications.

If you specify an override for a message with a predefined AT entry then the user override completely replaces the predefined specification. This is indicated in the AT preview section of the corresponding panels where only the AT entry based on the user specifications is shown. The AT is generated as is shown in the preview section, only the user override is used within the AT.

If you want to run with a predefined AT entry on most of your systems, but need a specific override for one application only, then it is recommended you specify the message ID for that single application together with a unique suffix and add that specific override there. This results in different message IDs and so the specific user override will not replace the predefined AT entry.

Note that AT and MRT specifications are independent of each other, so, for example, you can define changes for the MRT entry without affecting a predefined AT entry for the message.

How to Apply Service Updates
Whenever maintenance updates that affect the generation of NetView automation tables via SA z/OS are available, the AT/MRT APAR Apply Options panel is displayed. An example is shown in Figure 249.

Choose one of the following options by entering its number:

1 Apply
Copies the changes from the shipped tables in SINGITBL into the +SA_PREDEFINED_MSGS MVC entry. Processing continues with the option that you specified before this panel was displayed (for example, displaying the Entry Type Selection panel or Report Selection Menu). The message APAR applied successfully is shown on the next panel.

The changes that were made by applying the APAR are in effect for the next configuration file build and no further action is required. This option is recommended if you have not made any user modifications for the AT build.
This option also creates a detailed report about the differences and writes it to an output data set (see "Service Updates Report"). The report member name is SRVyyddd where yy is the year and ddd is the day, for example, SRV09327.

2 Show

Creates a detailed report as for option 1 (see "Service Updates Report").

No data is copied to the +SA_PREDEFINED_MSGS MVC entry.

3 Skip

Does not copy any data and continues with the option that you specified before this panel was displayed. The message APAR apply deferred is shown on the next panel. The APAR Apply Options panel is displayed again when you next open a policy database. This option is recommended if you want to check the changes before they are applied to the policy database.

The END, RETURN, and CANCEL commands have the same effect as this option.

Note: The APAR application process is sufficient only if you use a NetView automation table that is generated by SA z/OS. If you maintain your own AT manually you need to make the necessary changes yourself.

The panel shows the APAR that caused the panel to be displayed. It is possible that multiple APARs are shown if more than one APAR with AT-related changes has been installed, or a policy database has not been opened for a long time and several APARs with AT changes have been installed in the meantime.

Service Updates Report

The service updates report provides details of:

- New messages together with the generated AT entry.
- Deleted messages together with the generated AT entry.
- Messages that have a modified AT entry. It includes the old and new generated AT entry.

Furthermore for all messages the user-defined AT entry is shown, if it exists in the policy database. The following shows an example of the report.

********************************************************************************
* Function Service Updates for AT build
* Policy Database name USER_PDB
* Policy Database data set name 'OPER1.USER.PDB'
* Version SA 3.4
* UserId OPER1
* Creation Date Monday, 12 Apr 2012
* Creation Time 16:12
*
********************************************************************************

This report includes the following APAR(s). For details see the APAR description(s):

********************************************************************************
SLS0451I - SLS No matching HOSTid

New predefined Automation Table entry:

IF MSGID = . 'SLS0451I' THEN
  EXEC(CMD('TERMMSG FINAL=NO,BREAK=YES') ROUTE(ONE %AOFPGSSOPER%));

User defined Automation Table entry:

NONE

SLS0452I - SLS Attach failed for module

New predefined Automation Table entry:

IF MSGID = . 'SLS0452I' THEN
  EXEC(CMD('TERMMSG FINAL=NO,BREAK=YES') ROUTE(ONE %AOFPGSSOPER%));

User defined Automation Table entry:

NONE

OMV999I - VPS Subsystem request failure

Old predefined Automation Table entry:

IF MSGID = 'OMV999I' THEN EXEC(CMD('TERMMSG FINAL=YES') ROUTE(ONE * AUTGSS AUTSYS AUTO1));

User defined Automation Table entry:

NONE

OMV999I - OMII component has ended

Old predefined Automation Table entry:

IF MSGID = 'OMV999I' THEN EXEC(CMD('TERMMSG FINAL=YES') ROUTE(ONE * AUTGSS AUTSYS AUTO1));

User defined Automation Table entry:

NONE
HSAL1108I - Refresh of the takeover file completed

Old predefined Automation Table entry:

IF MSGID = 'HSAL1108I' THEN
  EXEC(CMD('ACTIVMSG UP=YES') ROUTE(ONE %AOFOPGSSOPER%));

New predefined Automation Table entry:

IF MSGID = 'HSAL1108I' & ATF('ING$QRY STAT') = 'HALTED' THEN
  EXEC(CMD('ACTIVMSG UP=YES') ROUTE(ONE %AOFOPGSSOPER%));

User defined Automation Table entry:

IF MSGID = 'HSAL1108I' THEN
  EXEC(CMD('USEREXEC PARM1, PARM2') ROUTE(ONE %AOFOPGSSOPER%));

IEF238D - Allocation error

Old predefined Automation Table entry:

IF MSGID = 'IEF238D' & JOBNAME = SVJOB
  & ACQUIRE('PAD /''/ /CODE2=/ 1 W 6.1 STRIP N ,CODE3=/ N
  S KIPTO /,, S KIPTO /''/ 2.4 N') = CODES THEN
  EXEC(CMD('ISSUEACT CODE1=' SVJOB ', CODES ', SYSTEMMSG=YES')
  ROUTE(ONE %AOFOPGSSOPER%));

New predefined Automation Table entry:

IF MSGID = 'IEF238D' & JOBNAME = SVJOB
  & ACQUIRE('PAD /''/ /CODE2=/ 1 W 6.1 STRIP N ,CODE3=/ N
  S KIPTO /,, S KIPTO /''/ 3.6 STRIP N') = CODES THEN
  EXEC(CMD('ISSUEACT CODE1=' SVJOB ', CODES ', SYSTEMMSG=YES')
  ROUTE(ONE %AOFOPGSSOPER%));

User defined Automation Table entry:

NONE
Chapter 9. Data Management

This chapter provides information about SA z/OS functions for:
- Importing selected entries from another policy database
- Importing sample add-on policies
- Making bulk policy updates via flat files

Start Panel for Data Management

Option 5 (Data Management) on the Customization Dialog Primary Menu takes you to the Data Management Menu panel, shown in Figure 250.

Your options on this panel are:

1 Import from PDB
   This option enables you to select entries in a policy database and add them to the current policy database. See “Importing Policy Database Data” for details.

2 Import from Add-on
   This option enables you to select sample add-on policies and add all or part of their entries to the current policy database. See “Importing Sample Add-On Policies” on page 341 for details.

3 Update via File
   This option enables you to export the policy database data of a selected entry type to a flat file. After you have modified this data or added new entries, the flat file can be read in again. All the modified or newly added entries will be updated in the policy database or added to it without any further panel interaction. See “Making Updates with Text Files” on page 344 for details.

Importing Policy Database Data

Selecting option 1 (Import from PDB) on the Data Management Menu panel leads to the panel shown in Figure 251 on page 338.
This panel enables you to select entries in a policy database and add them to your current one, and view the import report for the target policy database. It has the following options:

1 Import Policy Data

   This option allows you to define the following:

   • An existing policy database that data will be imported from.
     Entering a question mark (?) in this field and pressing Enter displays the Policy Database Selection panel where you can select one of the existing policy databases.
     You can import data from the current database. This allows you to clone existing definitions for SYS, APL, APG and MTR entry types.

   • The particular entry type that selected entries will be imported for.
     Entering a question mark (?) in this field and pressing Enter displays the Entry Type Selection panel where you can select one of the existing entry types.

   • Whether you want to import entries that are linked to the entry type that you selected in the Entry type field.
     The following rules determine which linked entries are also imported, depending on the entry type that is imported:

     **GRP**
     Linked entry types SBG or SYS are also imported. For sysplex groups, any linked APG entry type of type SYSPLEX is imported.
     The links between resources where the group is involved are imported.

     **SYS**
     All entry types that are linked to the selected systems are imported along with the resources that belong to those systems and the links between the resources in the systems.

     **APG**
     Linked entry types APL, SVP or TRG are also imported.

     **APL**
     If the application has an object type of CLASS, its instances and their service periods and triggers are imported.
     If the application has an object type of INSTANCE, the class application that it is linked to and its service periods and triggers are imported.

     **TRG**
     The entries of entry type EVT that are selected in its startup or shutdown conditions are imported along with it.
Importing Policy Database Data

Note: Importing entry type ENT (enterprise policies) is not supported.

2 View import report

This allows you to view the import report that is available in the target policy database. It invokes the ISPF VIEW function. If there is no import report, an empty screen is displayed.

The information that you specify here is saved in your ISPF profile data set and used as the initial values the next time you issue the import function.

Selecting 1 Import Policy Data on this panel takes you to a list of existing entries for the specified type (see Figure 252 for an example).

Figure 252. Entry Name Selection Panel for Import Policy Data

Here you can select entries that you want to:
• Copy to your current policy database
• Get a modified copy of for your current policy database

Making selections and pressing Enter takes you to the panel shown in Figure 253 which displays the entries that you selected.

Figure 253. Selected Entry Names For Import Panel

This panel allows you to modify the selected entries before you import them. You can:
• Rename entries in the list by:
  – Overotyping their names
  – Entering RENAME (R) in the Action field

If there is already an entry with the same name and type in the current policy database, it is marked as a duplicate (column D is set to Y).
Importing Policy Database Data

- Overtype an entry's **Short Description** field (this only affects non-duplicate entries).
- Remove entries from the list by:
  - Blanking their names
  - Entering REMOVE (M) in the **Action** field
- Locate an entry by entering LOCATE (L) at the command line.
- Modify data using character string replacement by entering 1 or MODIFY (M) at the command line. This opens a Modify data pop-up window, as shown in Figure 254.

![AOFGPMOD Modify data](image)

**Figure 254. Modify Data Pop-Up Window**

Type data in the **Old string** and **New string** fields.
Replacement happens in a predefined set of fields in the import data. This set of fields supports the cloning of complete systems, for example, IMS, CICS, or SAP solutions.

You can update all relevant fields at once with a single modify action. This only works if the naming of a solution was done systematically (for example, consistently using an IMS subsystem ID of IMS1).

The modification for certain fields is case-sensitive. For a complete list of modifiable fields (including whether data modification is case-sensitive) see the online help.

**Note:**
No checks are made to prevent &AOCCLONE variables that are used in fields being modified.

- Check the data that is to be imported by entering 2 or CHECK (K) at the command line.
This executes an integrity check of the data and links to be imported so that there are no conflicts with definitions in the current policy database. Error messages for any conflicting data properties that are detected are listed in the import report. An import is not carried out.
- Start the import process by entering 3 or RUN (R) at the command line.
This executes the same data integrity check as with option 2 and, if no conflicts are found, imports the data into the current policy database.
- View the import report by entering 4 or VIEW (V) at the command line.
This allows you to view the most recent import or data check report that is available in the current policy database. It invokes the ISPF VIEW function. If there is no import report, an empty screen is displayed.
- Return to the Entry Name Selection panel by pressing the END or PF3 key.
Go to the Data Management panel by entering **CANCEL (C)** at the command line.

**Note:** If the entry is a class, column **C** is set to *****.

### Batch Import with Data Modification

SA z/OS provides a batch interface that can be invoked from a separate JCL. The SINGSAMP sample library contains a sample job for this task in the member INGEBIMP. Before submitting this job, you need to edit the jobcard appropriately as described in the sample itself.

The batch job needs the following input parameters:
- Target policy database name
- Target policy database data set name
- Source policy database name
- Source policy database data set name
- Entry name (to be imported)
- Entry type (to be imported)
- Whether entries linked to the selected entries as well as the links themselves should be imported
- List of strings to be modified, separated by blanks
- List of replacement strings, which must have as many words as the list of strings to be modified

The batch import produces the following return codes:

- **0** Successful.
- **8** Import did not run due to incomplete or conflicting specifications
- **20** Severe error: no data imported.

### Importing Sample Add-On Policies

SA z/OS comes with a number of sample add-on policies that you can import into an existing policy database to speed up database development. Importing add-on policies into a policy database means adding a copy of the sample to that policy database. The add-on policies themselves are not modified.

To import sample add-on policies:
1. Specify the *target* policy database that you want to import the sample *into* in the **Current Policy Data Base** field of the Customization Dialog Primary Menu and select option **5 Data Management**.
2. On the Data Management Menu panel, select option **2 Import from Add-on**. This takes you to the Import Add-On Policies panel, shown in Figure 255 on page 342 which displays a list of the add-on sample policies that SA z/OS provides.
Importing Sample Add-On Policies

YES in the **Customizable** column indicates that a policy has several components. If you enter C in the **Action** field for a customizable add-on policy, this displays the Select Add-on Policy Components panel. From the list of the components you can select or deselect the components that you want to include in the import.

3. Select the add-on samples that you want to import and invoke option **1 Import selected add-on policies**.

This performs a check of the selected policies (a pop-up window shows the progress of the checks). A panel similar to Figure 256 is then displayed.

4. This panel allows you to modify the selected entries before you import them.
Importing Sample Add-On Policies

If the entry is a class, column C is set to "*.*".
If there is already an entry with the same name and type in the current policy database, it is marked as a duplicate (column D is set to Y).

You can:

- Rename entries in the list by:
  – Overtyping their names
  – Entering RENAME (R) in the Action field
- Overttype an entry’s Short Description field (this only effects non-duplicate entries).
- Remove entries from the list by:
  – Blanking their names
  – Entering REMOVE (M) in the Action field
- Locate an entry by entering LOCATE (L) on the command line.
- Modify data using character string replacement by entering 1 or MODIFY (M) at the command line. See the online help for details of which entry types and fields you can use character string replacement with.
  Note that the modify action is supported only for a single selected add-on policy, rather than several simultaneously.
  A Modify data pop-up window opens, similar to Figure 254 on page 340. For details about using this window, see “Modifying Data” on page 340.
- Check the data that is to be imported by entering 2 or CHECK (K) at the command line.
  This executes an integrity check of the data and links to be imported so that there are no conflicts with definitions in the current policy database. Error messages for any conflicting data properties that are detected are listed in the import report. An import is not carried out.
- Start the import process by entering 3 or RUN (R) at the command line.
  This executes the same data integrity check as with option 2 and, if no conflicts are found, imports the data into the current policy database.
- View the import report by entering 4 or VIEW (V) at the command line.
  This allows you to view the most recent import or data check report that is available in the current policy database. It invokes the ISPF VIEW function. If there is no import report, an empty screen is displayed.
- Return to the Import Add-On Policies panel by pressing the END or PF3 key.
- Go to the Data Management panel by entering CANCEL (C) at the command line.
**Warning: Consistent Naming**

Importing an add-on policy includes importing all relationships and links between policy objects in that database. You therefore need to make sure that naming is consistent across your current policy database and the sample that you want to add if you start from one of the following:

- A copy of the *BASE sample policy database, but you modify it before importing an add-on policy
- A policy database that you have created yourself on the basis of the *EMPTY sample policy database

This is so that relationships and links can be resolved during the import operation. When there are relationships or links that cannot be resolved, an “unresolved link” message is displayed in the progress indication panel.

---

**Making Updates with Text Files**

You can carry out bulk updates of policy objects using a flat text file. You can select one or more policy objects and their associated policy items and have their data written to a text file. You can then edit the text file to update the policies and then import the updated file into your policy database.

This update method can be quicker than using the customization dialog if you want to update several entries, but you must take care that the syntax is correct.

Currently, only policy objects of type APG, APL, MTR, MVC, and UET are supported, as well as certain policies of entry types GRP and SYS.

Note that the file update facility does not allow concurrent user access.

For example, to update APL policy objects via a text file, follow these steps:

**Step 1. Edit the text file:**

Use option 5 (Edit output file) on the Policy Data Base Update Selection panel to edit the data set where you put your text policy data.

**Step 2. Check syntax (optional):**

Although a syntax check is performed when you import a text file, you can use option 2 (Perform syntax check) to check whether the syntax of the file data complies with the data in the policy database before you import the file. This can save you time because it allows you to check the syntax without invoking the complete import process.

**Step 3. Import the updated text file:**

On the Data Management Menu panel, make sure that the Input File Name field specifies the data set with your updated policy data. If you do not provide a name in this field, SA z/OS will put in the name of the data set previously used for output.

Type 3 on the option line and press Enter.

Further update processing is controlled by the Mode setting:

- **ONLINE**

  This is the default value. It indicates that the update should run in the foreground. It displays a command progress display.

  Processing is as follows:
Making Updates with Text Files

a. A syntax check of the whole text file is performed. Only when the syntax check has finished successfully does the actual policy database update take place.

b. If there is a timestamp in the text file and it is more recent than the last modification of entries in the policy database, the policy database is modified.

Otherwise a confirmation panel with the names of the most recently modified entries is displayed, where you can either confirm or cancel the update processing.

Messages from the update are written to a report data set that can be viewed with option 4 in the Policy Database Update Selection panel.

The data set with the text policy data is not affected by the import. In particular it is not deleted.

BATCH

This indicates that the update should run in the background.
Processing is as follows:

a. A batch job is started, executing the policy database update as soon as it obtains exclusive access to the database. This requires you to specify valid JOB statement information in the panel.

b. A syntax check of the whole text file is performed. Only when the syntax check has finished successfully, does the actual policy database update take place.

c. If there is a timestamp in the text file and it is more recent than the last modification of entries the policy database, the policy database is modified.

Otherwise the update process ends with messages in the report data set giving the reasons for ending without updating the policy database.

The settings in the Mode field and Job statement information are saved in your ISPF profile data set and are used as the initial values the next time you issue the UPDATE command.

Hint:

If you run the policy update via batch overnight, it is recommended to check the syntax first, to ensure the overnight update does not fail due to syntax errors.

Supported Policies

Apart from the policies that are available for applications, the following policies are supported for other entry types:

Groups

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSPLEX</td>
</tr>
<tr>
<td>– Sysplex Name</td>
</tr>
<tr>
<td>– Sysplex Timer Monitoring</td>
</tr>
<tr>
<td>– Number Monitored Sysplex Timer</td>
</tr>
<tr>
<td>– Temporary Data Set HLQ</td>
</tr>
<tr>
<td>– Started Task Job Name</td>
</tr>
</tbody>
</table>
Making Updates with Text Files

- Couple Data Set HLQ

Systems
DESCRIPTION
SYSTEM INFO
Operating System (Read-only)
Image/System name
AUTOMATION SYMBOLS
WHERE USED

Application Groups
DESCRIPTION
APPLGROUP INFO
RELATIONSHIPS
MESSAGES/USER DATA
WHERE USED

Monitor Resources
DESCRIPTION
MONITOR INFO
RELATIONSHIPS
MESSAGES/USER DATA
HEALTHSTATE
WHERE USED

MVS Components
DESCRIPTION
MESSAGES/USER DATA
MINOR RESOURCES
WHERE USED

User E-T Pairs
DESCRIPTION
E-T DATA
WHERE USED

Updating Policy Objects Using Text Files

Each policy object in the text file is represented by the following structure:

```
UPD <Entry Type>  - <Entry Name>
  <Policy Item> ( field|block|policy replacement )
    <Field>    : <Value>
    <Field>    : <Value>
  ...
  <Policy Item> ( field|block|policy replacement )
    <Field>    : <Value>
    <Field>    : <Value>
  ...
  ...
```

The actual values for the Entry Type, Entry Name, Policy Item and Field variables depend on the selections that you made when the file was created and what data is in your policy database, for example:

```
UPD APL
  RELATIONSHIPS ( policy replacement )
    Relationship Type : MAKEUNAVAILABLE
    Supporting Resource : AM2/APL=
    Description        : Prevents takeover at system shutdown
    Sequence Number : 
    Automation    : PASSIVE
```
Making Updates with Text Files

Chaining : WEAK
Condition : WhenDown
MESSAGES/USER DATA : no data available
WHERE USED APG : AM_X

Do not modify the <Policy Item> or <Field> segments because the syntax of the file will be invalid and the file will fail syntax checking.

If the file does fail the syntax check, you can view the update report and fix any errors that are reported before trying the update again.

The text after each policy item indicates which type of data replacement is possible:

- Field replacement: the data that you specify in the text file replaces only the corresponding data in the policy database.
- Block replacement: the row and all data that is nested underneath it is treated as one block of data, and each such block replaces the corresponding data in the policy database.
- Policy replacement: the policy definition in the text file replaces the whole policy definition in the policy database.

Note: The <Policy Item> lines must begin in column 2 or 3 or the file will fail the syntax check.

Field Replacement

With field replacement the specified data in the file replaces only the corresponding data in the policy database. A blank value for a field in the file discards a value that may be in the policy database or resets it to a default in the same way as for the input field on the panel. The data in any other fields that belong to a policy but are not specified in the text file remains unchanged.

Block Replacement

With block replacement a data row marked with a dash indicates a data block that consists of that row and all data that is nested underneath it. Each such block replaces the corresponding data in the policy database. Other blocks for the same policy item remain unchanged. A data block where you have edited all the fields to blank would delete all the corresponding data from the policy database. For example, the following would delete the reply definitions from a message:

Action - REPLY
Pass/Selection :
Retry Count :
Reply Text :

You can also specify just the block header line to delete all the data in that block. Thus, the equivalent for previous example would be:

Action - REPLY

Note: For your convenience you can also delete data by specifying:

Message ID : ABC123I

or
Making Updates with Text Files

Minor Resource Name : aMinResName

If no related detail lines follow, then all related details are deleted. In this case, all actions for message ABC123I as well as the message description will be deleted, but the message ID will not be deleted. Correspondingly, all definitions made for a minor resource will be deleted, but the minor resource itself will not be deleted.

Example 1: Create or Delete Messages for application TSO in a Policy Database: All of these changes are made in one file.

1. The data in the text file is initially:

   UPD APL - TSO
   MESSAGES/USER DATA ( block replacement )
   Message id : IKT001D
   Description : Active users during TCAS start
   Action : REPLY
   Pass/Selection :
   Retry Count :
   Reply Text : U

   Change Message ID IKT001D to TEBULK and add some REPLY data. This will result in a new message definition in the Policy Database:

   UPD APL - TSO
   MESSAGES/USER DATA ( block replacement )
   Message id : TEBULK
   Description : Active users during TCAS start
   Action : REPLY
   Pass/Selection :
   Retry Count :
   Reply Text : U
   Pass/Selection : *
   Retry Count : 1
   Reply Text : dump

2. Create a new Message ID TEST for the application TSO:

   UPD APL - TSO
   MESSAGES/USER DATA ( block replacement )
   ...
   Message id : TEST
   Description : test bulk
   Action : code
   Code 1 : 9999
   Code 2 :
   Code 3 :
   Value Returned : TERM

3. Delete description and all actions of Message ID 'ABC123' for application TSO:

   UPD APL - TSO
   MESSAGES/USER DATA ( block replacement )
   ...
   Message id : ABC123

Example 2: Change or Insert SHUTDOWN for the application class C_TWS in a Policy Database:

1. Delete pass 4 for PHASE NORM, change the pass from 2 to 4 for PHASE IMMED, and insert a reply for PHASE INIT:

   UPD APL - C_TWS
   SHUTDOWN ( block replacement )
   Phase : NORM
   Action : CMD
   Pass : 1
   Automated Function/"*":
Making Updates with Text Files

Policy Replacement

With policy replacement the data in the text file replaces the whole policy item definition in the policy database. For example:

- An application has two relationships defined for it in the policy database and one relationship in the text file. After the file update has been completed the application has only the one relationship from the text file.

- An application has some USS data defined in the policy database, while all USS fields in the text file for the application are blank. After the file update has been completed the application no longer has any USS data.

Empty lines are allowed anywhere in the file. They are ignored when the file update is performed. Comment lines are also allowed anywhere in the file. They have an asterisk in the first column and are skipped during file import.

If a field is specified several times, the last definition wins, even if it is an empty field. The following sequence in the APPLICATION INFO policy would remove any Owner data from the policy database for the corresponding application:

```
Owner : OPERATOR
Owner :
```

Example: Create, Change or Delete Policy Relationships for applications in a Policy Database: All of these changes are made in one file.

1. The data in the text file is initially:

```
UPD APL - JES2
RELATIONSHIPS (policy replacement)
Relationship Type : HASMONITOR
Supporting Resource : JES2MON/MTR/= Description : Relationship to JES2 Monitor
Sequence Number :
Automation :
Chaining :
Condition :
```

```
* Relationship Type : PREPUNAVAILABLE
Supporting Resource : VTAM/APL/= Description : Delay DRAINJES until VTAM is down
Sequence Number :
Automation : PASSIVE
Chaining : WEAK
Condition : WhenDown
```
Making Updates with Text Files

Change the second relationship for JES2:

```
UPD APL - JES2
RELATIONSHIPS (policy replacement)
  Relationship Type : HASMONITOR
  Supporting Resource : JES2MON/MTR/= 
  Description        : Relationship to JES2 Monitor
  Sequence Number    :
  Automation         :
  Chaining           :
  Condition          :

*  Relationship Type : PREPAVAILABLE
  Supporting Resource : VTAM/APL/= 
  Description        : Edit the description for test
  Sequence Number    : 1
  Automation         : PASSIVE
  Chaining           : WEAK
  Condition          : WhenDown
```

2. Add a new relationship to JES3. The data in the text file is initially:

```
UPD APL - JES3
* RELATIONSHIPS 
  no data available

Edit the file as follows:

```
...
UPD APL - JES3
RELATIONSHIPS (policy replacement)
  Relationship Type : HASMONITOR
  Supporting Resource : JES3MON/MTR/= 
  Description        : Relationship to JES3 Monitor
  Sequence Number    :
  Automation         :
  Chaining           :
  Condition          :
```

3. Delete the second relationship from TCPIP (blank all fields or delete all lines for the second relationship). The data in the text file is initially:

```
UPD APL - TCPIP
RELATIONSHIPS (policy replacement)
  Relationship Type : HASPARENT
  Supporting Resource : RESOLVER/APL/= 
  Description        : TCPIP requires RESOLVER for addr xlation
  Sequence Number    :
  Automation         :
  Chaining           :
  Condition          :

*  Relationship Type : HASPARENT
  Supporting Resource : VTAM/APL/= 
  Description        : TCPIP is dependent on VTAM
  Sequence Number    :
  Automation         :
  Chaining           :
  Condition          :
```

Edit the file, for example, as follows:

```
...
UPD APL - TCPIP
RELATIONSHIPS (policy replacement)
  Relationship Type : HASPARENT
  Supporting Resource : RESOLVER/APL/= 
  Description        : TCPIP requires RESOLVER for addr xlation
  Sequence Number    :
  Automation         :
  Chaining           :
  Condition          :
```
Creating New Policy Objects Using Text Files

To create new policy objects, insert sections that have the following structure:

```
NEW <Entry Type> : <new entry name>
COPY : <name of existing entry to copy data from>
WHERE_USED <parent entry type> : <name of existing entry to link to>
LINK_TO_CLASS : <name of existing APL class entry to link to>
```

You can create new policy objects for any entry type except for type ENT (enterprise).

The 'Policy Database Update Selection' panel provides an option to create templates for generating new entries (NEW template) in the policy or respectively deleting existing entries (DEL template) from the policy.

These sections can be inserted anywhere in the text file, with the following conditions:
- Policy object sections are not broken apart by other ones.
- The keywords (NEW, COPY, WHERE_USED, and LINK_TO_CLASS) must begin in column 1. Any <Policy Item> lines must begin in column 2 or 3. All other data must be indented.

The keywords LINK_TO_CLASS (for entry type APL only), COPY, and WHERE_USED are optional. If they are specified, they can appear in any order.

The NEW keyword allows additional specifications, for example, subsystem name, in the same notation as for the update function. All the fields that are available on the NEW panel for the entry type that you are creating are also valid here. Ensure that the field names are consistent with the NEW panel used in the Customization Dialogs.

The COPY keyword works in the same way as the COPY policy in the customization dialog panels. This means that the same limitations apply with respect to type, subtype, or class-instance compatibilities.

For each keyword, the specification of only one value is allowed. To link an application to several application groups, the WHERE_USED keyword may appear as many times as you require.

The WHERE_USED keyword may need an optional parameter, <parent entry type>, for entries that might be linked to several parent types, for example, Service Periods to Applications, Application Groups or Monitor Resources.

If the parent entry type is omitted, the following defaults are assumed:
- For SYS the default is GRP
- For SVP the default is APL
- For TRG the default is APL.
The parent entry type for Application Groups defaults according to the Application Group's type. If the type is SYSPLEX the default link parent entry type is Group, otherwise it is System.

If either of the COPY or LINK_TO_CLASS keywords is specified several times, the last one is used (the last specification overwrites any previous one) even if it is blank.

The following example shows a more elaborate definition:

```
NEW APL : $IMS711_MSG_REGION5
  Subsystem Name : IMS711M5
  Application Type : IMS
  Job Name : IMS711M5
  Short description : IMS V7 - IMS711 Message region on SYS1
  LINK_TO_CLASS : CLASS_IMS_TP
  WHERE_USED APG : IMS711_CONTROL
  WHERE_USED APG : APPLGR_MAINT
  COPY : IMS713M0
```

**Note:**
You should take care with the order of your definitions. It is possible that a class link that you set with a LINK_TO_CLASS definition is overwritten by a class link that is inherited by a COPY command that is executed after the LINK_TO_CLASS definition.

The instructions to create new entries and the data to modify existing fields are both run from a flat file. As a consequence they can be combined so that the following would be possible:

```
NEW APL : IMS721RC
COPY : IMS713RC
WHERE_USED : IMS721 CONTROL
LINK_TO_CLASS: CLASS_IMS_DBRC

UPD APL : IMS721RC
DESCRIPTION
  Short description : IMS V7 - IMS721 DBRC region on SYS2
RELATIONSHIPS
  Relationship Type : HASPARENT
  Supporting Resource : IMS721C4/APL/= Description :
  Sequence Number : 1 Automation :
  Chaining :
  Condition :
```

If the keyword NEW or UPD is specified more than once with the same application name, only the last specification is processed. All previous are ignored. If a specification is ignored, all data rows are skipped up to the next occurrence of NEW or UPD keyword. If a NEW or UPD keyword is skipped, an informational message is issued.

Here are some examples for the NEW keyword with all additional specifications for the following entry types:

- APL
- APG
- MTR
- UET
These are the most common entry types that you are likely to want to create new policy objects for via file updates.

**Application**
The following is an example for a new IMS Application Class entry, APLNEW, with data copied from an existing application, OTHERAPL, and linked to another class, IMSCLASS1:

NEW APL : APLNEW
Subsystem Name : APLNEW
Object Type : CLASS
Application Type : IMS
Subtype : CTL
Job Type :
Job Name :
Transient Rerun :
Scheduling Subsystem :
JCL Procedure Name :
Short Description : New IMS application
Long Description 1 :
Long Description 2 :
Long Description 3 :
Long Description 4 :
Long Description 5 :
COPY : OTHERAPL
LINK_TO_CLASS : IMSCLASS1

**Application Group**
The following is an example for a new system Application Group entry, APGNEW, with data copied from an existing application group, OTHERAPG, and linked to an existing system, SYSTEM1:

NEW APG : APGNEW
Type : SYSTEM
Nature : BASIC
Default Preference : *DEF
Automation Name : APGAUTO
Automatically link : YES
Behaviour : ACTIVE
Move Mode :
Prepare Move : YES
Short Description : New Application Group of type System
Long Description 1 :
Long Description 2 :
Long Description 3 :
Long Description 4 :
Long Description 5 :
COPY : OTHERAPG
WHERE_USED SYS : SYSTEM1

**Monitor Resource**
The following is an example for a new Monitor Resource entry, MTRNEW:

NEW MTR : MTRNEW
Monitored object : OBJECT1
Monitored jobname : JOB1
Short Description : New Monitor Resource
Long Description 1 :
Long Description 2 :
Long Description 3 :
Long Description 4 :
Long Description 5 :

**User Entry-Type Pair**
The following is an example for a new User E-T Pair entry, UETNEW:
Deleting Policy Objects Using Text Files

You can delete policy objects for any entry type except for type ENT (Enterprise).

The DEL keyword has the following syntax:

```
DEL <Entry Type> : <existing entry name>
```

The keyword can appear anywhere in the file with the following restrictions:

- the keyword DEL must begin in column 1,
- the keyword does not break other sequences, for example, a new entry consisting of NEW and related COPY, WHERE_USED keywords,
- no detail specifications are needed in subsequent lines (unlike, for example, NEW or UPD).

For each specification of DEL, only one entry name is allowed. To delete multiple entries, a single DEL specification is required for each entry.

If there are multiple DEL specifications for the same entry type/name this is not handled as an error. The entry is deleted and WARNING messages are issued additionally for each duplicate specification to indicate there may be an error.

An entry can only be deleted via the DEL keyword if it is not referenced anywhere else in the file. For example, an application can only be deleted via file if it is not modified via UPD or created via NEW in the same file. If there is a need to delete an entry and recreate it, then the corresponding specifications have to be split into two files that need to be run one after the other.

Generating a Job for Batch Update

You can run a job for batch update that performs the same processing as the UPDATE function of the customization dialog. The SINGSAMP sample library contains a sample job for this task in member INGEBFLT. Before submitting this job, you need to edit the jobcard appropriately as described in the sample itself.

The batch update produces the following return codes:

- **0** Successful
- **4** Syntax errors or timestamp conflicts were detected; the update was terminated. See the job log or report file for details.
- **8** The update text file could not be read or the report data set could not be reallocated. The update was terminated.
- **>8** Severe error; update terminated. See the job log for details.
Chapter 10. Conversion Function

This chapter provides information on what you should be aware of before using the new release of SA z/OS. In detail, the following topics are discussed:

- "Moved and Restructured Data"
- "Initial Policy Database Conversion" on page 356

The conversion function creates a report called $RPTCNV as a member in the data set of the policy database. All messages are logged.

Moved and Restructured Data

Converting from SA z/OS 3.2

The following changes apply when converting from SA z/OS 3.2:

- The values eServer, 9672, 9674 and 3000 are converted to Mainframe.
- The Connection Protocol is defaulted to TCPIP for PSM processors.
- The ProcOps Target HW Name is defaulted to the processors entry name for SNMP and NVC processors.
- The Auth Token value for SNMP and NVC processors is forced to blank.
- The Command Retry value for SNMP processors is defaulted to 10.
- The Command Retry Wait Time for SNMP processors is defaulted to 1.
- For non-MVS systems the data from the old Image/ProcOps name is taken to set the new System/Image name.
- For MVS systems the data from the old MVS system name is taken to set the System/Image name. If the old MVS system name and the Image/ProcOps name are not identical, conversion stops with a message telling you that this must first be changed.
- All data of IRN and ISF entry types is deleted.
- Data from minor resource flags and thresholds are merged into the MINOR RESOURCES policy item.
- Forces the CODE3 definitions for the message ID INGALERT to the value IOM.
- The character % in CODE1, 2, and 3 definitions are escaped, that is, prefixed with a \ character, to indicate that the % character is not a wildcard character.

The conversion process issues a warning message if:

- There are thresholds in UET entries
- There are processor definitions that are missing mandatory data

The conversion process transforms AT specifications in the following cases:

- If you have specified message IEF403I with a status of UP in the MESSAGES/USER DATA policy item for applications, the new field Skip ACTIVE status in the APPLICATION INFO policy is set to YES and this message entry is deleted.
- If multiple overrides for one message are combined into a single new override the keywords *BEGINAPL and *ENDAPL are inserted for applications and keywords *BEGINMVC and *ENDMVC are inserted for MVS components.
Moved and Restructured Data

- The variable &SUBSJOB is replaced with &*JOBNAME. to have a clear distinction from &SUBSJOB which is a runtime variable, while &*JOBNAME. is resolved during AT build. Whenever the variable is replaced, an informational message is issued.
- For any conversion of AT data an informational message is issued.

Converting from SA z/OS 3.3

The following changes apply when converting from SA z/OS 3.3:
- All data of ISA and TMO entry types is deleted
- Status INACTIVE is deleted from Monitor Resources
- Internal data structure for Automation Flags and Minor Resources changed.

Initial Policy Database Conversion

The first time that a policy database is opened with the customization dialog, conversion to the current layout is done automatically. There are special conversion options available by selecting 0 Settings in the Customization Dialog Primary Menu, and then 4 Convert in the Customization Dialog Settings Menu. This displays the PDB Convert Option panel. This panel is used to specify the options for the PDB Conversion. These options are saved in your ISPF profile pool and remain in effect until set again.

On this panel, SA z/OS allows you to request the generation of an automation name for application groups during the very first conversion of a policy database. If an application group has an automation name, SA z/OS can generate resources from such entries.

Selective Policy Database Conversion

After your entries have been migrated to the policy database, the conversion is done to convert the old data which might have been migrated to the policy database to the new format. The migration functions automatically start the conversion function.

Generating a Job for Batch Conversion

You can run a job for batch conversion that performs the same processing as the conversion function of the customization dialog. The SINGSAMP sample library contains a sample job for this task in member INGEBCON. Before submitting this job, you need to edit the jobcard appropriately as described in the sample itself.
Appendix A. Concurrent Customization by Multiple Users

Concurrent user access is supported for entry types SYS, APG, APL, MSG, PRO, MTR, ENS, MVC, and UET. These are referred to as “multi-user capable entry types”.

The following subsections describe the consequences for users that want to work simultaneously on entry types within one policy database.

Concurrent User Access Capabilities

If a policy database is selected for processing (using the OPEN command), 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 the multi-user capable ones with several users, or performing commands other than OPEN or SELECT against a policy database while an entry in a multi-user capable entry type is being edited, may not work.

This limitation depends on the data tables that are needed for the selected action or command, and it is indicated by specific messages. The most common situations are described in the following sections.

Concurrent User Access within the Customization Dialog

The customization dialog allows the editing of certain entry types by several users at the same time. To prevent simultaneous usage of the same resource, serialization is implemented using the ENQ and DEQ macros with global resources. If an installation runs several systems in a global resource serialization (GRS) complex, but GRS is not installed, you should ensure that the following major names (qname of the ENQ and DEQ macros) are propagated:

- **INGPDB** This name is used by SA z/OS to serialize access within the customization dialog. It is mandatory to have this major name distributed among all systems where the customization dialog may run. Otherwise different users may overwrite their changes and data may get lost.

- **SYSDSN** This name is used by SA z/OS to obtain the user ID of a resource holder in case a policy database cannot be accessed. If this major name is not distributed, messages indicating that a resource cannot be accessed would not show a user ID.

- **SPFEDIT** This name is used by SA z/OS to obtain the user ID of a resource holder in case an ISPF table, which holds the data within a policy database, cannot be accessed. If this major name is not distributed, messages indicating that a resource cannot be accessed would not show a user ID.

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 257 on page 361), while another user works with a policy database, the selection is not possible and an appropriate message is displayed.
Concurrent User Access within the Customization Dialog

On the other hand, if you are accessing the SA z/OS policy items for modification, this will prevent other users from accessing any policy item and an appropriate message is displayed.

Policy Database Selection
If you want to issue a BUILD, DELETE, or REPORT command from the Policy Database Selection panel or use a DATA MANAGEMENT function while another user has accessed the same policy database, the command is rejected and an appropriate message is displayed.

Conversely, if you request access to a policy database via S (Select) or O (Open), while a BUILD, DELETE, DATA MANAGEMENT function, or REPORT is in progress for this policy database, the access is rejected and an appropriate message is displayed.

For other commands there are also limitations, if a user has already opened the specific policy database. These limitations depend upon the selected entry types and actions, and are indicated by an appropriate message.

Entry Type Selection
If a policy database is opened successfully (with S (Select) or O (Open)), the Entry Type Selection panel is displayed. Here a specific Entry Type can be selected, or a BUILD can be requested.

If a BUILD command is issued from this screen, but another user has also opened this policy database to work with applications or any other entry type, the command is rejected and an appropriate message is displayed.

If an entry type other than the multi-user capable ones is selected, and another user has also selected an entry type of the same policy database, the selection may be rejected depending on the data tables needed for the selected entry type, and an appropriate message is displayed. In this case, editing is not possible, because concurrent user access is provided only for the multi-user capable entry types.

Entry Name Selection
On the Entry Name Selection panel a specific application can be selected, and a BUILD can be requested.

If you try a BUILD action, and the policy database has been simultaneously opened by another user, the command is rejected and an appropriate message is displayed.

If another user is currently editing the same application in the same policy database, access is allowed only to view the data, that is, changing the displayed data is not allowed. In this case, however, it is possible to select that application and view the data. A warning message is displayed on the Policy Selection panel, indicating that no updates will be accepted.

If, nevertheless, the displayed values are changed on the data entry panel for that policy option, these changes are not accepted when leaving that panel. If you try to leave the panel with the END command, a message will remind you that you must leave this panel with CANCEL.
Changing Links

The WHERE USED policy item for Systems allows you to change the links of one system to one or more groups or subgroups.

The WHERE USED policy item for Applications allows you to change the links of one application to one or more application groups.

The WHERE USED policy item for ApplicationGroups allows you to change the links of one application group to one or more systems or sysplexes.

The WHERE USED policy item for Monitor Resources allows you to change the links of one monitor resource to one or more monitor resources.

You can only add or remove such links, if you exclusively access the entry types System, Application, ApplicationGroup, or Monitor Resource. If another user is currently editing them, a message is displayed indicating that updates are not possible.

The same is true for the trigger policy item where you can maintain the link of the current application, application group, or monitor resource to a trigger.
Concurrent User Access within the Customization Dialog
Appendix B. More about Using the Customization Dialog

Use this appendix to find out how to get help when using the customization dialog and how to change the customization dialog environment.

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 Database Selection panel.

The first three options in the Customization Dialog Settings Menu panel (see [Figure 257]) allow you to change the customization dialog environment for your user ID. This is a usability function only and has no effect on the policy database contents.

```
MENU HELP
-----------------------------------------------
AOFOCUST Settings Menu
Option ===>

The following options apply only to your userid:

  1 Alias Define alias for standard commands
  2 Display Define display characteristics
  3 Message Define message and logging options
  4 Convert Define policy database conversion options
  5 Color Define color for inherited class values

The following options can be made available to all SA z/OS users:

  10 Policy Modify SA z/OS policy definitions
  11 Dsnames Define SA z/OS data set names
  12 Workdsn Define ISPF temporary data sets for batch build

Figure 257. SA z/OS 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 the messages are written to the ISPF log.
Changing Your Customization Dialog Environment

Note

If you want to use SA z/OS 3.3 processor operations ACF entries to build a processor operations control file for a focal point system that is running an earlier release of SA z/OS, use this option to specify whether SA z/OS should invoke additional checking for processor operations-related warning messages during configuration build. SA z/OS writes these messages to the build report $BLDRPT. The default value is OFF.

These settings are stored in the ISPF application profile. To have them available for batch processing (build or migration), 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 367.

Convert

This option lets you specify whether SA z/OS should create an automation name for an ApplicationGroup resource during the very first conversion of a policy database. See "Initial Policy Database Conversion" on page 356 for more information.

Color

This option allows you to specify the color and reverse video option that is used to display inherited class values.

Policy

See "Extending Policy Definitions" on page 363.

Dsnames

See "Defining Data Sets for Batch Processing" on page 367.

Workdsn

See "Defining ISPF Temporary Data Sets for Batch Build" on page 369.

Enabling and Disabling AT / MRT Override Syntax Checking

SA z/OS allows you to specify overrides for NetView automation table entries or Message Revision Table entries for message definitions in the MESSAGES/USER DATA policy. For these overrides a syntax check is performed which, however, requires a special setup because the check is not run within the customization dialog itself, but within a NetView address space.

If a syntax check cannot be performed due to communication problems with NetView or because NetView is not available at all, a warning message is issued. This message is issued whenever an override is modified. To avoid this message, the checking can be disabled with the following option on the 'Message' selection of the 'Settings Menu' in Figure 257 on page 361.

Enable checking of Automation Table overrides requiring NetView:
AT syntax check......YES (YES NO)

Figure 258. Automation Table Override Checking

The installation prerequisites are:
• the REXX function package INGTXFPG available in TSO/E
• a NetView running on the local system
Changing Your Customization Dialog Environment

• the command receiver task CMDRCVR and the automation operators AUTCMD0x must be up and running in the local NetView.

The details are described in the "Command Receivers" chapter of IBM Tivoli System Automation for z/OS Customizing and Programming.

The syntax check run completes the following steps:
• The SA synonyms ('%INCLUDE AOFMSGSY') are inserted on top of the AT override
• The AT or MRT override is sent to the local NetView via the command receiver infrastructure
• A temporary in-storage member TEMPAT is created
• The AT syntax is checked with the command 'AUTOTBL MEMBER=TEMPAT,TEST'
• The MRT syntax is checked with the command 'REVISE MSG MEMBER=TEMPAT,TEST'
• Each check is recorded in the netlog with the message 'ING335I Execute remote command on behalf of JOBSD=RTL CLIST=AOFSTADV COMMAND=.....'
• The result of the syntax check, which is sent to the console, is caught by the work operator and returned to TSO
• In the customization dialog some superfluous messages like 'DSI412I THE FOLLOWING ERRORS ENCOUNTERED IN PROCESSING DSIPARM MEMBER TEMPAT' are dropped and the remaining syntax error messages are displayed in a pop-up window.

Extending Policy Definitions

With SA z/OS you can extend your policy definitions by creating new policy items or modifying existing ones. Any user-defined policy items that you create or modify are now kept separate from the SA z/OS policies. This means that with future releases of SA z/OS you will not have to redefine those policy items that you have created or modified.

To create new policy items or modify existing ones, you should perform the following steps:
1. Select option 10 Policy from the Customization Dialog Settings Menu (see Figure 257 on page 361). An Entry Type Selection panel is displayed.
2. Select the entry type that you want to create a new policy item or modify one for. The Edit Policy Selections panel is displayed, listing all current policy items for that entry type, as shown in Figure 259 on page 364. Any user-defined policy items are indicated by USR in the Status column.
3. In the Edit Policy Selections panel you can:
   - Create a new entry by entering NEW at the command line. The new entry is added to the top of the list.
   - Create a new entry by entering INSERT in the Act column. The new entry is added after the entry where INSERT was specified.
   - Edit an entry by entering EDIT in the Act column.
   - Delete a user-defined entry by entering DELETE in the Act column.
   - Suppress an SA z/OS entry by entering REMOVE in the Act column.
   - Activate a deselected entry by entering SELECT in the Act column.
   - Leave the panel and save all changes by entering the END command.
   - Discard all changes by entering the CANCEL command.

   **Note:**
   Any modifications to the existing SA z/OS policy definitions are solely your responsibility.

4. If you choose to insert a new policy item for an entry type, or edit an existing policy item, the Define Policy Selection panel is displayed. The example in Figure 260 on page 365 shows how to define policy item APPLICATION INFO of entry type Application.
If you want to create policy items you must specify an ISPF action for the selection of the new policy item from the Policy Selection panel. You must then create your own ISPF input panel for the new policy item.

Each policy selection consists of information defining:

- How the policy appears on the Policy Selection panel.
- How the policy is processed when it is selected.
- Under what conditions the policy should be displayed.

This panel has the following fields:

**Entry Type**
Indicates the entry that the policy is being defined for. This field is set based on the entry type you had previously selected.

**Policy Name**
This is the name of the policy you are defining. It appears in the first column of the Policy Selection panel. This field is required.

**Description**
This is a short description of the policy. It appears in the second column of the Policy Selection panel. This field is required.

**Help Action**
Specifies the action to be taken when policy help is requested (by entering HELP in the Action column on the Policy Selection panel). The action must be a valid ISPF SELECT service. For example, to display an ISPF panel enter:

```
PGM(ISPTUTOR) PARM(AOFHLPA0)
```

The help text should describe the function provided by this policy rather than describing how to enter the data for the policy.

This field is optional; if no action is specified HELP will not be available from the Policy Selection panel for this policy.

**Edit Action**
Specifies the action to be taken when the policy is selected from the Policy Selection panel. The action must be a valid ISPF SELECT Service. This field is required.

---

**Figure 260. Define Policy Selection Panel**

If you want to create policy items you must specify an ISPF action for the selection of the new policy item from the Policy Selection panel. You must then create your own ISPF input panel for the new policy item.

Each policy selection consists of information defining:

- How the policy appears on the Policy Selection panel.
- How the policy is processed when it is selected.
- Under what conditions the policy should be displayed.

This panel has the following fields:

**Entry Type**
Indicates the entry that the policy is being defined for. This field is set based on the entry type you had previously selected.

**Policy Name**
This is the name of the policy you are defining. It appears in the first column of the Policy Selection panel. This field is required.

**Description**
This is a short description of the policy. It appears in the second column of the Policy Selection panel. This field is required.

**Help Action**
Specifies the action to be taken when policy help is requested (by entering HELP in the Action column on the Policy Selection panel). The action must be a valid ISPF SELECT service. For example, to display an ISPF panel enter:

```
PGM(ISPTUTOR) PARM(AOFHLPA0)
```

The help text should describe the function provided by this policy rather than describing how to enter the data for the policy.

This field is optional; if no action is specified HELP will not be available from the Policy Selection panel for this policy.

**Edit Action**
Specifies the action to be taken when the policy is selected from the Policy Selection panel. The action must be a valid ISPF SELECT Service. This field is required.
User Note
These fields are not shown on the Policy Selection panel. They can be used for documentation purposes. These fields are optional.

You can also specify filtering conditions, that is, attributes of the entry being processed from the Policy Selection panel that can be used to determine whether this policy should be displayed. The following filtering information can be specified:

Filter attributes
This specifies attributes that an entry must have in order for this policy to be displayed. This field contains multiple words, each word identifying one attribute. All the attributes specified must be satisfied in order for the policy to be displayed. If no attributes are specified then the policy is valid for all entries. The attributes that can be specified are defined by SA z/OS and differ based on the entry type, as shown in Table 8.

<table>
<thead>
<tr>
<th>Entry Type</th>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td>Group type</td>
<td>(blank) STANDARD SYSPLEX</td>
</tr>
<tr>
<td>System</td>
<td>Operating System type</td>
<td>(blank) CF LINUX MVS VM VSE</td>
</tr>
<tr>
<td>ApplicationGroup</td>
<td>ApplicationGroup type</td>
<td>(blank) SYSPLEX SYSTEM</td>
</tr>
<tr>
<td>Application</td>
<td>Object type</td>
<td>(blank) CLASS INSTANCE</td>
</tr>
<tr>
<td></td>
<td>Application type</td>
<td>(blank) IMAGE JES2 JES3 CICS IMS DB2 OPC USS</td>
</tr>
<tr>
<td>Processor</td>
<td>Processor type/mode</td>
<td>(blank) LPAR ESA PSM</td>
</tr>
<tr>
<td>Others</td>
<td></td>
<td>-</td>
</tr>
</tbody>
</table>

The following rules apply to the filtering attributes:
1. If no attributes are specified (indicated by (blank) in Table 8), the policy is always displayed for all entries.
2. If multiple attributes are specified then all attributes must be satisfied in order for the policy to be displayed. For example:
Changing Your Customization Dialog Environment

JES2 CLASS This displays the policy for application entries that are classes and are of type JES2.

JES2 This displays the policy for application entries that are of type JES2 and either class or instances.

3. For each ATTRIBUTE, the words listed in the VALUE column are mutually exclusive. If you specify both values the policy will never be displayed as both conditions cannot be true at the same time. For example:

JES2 JES3 This is never displayed because an application can be either JES2 or JES3 but not both.

4. No validity checking is done on the attribute values. If an invalid value is specified the policy will never be displayed since the condition cannot be satisfied. For example:

JES4 This is never displayed because an application cannot be of type JES4.

Defining Data Sets for Batch Processing

Use Option 11 of the Customization Dialog Settings Menu panel to define the batch processing data sets. This needs to be done only once when you install SA z/OS. For details, see [IBM Tivoli System Automation for z/OS 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.

```
 Figure 261. SA z/OS Data Set Customization Details Panel

On the Data Set Customization Details panel, a table is displayed with groups of data definition (DD) names and data set names. You must modify the data set names in accordance with your local naming conventions.

Each column can be modified as follows:
```
### DDname

No new DD names can be added, but a new entry for an existing DD name can be added at the end of the table. It is inserted at the correct position within the table when it is saved. An entry can be deleted by blanking out the DD name, but if an entry is removed incorrectly the batch functions become inoperative.

### Seq

This entry should be a number from 1 through 99. This value positions the entry in the correct concatenation sequence when the JCL is built. The higher the number, the further down the concatenation sequence it appears. 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 is represented by this DD name.

### Volume

May be left blank for catalogued data sets, otherwise enter the label ID of the volume where the data set is located (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 space managed data sets, otherwise you must enter the unit relevant to the volume where the data set is located (as would be coded for an ALLOC JCL entry).

AOFIPDB is needed to get access to the automation manager logic file.

For AOFTLIB the following data sets may be additionally allocated to the ISPF and SA z/OS table libraries:

- The user's ISPF table output library is needed to get access to the tables with user modifications, for example, the data set specifications
- The user's ISPF profile may be added to obtain access to the dialog, messages and logging options. If you omit this, the options that provide the most information are always used. This relates only to the settings in option 3 Message of the Customization Dialog Settings Menu panel.

In the example in Figure 261 on page 367 these are:

- `OPER.ISPF.ISPPROF`
- `OPER.USER.SINGITBL`

To avoid ISPF enqueues with TSO users a temporary data set is always added as the first data set of the ISPTLIB data set allocations.

---

**Note:**

The user's ISPF table output library should also be allocated to DDname AOFTABL for functions such as the PDB report or updating the PDB using text files.

The ISPF load libraries should be allocated to DDname AOFSTEP if they have not been defined in LPA.
Changing Your Customization Dialog Environment

To avoid ISPF enqueues with the ISPF input table library name, a temporary data set is always added as the first data set of the ISPTLIB data set allocation.

**Defining ISPF Temporary Data Sets for Batch Build**

Use Option 12 in the Customization Dialog Settings Menu panel to define the data sets for the batch build. SA z/OS displays the ISPF Temporary Data Sets Preallocation for Batch panel (see Figure 262).

```
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=SYSLDATA,SPACE=(CYL,(5,10)), DCB=(LRECL=256,RECFM=FB)
//ISPWRK2 DD DISP=NEW,UNIT=SYSLDATA,SPACE=(CYL,(5,10)), DCB=(LRECL=256,RECFM=FB)
//ISPWRK3 DD DISP=NEW,UNIT=SYSLDATA,SPACE=(CYL,(5,10)), DCB=(LRECL=256,RECFM=FB)
//ISPWRK4 DD DISP=NEW,UNIT=SYSLDATA,SPACE=(CYL,(5,10)), DCB=(LRECL=256,RECFM=FB)
//ISPCTL1 DD DISP=NEW,UNIT=SYSLDATA,SPACE=(CYL,(5,10)), DCB=(LRECL=80,RECFM=FB)
```

Figure 262. ISPF Temporary Data Sets Preallocation for Batch Panel

This panel allows you to add or modify job control data to preallocate temporary ISPF data sets that are used for the batch build. The build function creates the ACF and AMC output files in two steps:

1. All members are created and stored in a temporary file using ISPF file tailoring.
2. This file is written to disk creating all the individual members.

To hold the intermediate files, ISPF work and control data sets are used. If the default allocations of ISPF are not sufficient, they can be changed on this panel.

The ISPWRKx ddnames are used by ISPF for file tailoring services. The ISPCTLx ddnames are used by ISPF to hold the file tailoring output and to pass it to the subsequent functions that write the members to the output data set.

**User-written Functions**

Option **U User** on the Customization Dialog Primary Menu displays the User-written Functions Selection Menu. This menu allows you to add installation-specific functions to the customization dialog.

As an example, one selection has already been provided:

1 - Transmit

Transmit build output to a remote node

If you select this option, SA z/OS displays the Send Policy Data Base panel (see Figure 263 on page 370), where you can distribute policy databases to remote nodes.
To add further selections, edit the panel (member AOFGUSR) and do the following:

- Add selection number and description to the panel)BODY section
- Add a line to the TRANSLATE instruction in the panel)PROC section, containing the selection number and the command string which invokes the new function.

The Send Policy Data Base panel allows you to transmit a build output data set to a specified node and user ID.

```
MENU HELP
------------------------------------------------------------------------------
AOFGCMDX Send Automation Configuration Data Sets
Command ===>
To transmit the build output data set to a remote node, specify or revise the following values:
- Data set name... 'OPER1.USER.SOCNTL'
- Target:
  - User ID or nickname... OPER1
  - Node ID... SYS1
- 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 263. Send Automation Configuration Data Sets 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
Alternate Configuration Support

- allowing a specific build for normal or alternate configurations
- enforcing the build of normal or alternate configurations into different output data sets

All other definitions and links remain unchanged, for example, an application group can be linked to a normal systems, an alternate systems, or to both.

How to Define Alternate Configurations for Systems

For each system, either on the Define New Entry panel or in the policy item SYSTEM INFO, the existing MVS system name in the MVS SYSNAME field allows you to add a "+" sign to the system name. This is 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 the ProcOps name field) is defaulted with the MVS system name, any "+" is removed.

When linking systems to a group or subgroup, the configuration type is not checked. If, for example, a sysplex is intended for an alternate configuration, you have to ensure that only systems for this configuration are linked to that sysplex.

How to Build Alternate Configurations

The Build Parameters panel (see Figure 246 on page 319) has an option to select either a normal or alternate configuration build. The default is NORMAL. To avoid overwriting an existing configuration with wrong data, the build checks the configuration type in the build output data set. If there is already a configuration of a different type, the build is not started, and a warning message is displayed. If the policy database contains a normal and an alternate configuration, they have to be built into different output data sets.

Furthermore, the policy database name is checked to prevent an automation control file being destroyed by building a configuration into the wrong output data set.

When building an alternate configuration, all "+" signs are removed from the MVS system name. Thus the generated automation configuration files can be used for automation simply by switching the data sets.

If a policy database contains two configurations, the build picks out the systems that are linked to the selected configuration:

- The ACF map 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 standalone system. In each sysplex, it contains only the systems that are marked for the selected configuration, and standalone system members are created only for systems of the selected configuration.
- The ATs are generated only for the systems of the selected configuration.
Alternate Configuration Support

- The MPFLSTSA member is built for the messages that belong to the selected configuration.

Restrictions

For performance reasons, building the complete policy database (option 1), does not check which system each object is linked to. Thus the build output data set will contain fragments of each object in the policy database, regardless of which configuration they are linked to. Of course, fragments of the wrong configuration cannot be accessed by automation because they are not included via system main include members. However, the build output data sets need to be large enough to hold all fragments.

Furthermore, with build option 3 (to build an entry type or entry name), it is possible to update each fragment in a data set, independent of its link to a configuration.

*The Unlinked report* (see Chapter 6, “Creating Reports about Policy Databases,” on page 309) 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 [IBM Tivoli System Automation for z/OS Customizing and Programming](#).
Appendix C. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user’s responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

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

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

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Websites are provided for convenience only and do not in any manner serve as an endorsement of those Websites. The materials at those Websites are not part of the materials for this IBM product and use of those Websites is at your own risk.
IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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

IBM Deutschland Research & Development GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany

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

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming Interface Information

This publication documents information that is not intended to be used as a programming interface of IBM Tivoli System Automation for z/OS.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.
Glossary

This glossary includes terms and definitions from:
- The Information Technology Vocabulary developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.

The following cross-references are used in this glossary:
- Contrast with. This refers to a term that has an opposed or substantively different meaning.
- Deprecated term for. This indicates that the term should not be used. It refers to a preferred term, which is defined in its proper place in the glossary.
- See. This refers the reader to multiple-word terms in which this term appears.
- See also. This refers the reader to terms that have a related, but not synonymous, meaning.
- Synonym for. This indicates that the term has the same meaning as a preferred term, which is defined in the glossary.
- Synonymous with. This is a backward reference from a defined term to all other terms that have the same meaning.

A

ACF. See automation configuration file

ACF/NCP. Advanced Communications Function for the Network Control Program. See Advanced Communications Function and Network Control Program

ACF/VTAM. Advanced Communications Function for the Virtual Telecommunications Access Method. Synonym for VTAM. See Advanced Communications Function and Virtual Telecommunications Access Method

active monitoring. In SA z/OS automation control file, the acquiring of resource status information by soliciting such information at regular, user-defined intervals. See also passive monitoring

adapter. Hardware card that enables a device, such as a workstation, to communicate with another device, such as a monitor, a printer, or some other I/O device.

adjacent hosts. Systems connected in a peer relationship using adjacent NetView sessions for purposes of monitoring and control.

adjacent NetView. In SA z/OS, the system defined as the communication path between two SA z/OS systems that do not have a direct link. An adjacent NetView is used for message forwarding and as a communication link between two SA z/OS systems. For example, the adjacent NetView is used when sending responses from a focal point to a remote system.

Advanced Communications Function (ACF). A group of IBM licensed programs (principally VTAM, TCAM, NCP, and SSP) that use the concepts of Systems Network Architecture (SNA), including distribution of function and resource sharing.

advanced program-to-program communication (APPC). A set of inter-program communication services that support cooperative transaction processing in a Systems Network Architecture (SNA) network. APPC is the implementation, on a given system, of SNA's logical unit type 6.2.

alert. (1) In SNA, a record sent to a system problem management focal point or to a collection point to communicate the existence of an alert condition. (2) In NetView, a high-priority event that warrants immediate attention. A database record is generated for certain event types that are defined by user-constructed filters.

alert condition. A problem or impending problem for which some or all of the process of problem determination, diagnosis, and resolution is expected to require action at a control point.
alert focal-point system. See NPDA focal point system.

alert threshold. An application or volume service value that determines the level at which SA z/OS changes the associated icon in the graphical interface to the alert color. SA z/OS may also issue an alert. See warning threshold.

AMC. (1) See Automation Manager Configuration. (2) The Auto Msg Classes entry type.

American Standard Code for Information Interchange (ASCII). A standard code used for information exchange among data processing systems, data communication systems, and associated equipment. ASCII uses a coded character set consisting of 7-bit coded characters (8-bit including parity check). The ASCII set consists of control characters and graphic characters. See also Extended Binary Coded Decimal Interchange Code.

APF. See authorized program facility.

API. See application programming interface.

APPC. See advanced program-to-program communication.

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

Application entry. A construct, created with the customization dialogs, used to represent and contain policy for an application.

application group. A named set of applications. An application group is part of an SA z/OS enterprise definition and is used for monitoring purposes.

application program. (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.

application programming interface (API). An interface that allows an application program that is written in a high-level language to use specific data or functions of the operating system or another program.

ApplicationGroup entry. A construct, created with the customization dialogs, used to represent and contain policy for an application group.

ARM. See automatic restart management.

ASCB. Address space control block.

ASCI. See American Standard Code for Information Interchange.

ASF. See automation status file.

authorized program facility (APF). A facility that permits identification of programs that are authorized to use restricted functions.

automated console operations (ACO). The use of an automated procedure to replace or simplify the action that an operator takes from a console in response to system or network events.

automated function. SA z/OS automated functions are automation operators, NetView autotasks that are assigned to perform specific automation functions. However, SA z/OS defines its own synonyms, or automated function names, for the NetView autotasks, and these function names are referred to in the sample policy databases provided by SA z/OS. For example, the automation operator AUTBASE corresponds to the SA z/OS automated function BASEOPER.

automatic restart management (ARM). A z/OS recovery function that improves the availability of specified subsystems and applications by automatically restarting them under certain circumstances. Automatic restart management is a function of the Cross-System Coupling Facility (XCF) component of z/OS.

automatic restart management element name. In MVS 5.2 or later, z/OS automatic restart management requires the specification of a unique sixteen character name for each address space that registers with it. All automatic restart management policy is defined in terms of the element name, including SA z/OS's interface with it.

automation. The automatic initiation of actions in response to detected conditions or events. SA z/OS provides automation for z/OS applications, z/OS components, and remote systems that run z/OS. SA z/OS also provides tools that can be used to develop additional automation.

automation agent. In SA z/OS, the automation function is split up between the automation manager and the automation agents. The observing, reacting and doing parts are located within the NetView address space, and are known as the automation agents. The automation agents are responsible for:
- Recovery processing
- Message processing
- Active monitoring: they propagate status changes to the automation manager

**automation configuration file.** The SA z/OS customization dialogs must be used to build the automation configuration file. It consists of:
- The automation manager configuration file (AMC)
- The NetView automation table (AT)
- The NetView message revision table (MRT)
- The MPFLSTSA member

**automation control file (ACF).** In SA z/OS, a file that contains system-level automation policy information. There is one master automation control file for each NetView system that SA z/OS is installed on. Additional policy information and all resource status information is contained in the policy database (PDB). The SA z/OS customization dialogs must be used to build the automation control files. They must not be edited manually.

**automation flags.** In SA z/OS, the automation policy settings that determine the operator functions that are automated for a resource and the times during which automation is active. When SA z/OS is running, automation is controlled by automation flag policy settings and override settings (if any) entered by the operator. Automation flags are set using the customization dialogs.

**automation manager.** In SA z/OS, the automation function is split up between the automation manager and the automation agents. The coordination, decision making and controlling functions are processed by each sysplex's automation manager.

The automation manager contains an image of all of the automated resources within the sysplex. The automation agents feed the automation manager with status information and perform the actions that the automation manager tells them to.

The automation manager provides *sysplex-wide* automation.

**Automation Manager Configuration.** The Automation Manager Configuration file (AMC) contains an image of the automated systems in a sysplex or of a standalone system. See also "automation configuration file."

**Automation NetView.** In SA z/OS the NetView that performs routine operator tasks with command procedures or uses other ways of automating system and network management, issuing automatic responses to messages and management services units.

**automation operator.** NetView automation operators are NetView autotasks that are assigned to perform specific automation functions. See also automated function. NetView automation operators may receive messages and process automation procedures. There are no logged-on users associated with automation operators. Each automation operator is an operating system task and runs concurrently with other NetView tasks. An automation operator could be set up to handle JES2 messages that schedule automation procedures, and an automation statement could route such messages to the automation operator. Similar to operator station task, SA z/OS message monitor tasks and target control tasks are automation operators.

**automation policy.** The policy information governing automation for individual systems. This includes automation for applications, z/OS subsystems, z/OS data sets, and z/OS components.

**automation policy settings.** The automation policy information contained in the automation control file. This information is entered using the customization dialogs. You can display or modify these settings using the customization dialogs.

**automation procedure.** A sequence of commands, packaged as a NetView command list or a command processor written in a high-level language. An automation procedure performs automation functions and runs under NetView.

**automation routines.** In SA z/OS, a set of self-contained automation routines that can be called from the NetView automation table, or from user-written automation procedures.

**automation status file (ASF).** In SA z/OS, a file containing status information for each automated subsystem, component or data set. This information is used by SA z/OS automation when taking action or when determining what action to take. In Release 2 and above of AOC/MVS, status information is also maintained in the operational information base.

**automation table (AT).** See NetView automation table

**autotask.** A NetView automation task that receives messages and processes automation procedures. There are no logged-on users associated with autotasks. Each autotask is an operating system task and runs concurrently with other NetView tasks. An autotask could be set up to handle JES2 messages that schedule automation procedures, and an automation statement could route such messages to the autotasks. Similar to operator station task, SA z/OS message monitor tasks and target control tasks are autotasks. Also called automation operator.

**available.** In VTAM programs, pertaining to a logical unit that is active, connected, enabled, and not at its session limit.

**B**

**Base Control Program (BCP).** A program that provides essential services for the MVS and z/OS operating systems. The program includes functions that
manage system resources. These functions include input/output, dispatch units of work, and the z/OS UNIX System Services kernel. See also Multiple Virtual Storage and z/OS.

**basic mode.** A central processor mode that does not use logical partitioning. Contrast with logically partitioned mode.

**BCP.** See Base Control Program.

**BCP Internal Interface.** Processor function of CMOS-390 and System z® processor families. It allows for communication between basic control programs such as z/OS and the processor support element in order to exchange information or to perform processor control functions. Programs using this function can perform hardware operations such as ACTIVATE or SYSTEM RESET.

**beaconing.** The repeated transmission of a frame or messages (beacon) by a console or workstation upon detection of a line break or outage.

**blade.** A hardware unit that provides application-specific services and components. The consistent size and shape (or form factor) of each blade allows it to fit in a BladeCenter chassis.

**BladeCenter chassis.** A modular chassis that can contain multiple blades, allowing the individual blades to share resources such as management, switch, power, and blower modules.

**BookManager®.** An IBM product that lets users view softcopy documents on their workstations.

**C**

**central processor (CP).** The part of the computer that contains the sequencing and processing facilities for instruction execution, initial program load (IPL), and other machine operations.

**central processor complex (CPC).** A physical collection of hardware that consists of central storage, one or more central processors, 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.

**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.

**CHPID.** In SA z/OS, channel path ID; the address of a channel.

**CHPID port.** A label that describes the system name, logical partitions, and channel paths.

**CI.** See console integration.

**CICS/VS.** Customer Information Control System for Virtual Storage. See Customer Information Control System.

**CLIST.** See command list.

**clone.** A set of definitions for application instances that are derived from a basic application definition by substituting a number of different system-specific values into the basic definition.

**clone ID.** A generic means of handling system-specific values such as the MVS SYSCONigure or the VTAM subarea number. Clone IDs can be substituted into application definitions and commands to customize a basic application definition for the system that it is to be instantiated on.

**CNC.** A channel path that transfers data between a host system image and an ESCON® control unit. It can be point-to-point or switchable.

**command.** A request for the performance of an operation or the execution of a particular program.

**command facility.** The component of NetView that is a base for command processors that can monitor, control, automate, and improve the operation of a network. The successor to NCCF.

**command list (CLIST).** (1) A list of commands and statements, written in the NetView command list language or the REXX language, designed to perform a specific function for the user. In its simplest form, a command list is a list of commands. More complex command lists incorporate variable substitution and conditional logic, making the command list more like a conventional program. Command lists are typically interpreted rather than being compiled. (2) In SA z/OS, REXX command lists that can be used for automation procedures.

**command procedure.** In NetView, either a command list or a command processor.
command processor. A module designed to perform a specific function. Command processors, which can be written in assembler or a high-level language (HLL), are issued as commands.

command processor control block. An I/O operations internal control block that contains information about the command being processed.

Command Tree/2. An OS/2-based program that helps you build commands on an OS/2 window, then routes the commands to the destination you specify (such as a 3270 session, a file, a command line, or an application program). It provides the capability for operators to build commands and route them to a specified destination.

common commands. The SA z/OS subset of the CPC operations management commands.

Common User Access (CUA) architecture. Guidelines for the dialog between a human and a workstation or terminal.

communication controller. A type of communication control unit whose operations are controlled by one or more programs stored and executed in the unit or by a program executed in a processor to which the controller is connected. It manages the details of line control and the routing of data through a network.

communication line. Deprecated term for telecommunication line.

connectivity view. In SA z/OS, a display that uses graphic images for I/O devices and lines to show how they are connected.

console automation. The process of having NetView facilities provide the console input usually handled by the operator.

console connection. In SA z/OS, the 3270 or ASCII (serial) connection between a PS/2 computer and a target system. Through this connection, the workstation appears (to the target system) to be a console.

console integration (CI). A hardware facility that if supported by an operating system, allows operating system messages to be transferred through an internal hardware interface for display on a system console. Conversely, it allows operating system commands entered at a system console to be transferred through an internal hardware interface to the operating system for processing.

consoles. Workstations and 3270-type devices that manage your enterprise.

Control units. Hardware units that control I/O operations for one or more devices. You can view information about control units through I/O operations, and can start or stop data going to them by blocking and unblocking ports.

controller. A unit that controls I/O operations for one or more devices.

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

couple data set. A data set that is created through the XCF couple data set format utility and, depending on its designated type, is shared by some or all of the z/OS systems in a sysplex. See also sysplex couple data set and XCF couple data set.

coupling facility. The hardware element that provides high-speed caching, list processing, and locking functions in a sysplex.

CP. See central processor.

CPC. See central processor complex.

CPC operations management commands. A set of commands and responses for controlling the operation of System/390® CPCs.

CPC subset. All or part of a CPC. It contains the minimum resource to support a single control program.

CPCB. See command processor control block.

CPU. Central processing unit. Deprecated term for processor.

cross-system coupling facility (XCF). A component of z/OS that provides functions to support cooperation between authorized programs running within a sysplex.

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

Customer Information Control System (CICS). A general-purpose transactional program that controls online communication between terminal users and a database for a large number of end users on a real-time basis.

customization dialogs. The customization dialogs are an ISPF application. They are used to customize the enterprise policy, like, for example, the enterprise resources and the relationships between resources, or the automation policy for systems in the enterprise. How to use these dialogs is described in IBM Tivoli System Automation for z/OS Customizing and Programming.
CVC. See converted mode.

DataPower X150z. See IBM Websphere DataPower Integration Appliance X150 for zEnterprise (DataPower X150z).

DASD. See direct access storage device.

data services task (DST). The NetView subtask that gathers, records, and manages data in a VSAM file or a network device that contains network management information.

data set. The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

data set members. Members of partitioned data sets that are individually named elements of a larger file that can be retrieved by name.

DBCS. See double-byte character set.

DCCF. See disabled console communication facility.

DCF. See Document Composition Facility.

DELAY Report. An RMF report that shows the activity of each job in the system and the hardware and software resources that are delaying each job.

device. A piece of equipment. Devices can be workstations, printers, disk drives, tape units, remote systems or communications controllers. You can see information about all devices attached to a particular switch, and control paths and jobs to devices.

DEVR Report. An RMF report that presents information about the activity of I/O devices that are delaying jobs.

dialog. Interactive 3270 panels.

direct access storage device (DASD). A device that allows storage to be directly accessed, such as a disk drive.

disabled console communication facility (DCCF). A z/OS component that provides limited-function console communication during system recovery situations.

disk operating system (DOS). (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).

display. (1) To present information for viewing, usually on the screen of a workstation or on a hardcopy device. (2) Deprecated term for panel.

distribution manager. The component of the NetView program that enables the host system to use, send, and delete files and programs in a network of computers.

Document Composition Facility (DCF). An IBM licensed program used to format input to a printer.

domain. (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 with activation requests and deactivation requests.

double-byte character set (DBCS). A character set, such as Kanji, in which each character is represented by a 2-byte code.

DP enterprise. Data processing enterprise.

DSIPARM. This file is a collection of members of NetView’s customization.

DST. Data Services Task.

E

EBCDIC. See Extended Binary Coded Decimal Interchange Code.

ECB. See event control block.

EMCS. Extended multiple console support. See also multiple console support.

ensemble. A collection of one or more zEnterprise nodes (including any attached zBX) that are managed as a single logical virtualized system by the Unified Resource Manager, through the Hardware Management Console.

ensemble member. A zEnterprise node that has been added to an ensemble.

enterprise. The composite of all operational entities, functions, and resources that form the total business concern and that require an information system.

enterprise monitoring. Enterprise monitoring is used by SA z/OS to update the NetView Management Console (NMC) resource status information that is stored in the Resource Object Data Manager (RODM). Resource status information is acquired by enterprise monitoring of the Resource Measurement Facility (RMF) Monitor III service information at user-defined intervals. SA z/OS stores this information in its operational information base, where it is used to update the information presented to the operator in graphic displays.
Enterprise Systems Architecture (ESA). A hardware architecture that reduces the effort required for managing data sets and extends addressability for system, subsystem, and application functions.

**entries.** Resources, such as processors, entered on panels.

**entry type.** Resources, such as processors or applications, used for automation and monitoring.

**environment.** Data processing enterprise.

**error threshold.** An automation policy setting that specifies when SA z/OS should stop trying to restart or recover an application, subsystem or component, or offload a data set.

**ESA.** See Enterprise Systems Architecture.

**eServer.** Processor family group designator used by the SA z/OS customization dialogs to define a target hardware as member of the System z or 390-CMOS processor families.

**event.** (1) In NetView, a record indicating irregularities of operation in physical elements of a network. (2) An occurrence of significance to a task; for example, the completion of an asynchronous operation, such as an input/output operation. (3) Events are part of a trigger condition, such that if all events of a trigger condition have occurred, a startup or shutdown of an application is performed.

**event control block (ECB).** A control block used to represent the status of an event.

**exception condition.** An occurrence on a system that is a deviation from normal operation. SA z/OS monitoring highlights exception conditions and allows an SA z/OS enterprise to be managed by exception.

**Extended Binary Coded Decimal Interchange Code (EBCDIC).** A coded character set of 256 8-bit characters developed for the representation of textual data. See also American Standard Code for Information Interchange.

**extended recovery facility (XRF).** A facility that minimizes the effect of failures in z/OS, VTAM, the host processor, or high availability applications during sessions between high availability applications and designated terminals. This facility provides an alternate subsystem to take over sessions from the failing subsystem.

**file manager commands.** A set of SA z/OS commands that read data from or write data to the automation control file or the operational information base. These commands are useful in the development of automation that uses SA z/OS facilities.

**focal point.** In NetView, the focal-point domain is the central host domain. It is the central control point for any management services element containing control of the network management data.

**focal point system.** (1) A system that can administer, manage, or control one or more target systems. There are a number of different focal point system associated with IBM automation products. (2) NMC focal point system. The NMC focal point system is a NetView system with an attached workstation server and LAN that gathers information about the state of the network. This focal point system uses RODM to store the data it collects in the data model. The information stored in RODM can be accessed from any LAN-connected workstation with NetView Management Console installed. (3) NPDA focal point system. This is a NetView system that collects all the NPDA alerts that are generated within your enterprise. It is supported by NetView. If you have SA z/OS installed the NPDA focal point system must be the same as your NMC focal point system. The NPDA focal point system is also known as the alert focal point system. (4) SA z/OS Processor Operations focal point system. This is a NetView system that has SA z/OS host code installed. The SA z/OS Processor Operations focal point system receives messages from the systems and operator consoles of the machines that it controls. It provides full systems and operations console function for its target systems. It can be used to IPL these systems. Note that some restrictions apply to the Hardware Management Console for an S/390 microprocessor cluster. (5) SA z/OS SDF focal point system. The SA z/OS SDF focal point system is an SA z/OS NetView system that collects status information from other SA z/OS NetViews within your enterprise. (6) Status focal point system. In NetView, the system to which STATMON, VTAM and NLDN send status information on network resources. If you have a NMC focal point, it must be on the same system as the Status focal point. (7) Hardware Management Console. Although not listed as a focal point, the Hardware Management Console acts as a focal point for the console functions of an S/390 microprocessor cluster. Unlike all the other focal points in this definition, the Hardware Management Console runs on a LAN-connected workstation.

**frame.** For a System/390 microprocessor cluster, a frame contains one or two central processor complexes (CPCs), support elements, and AC power distribution.

**full-screen mode.** In NetView, a form of panel presentation that makes it possible to display the contents of an entire workstation screen at once.
Full-screen mode can be used for fill-in-the-blanks prompting. Contrast with line mode.

**Gateway session.** An NetView-NetView Task session with another system in which the SA z/OS outbound gateway operator logs onto the other NetView session without human operator intervention. Each end of a gateway session has both an inbound and outbound gateway operator.

**Generic alert.** Encoded alert information that uses code points (defined by IBM and possibly customized by users or application programs) stored at an alert receiver, such as NetView.

**Group.** A collection of target systems defined through configuration dialogs. An installation might set up a group to refer to a physical site or an organizational or application entity.

**Group entry.** A construct, created with the customization dialogs, used to represent and contain policy for a group.

**Group entry type.** A collection of target systems defined through the customization dialog. An installation might set up a group to refer to a physical site or an organizational entity. Groups can, for example, be of type STANDARD or SYSPLEX.

**Hardware Management Console (HMC).** A user interface through which data center personnel configure, control, monitor, and manage System z hardware and software resources. The HMC communicates with each central processor complex (CPC) through the Support Element. On an IBM zEnterprise 196 (z196), using the Unified Resource Manager on the HMCs or Support Elements, personnel can also create and manage an ensemble.

**Hardware Management Console Application (HWMCA).** A direct-manipulation object-oriented graphical user interface that provides a single point of control and single system image for hardware elements. The HWMCA provides grouping support, aggregated and real-time system status using colors, consolidated hardware messages support, consolidated operating system messages support, consolidated service support, and hardware commands targeted at a single system, multiple systems, or a group of systems.

**Heartbeat.** In SA z/OS, a function that monitors the validity of the status forwarding path between remote systems and the NMC focal point, and monitors the availability of remote z/OS systems, to ensure that status information displayed on the SA z/OS workstation is current.

**Help panel.** An online panel that tells you how to use a command or another aspect of a product.

**Hierarchy.** In the NetView program, the resource types, display types, and data types that make up the organization, or levels, in a network.

**High-level language (HLL).** A programming language that provides some level of abstraction from assembler language and independence from a particular type of machine. For the NetView program, the high-level languages are PL/I and C.

**HLL.** See high-level language.

**Host (primary processor).** The processor that you enter a command at (also known as the issuing processor).

**Host system.** In a coupled system or distributed system environment, the system on which the facilities for centralized automation run. SA z/OS publications refer to target systems or focal-point systems instead of hosts.

**HWMCA.** See Hardware Management Console Application.

**Hypervisor.** A program that allows multiple instances of operating systems or virtual servers to run simultaneously on the same hardware device. A hypervisor can run directly on the hardware, can run within an operating system, or can be imbedded in platform firmware. Examples of hypervisors include PR/SM, z/VM, and PowerVM Enterprise Edition.

**IBM blade.** A customer-acquired, customer-installed select blade to be managed by IBM zEnterprise Unified Resource Manager. One example of an IBM blade is a POWER7 blade.

**IBM Smart Analyzer for DB2 for z/OS.** An optimizer that processes certain types of data warehouse queries for DB2 for z/OS.

**IBM System z Application Assist Processor (zAAP).** A specialized processor that provides a Java execution environment, which enables Java-based web applications to be integrated with core z/OS business applications and backend database systems.

**IBM System z Integrated Information Processor (zIIP).** A specialized processor that provides computing capacity for selected data and transaction processing workloads and for selected network encryption workloads.

**IBM Websphere DataPower Integration Appliance X150 for zEnterprise (DataPower X150z).** A purpose-built appliance that simplifies, helps secure, and optimizes XML and Web services processing.
IBM Enterprise 196 (z196). The newest generation of System z family of servers built on a new processor chip, with enhanced memory function and capacity, security, and on demand enhancements to support existing mainframe workloads and large scale consolidation.

IBM zEnterprise BladeCenter Extension (zBX). A heterogeneous hardware infrastructure that consists of a BladeCenter chassis attached to an IBM zEnterprise 196 (z196). A BladeCenter chassis can contain IBM blades or optimizers.

IBM zEnterprise BladeCenter Extension (zBX) blade. Generic name for all blade types supported in an IBM zEnterprise BladeCenter Extension (zBX). This term includes IBM blades and optimizers.

IBM zEnterprise System (zEnterprise). A heterogeneous hardware infrastructure that can consist of an IBM zEnterprise 196 (z196) and an attached IBM zEnterprise BladeCenter Extension (zBX) Model 002, managed as a single logical virtualized system by the Unified Resource Manager.

IBM zEnterprise Unified Resource Manager. Licensed Internal Code (LIC), also known as firmware, that is part of the Hardware Management Console. The Unified Resource Manager provides energy monitoring and management, goal-oriented policy management, increased security, virtual networking, and data management for the physical and logical resources of a given ensemble.

I/O operations. The part of SA z/OS that provides you with a single point of logical control for managing connectivity in your active I/O configurations. I/O operations takes an active role in detecting unusual conditions and lets you view and change paths between a processor and an I/O device, using dynamic switching (the ESCON director). Also known as I/O Ops.

I/O Ops. See I/O operations.

I/O resource number. Combination of channel path identifier (CHPID), device number, etc. See internal token.

images. A grouping of processors and I/O devices that you define. You can define a single-image mode that allows a multiprocessor system to function as one central processor image.

IMS. See Information Management System.

IMS/VS. See Information Management System/Virtual Storage.

inbound. In SA z/OS, messages sent to the focal-point system from the PC or target system.

inbound gateway operator. The automation operator that receives incoming messages, commands, and responses from the outbound gateway operator at the sending system. The inbound gateway operator handles communications with other systems using a gateway session.

Information Management System (IMS). Any of several system environments available with a database manager and transaction processing that are capable of managing complex databases and terminal networks.

Information Management System/Virtual Storage (IMS/VS). A database/data communication (DB/DC) system that can manage complex databases and networks. Synonymous with Information Management System.

INGEIO PROC. The I/O operations default procedure name. It is part of the SYSI.PROCLIB.

initial microprogram load. The action of loading microprograms into computer storage.

initial program load (IPL). (1) The initialization procedure that causes an operating system to commence operation. (2) The process by which a configuration image is loaded into storage at the beginning of a workday or after a system malfunction. (3) The process of loading system programs and preparing a system to run jobs.

initialize automation. SA z/OS-provided automation that issues the correct z/OS start command for each subsystem when SA z/OS is initialized. The automation ensures that subsystems are started in the order specified in the automation control files and that prerequisite applications are functional.

input/output configuration data set (IOCDS). A configuration definition built by the I/O configuration program (IOCP) and stored on disk files associated with the processor controller.

input/output support processor (IORSP). The hardware unit that provides I/O support functions for the primary support processor and maintenance support functions for the processor controller.

Interactive System Productivity Facility (ISPF). An IBM licensed program that serves as a full-screen editor and dialog manager. Used for writing application programs, it provides a means of generating standard screen panels and interactive dialogs between the application programmer and the terminal user. See also Time Sharing Option.

interested operator list. The list of operators who are to receive messages from a specific target system.

internal token. A logical token (LTOK); name by which the I/O resource or object is known; stored in IODE.
IOCDS. See input/output configuration data set

IOSP. See input/output support processor

IPL. See initial program load

ISPF. See Interactive System Productivity Facility

ISPF console. You log on to ISPF from this 3270-type console to use the runtime panels for I/O operations and SA z/OS customization panels.

issuing host. The base program that you enter a command for processing with. See primary host

J

JCL. See job control language

JES. See job entry subsystem

JES2. An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system. In an installation with more than one processor, each JES2 processor independently controls its job input, scheduling, and output processing. See job entry subsystem and JES3

JES3. An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system. In complexes that have several loosely coupled processing units, the JES3 program manages processors so that the global processor exercises centralized control over the local processors and distributes jobs to them using a common job queue. See job entry subsystem and JES2

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). An IBM licensed program that receives jobs into the system and processes all output data that is produced by jobs. In SA z/OS publications, JES refers to JES2 or JES3, unless otherwise stated. See also JES2 and JES3

K

Kanji. An ideographic character set used in Japanese. See also double-byte character set

L

LAN. See local area network

line mode. A form of screen presentation in which the information is presented a line at a time in the message area of the terminal screen. Contrast with full-screen mode

link. (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). See synchronous data link control. (2) In SA z/OS, link connection is the physical medium of transmission.

link-attached. Describes devices that are physically connected by a telecommunication line. Contrast with channel-attached

Linux on System z. UNIX-like open source operating system conceived by Linus Torvalds and developed across the internet.

local. Pertaining to a device accessed directly without use of a telecommunication line. Synonymous with channel-attached

local area network (LAN). (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 that connects several devices in a limited area (such as a single building or campus) and that can be connected to a larger network.

logical partition (LP). A subset of the processor hardware that is defined to support an operating system. See also logically partitioned mode

logical switch number (LSN). Assigned with the switch parameter of the CHPID macro of the IOCP.

logical token (LTOK). Resource number of an object in the IODF.

logical unit (LU). In SNA, a port through which an end user accesses the SNA network and the functions provided by system services control points (SSCPs). An LU can support at least two sessions, one with an SSCP and one with another LU, and may be capable of supporting many sessions with other LUs. See also physical unit and system services control point

logical unit 6.2 (LU 6.2). A type of logical unit that supports general communications between programs in a distributed processing environment. LU 6.2 is characterized by:
- A peer relationship between session partners
- Efficient use of a session for multiple transactions
- A comprehensive end-to-end error processing
A generic application program interface (API) consisting of structured verbs that are mapped to a product implementation.

logically partitioned (LPAR) mode. A central processor mode that enables an operator to allocate system processor hardware resources among several logical partitions. Contrast with basic mode.

LOGR. The sysplex logger.

LP. See logical partition.

LPAR. See logically partitioned mode.

LSN. See logical switch number.

LU. See logical unit.

LU 6.2. See logical unit 6.2.

LU 6.2 session. A session initiated by VTAM on behalf of an LU 6.2 application program, or a session initiated by a remote LU in which the application program specifies that VTAM is to control the session by using the APPCCMD macro. See logical unit 6.2.

LU-LU session. In SNA, a session between two logical units (LUs) in an SNA network. It provides communication between two end users, or between an end user and an LU services component.

message forwarding. The SA z/OS process of sending messages generated at an SA z/OS target system to the SA z/OS focal-point system.

message group. Several messages that are displayed together as a unit.

message monitor task. A task that starts and is associated with a number of communications tasks. Message monitor tasks receive inbound messages from a communications task, determine the originating target system, and route the messages to the appropriate target control tasks.

message processing facility (MPF). A z/OS table that screens all messages sent to the z/OS console. The MPF compares these messages with a customer-defined list of messages on which to automate, suppress from the z/OS console display, or both, and marks messages to automate or suppress. Messages are then broadcast on the subsystem interface (SSI).

message suppression. The ability to restrict the amount of message traffic displayed on the z/OS console.

Micro Channel architecture. The rules that define how subsystems and adapters use the Micro Channel bus in a computer. The architecture defines the services that each subsystem can or must provide.

microprocessor. A processor implemented on one or a small number of chips.

migration. Installation of a new version or release of a program to replace an earlier version or release.

MP. Multiprocessor.

MPF. See message processing facility.

MPFLSTSA. The MPFLST member that is built by SA z/OS.

multi-MVS environment. A physical processing system that is capable of operating more than one MVS image. See also MVS image.

multiple console support (MCS). A feature of MVS that permits selective message routing to multiple consoles.

Multiple Virtual Storage (MVS). An IBM operating system that accesses multiple address spaces in virtual storage. The predecessor of z/OS.

multiprocessor (MP). A CPC that can be physically partitioned to form two operating processor complexes.

multisystem application. An application program that has various functions distributed across z/OS images in a multisystem environment.
multisystem environment. An environment in which two or more systems reside on one or more processors. Or one or more processors can communicate with programs on the other systems.

MVS. See Multiple Virtual Storage

MVS image. A single occurrence of the MVS operating system that has the ability to process work. See also multi-MVS environment and single-MVS environment

MVS/ESA. Multiple Virtual Storage/Enterprise Systems Architecture. See z/OS

MVS/JES2. Multiple Virtual Storage/Job Entry System 2. A z/OS subsystem that receives jobs into the system, converts them to an internal format, selects them for execution, processes their output, and purges them from the system. In an installation with more than one processor, each JES2 processor independently controls its job input, scheduling, and output processing.

NAU. (1) See network addressable unit (2) See network accessible unit

NCCF. See Network Communications Control Facility

NCP. (1) See network control program (general term). (2) See Network Control Program (an IBM licensed program). Its full name is Advanced Communications Function for the Network Control Program. Synonymous with ACF/NCP

NCP/token ring interconnection. A function used by ACF/NCP to support token ring-attached SNA devices. NTRI also provides translation from token ring-attached SNA devices (PUs) to switched (dial-up) devices.

NetView. An IBM licensed program used to monitor a network, manage it, and diagnose network problems. NetView consists of a command facility that includes a presentation service, command processors, automation based on command lists, and a transaction processing structure on which the session monitor, hardware monitor, and terminal access facility (TAF) network management applications are built.

NetView (NCCF) console. A 3270-type console for NetView commands and runtime panels for system operations and processor operations.

NetView automation procedures. A sequence of commands, packaged as a NetView command list or a command processor written in a high-level language. An automation procedure performs automation functions and runs under the NetView program.

NetView automation table (AT). A table against which the NetView program compares incoming messages. A match with an entry triggers the specified response. SA z/OS entries in the NetView automation table trigger an SA z/OS response to target system conditions. Formerly known as the message automation table (MAT).

NetView command list language. An interpretive language unique to NetView that is used to write command lists.

NetView Graphic Monitor Facility (NGMF). Deprecated term for NetView Management Console

NetView hardware monitor. The component of NetView that helps identify network problems, such as hardware, software, and microcode, from a central control point using interactive display techniques. Formerly called network problem determination application.

NetView log. The log that NetView records events relating to NetView and SA z/OS activities in.

NetView Management Console (NMC). A function of the NetView program that provides a graphic, topological presentation of a network that is controlled by the NetView program. It provides the operator different views of a network, multiple levels of graphical detail, and dynamic resource status of the network. This function consists of a series of graphic windows that allows you to manage the network interactively. Formerly known as the NetView Graphic Monitor Facility (NGMF).

NetView message table. See NetView automation table

NetView paths via logical unit (LU 6.2). A type of network-accessible port (VTAM connection) that enables end users to gain access to SNA network resources and communicate with each other. LU 6.2 permits communication between processor operations and the workstation. See logical unit 6.2

NetView-NetView task (NNT). The task that a cross-domain NetView operator session runs under. Each NetView program must have a NetView-NetView task to establish one NNT session. See also operator station task

NetView-NetView task session. A session between two NetView programs that runs under a NetView-NetView task. In SA z/OS, NetView-NetView task sessions are used for communication between focal point and remote systems.

network. (1) An interlinked group of nodes. (2) In data processing, a user application network. See SNA network

network accessible unit (NAU). In SNA networking, any device on the network that has a network address, including a logical unit (LU), physical unit (PU), control point (CP), or system services control point (SSCP). It is
the origin or the destination of information transmitted
by the path control network. Synonymous with
network addressable unit

network addressable unit (NAU). Synonym for
network accessible unit

Network Communications Control Facility (NCCF).
The operations control facility for the network. NCCF
consists of a presentation service, command processors,
automation based on command lists, and a transaction
processing structure on which the network
management applications NLDM and NPDA are built.
NCCF is a precursor to the NetView command facility.

Network Control Program (NCP). An IBM licensed
program that provides communication controller
support for single-domain, multiple-domain, and
interconnected network capability. Its full name is
Advanced Communications Function for the Network
Control Program.

network control program (NCP). (1) A program that
controls the operation of a communication controller.
(2) A program used for requests and responses
exchanged between physical units in a network for
data flow control.

Network Problem Determination Application
(NPDA). An NCCF application that helps you identify
network problems, such as hardware, software, and
microcode, from a central control point using
interactive display methods. The alert manager for the
network. The precursor of the NetView hardware
monitor.

Netviewing Netview. In SA z/OS the NetView that
performs network management functions, such as
managing the configuration of a network. In SA z/OS
it is common to also route alerts to the Netviewing
Netview.

NGME. Deprecated term for NetView Management
Console

NGMF focal-point system. Deprecated term for NMC
focal point system

NIP. See nucleus initialization program

NMC focal point system. See focal point system

NMC workstation. The NMC workstation is the
primary way to dynamically monitor SA z/OS
systems. From the windows, you see messages, monitor
status, view trends, and react to changes before they
cause problems for end users. You can use multiple
windows to monitor multiple views of the system.

NNT. See NetView-NetView task

notification message. An SA z/OS message sent to a
human notification operator to provide information
about significant automation actions. Notification
messages are defined using the customization dialogs.

notification operator. A NetView console operator
who is authorized to receive SA z/OS notification
messages. Authorization is made through the
customization dialogs.

NPDA. See Network Problem Determination
Application

NPDA focal-point system. See focal point system

NTRI. See NCP/token ring interconnection

nucleus initialization program (NIP). The program
that initializes the resident control program; it allows
the operator to request last-minute changes to certain
options specified during system generation.

O

objective value. An average Workflow or Using value
that SA z/OS can calculate for applications from past
service data. SA z/OS uses the objective value to
calculate warning and alert thresholds when none are
explicitly defined.

OCA. In SA z/OS, operator console A, the active
operator console for a target system. Contrast with
OCB

OCB. In SA z/OS, operator console B, the backup
operator console for a target system. Contrast with
OCA

OCF. See operations command facility

OCF-based processor. A central processor complex
that uses an operations command facility for interacting
with human operators or external programs to perform
operations management functions on the CPC.

OPC/A. See Operations Planning and
Control/Advanced

OPC/ESA. See Operations Planning and
Control/Enterprise Systems Architecture

Open Systems Adapter (OSA). I/O operations can
display the Open System Adapter (OSA) channel
logical definition, physical attachment, and status. You
can configure an OSA channel on or off.

operating system (OS). Software that controls the
execution of programs and that may provide services
such as resource allocation, scheduling, input/output
control, and data management. Although operating
systems are predominantly software, partial hardware
implementations are possible. (T)

operations. The real-time control of a hardware device
or software function.
operations 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/Enterprise Systems Architecture (OPC/ESA). A set of IBM licensed programs that automate, plan, and control batch workload. OPC/ESA analyzes system and workload status and submits jobs accordingly. The successor to OPC/A.

operator. (1) A person who keeps a system running. (2) A person or program responsible for managing activities controlled by a given piece of software such as z/OS, the NetView program, or IMS. (3) A person who operates a device. (4) In a language statement, the lexical entity that indicates the action to be performed on operands.

operator console. (1) A functional unit containing devices that are used for communications between a computer operator and a computer. (2) A display console used for communication between the operator and the system, used primarily to specify information concerning application programs and I/O operations and to monitor system operation. (3) In SA z/OS, a console that displays output from and sends input to the operating system (z/OS, LINUX, VM, VSE). Also called operating system console. In the SA z/OS operator commands and configuration dialogs, OC is used to designate a target system operator console.

operator station task (OST). The NetView task that establishes and maintains the online session with the network operator. There is one operator station task for each network operator who logs on to the NetView program.

operator view. A set of group, system, and resource definitions that are associated together for monitoring purposes. An operator view appears as a graphic display in the graphical interface showing the status of the defined groups, systems, and resources.

OperatorView entry. A construct, created with the customization dialogs, used to represent and contain policy for an operator view.

optimizer. A special-purpose hardware component or appliance that can perform a limited set of specific functions with optimized performance when compared to a general-purpose processor. Because of its limited set of functions, an optimizer is an integrated part of a processing environment, rather than a stand-alone unit. One example of an optimizer is the IBM Smart Analytics Optimizer for DB2 for z/OS.

OS. See operating system

OSA. See Open Systems Adapter

OST. See operator station task

outbound. In SA z/OS, messages or commands from the focal-point system to the target system.

outbound gateway operator. The automation operator that establishes connections to other systems. The outbound gateway operator handles communications with other systems through a gateway session. The automation operator sends messages, commands, and responses to the inbound gateway operator at the receiving system.

P

page. (1) The portion of a panel that is shown on a display surface at one time. (2) To transfer instructions, data, or both between real storage and external page or auxiliary storage.

panel. (1) A formatted display of information that appears on a terminal screen. Panels are full-screen 3270-type displays with a monospaced font, limited color and graphics. (2) By using SA z/OS panels you can see status, type commands on a command line using a keyboard, configure your system, and passthru to other consoles. See also help panel. (3) In computer graphics, a display image that defines the locations and characteristics of display fields on a display surface. Contrast with screen.

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

parameter. (1) A variable that is given a constant value for a specified application and that may denote the application. (2) An item in a menu for which the user specifies a value or for which the system provides a value when the menu is interpreted. (3) Data passed to a program or procedure by a user or another program, specifically 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.
partitioned data set (PDS). A data set in direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

passive monitoring. In SA z/OS, the receiving of unsolicited messages from z/OS systems and their resources. These messages can prompt updates to resource status displays. See also active monitoring.

PCE. A processor controller. Also known as the support processor or service processor in some processor families.

PDB. See policy database

PDS. See partitioned data set

physical partition. Part of a CPC that operates as a CPC in its own right, with its own copy of the operating system.

physical unit (PU). In SNA, the component that manages and monitors the resources (such as attached links and adjacent link stations) of a node, as requested by a system services control point (SSCP) through an SSCP-PU session. An SSCP activates a session with the physical unit to indirectly manage, through the PU, resources of the node such as attached links.

physically partitioned (PP) configuration. A mode of operation that allows a multiprocessor (MP) system to function as two or more independent CPCs having separate power, water, and maintenance boundaries. Contrast with single-image mode.

POI. See program operator interface

policy. The automation and monitoring specifications for an SA z/OS enterprise. See IBM Tivoli System Automation for z/OS Defining Automation Policy

policy database. The automation definitions (automation policy) that the automation programmer specifies using the customization dialog is stored in the policy database. Also known as the PDB. See also automation policy.

POR. See power-on reset

port. (1) System hardware that the I/O devices are attached to. (2) In an ESCON switch, a port is an addressable connection. The switch routes data through the ports to the channel or control unit. Each port has a name that can be entered into a switch matrix, and you can use commands to change the switch configuration. (3) An access point (for example, a logical unit) for data entry or exit. (4) A functional unit of a node that data can enter or leave a data network through. (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.

power-on reset (POR). A function that re-initializes all the hardware in a CPC and loads the internal code that enables the CPC to load and run an operating system. See initial microprogram load.

PP. See physical partition

PPI. See program to program interface

PPT. See primary POI task

PR/SM. See Processor Resource/Systems Manager

primary host. The base program that you enter a command for processing at.

primary POI task (PPT). The NetView subtask that processes all unsolicited messages received from the VTAM program operator interface (POI) and delivers them to the controlling operator or to the command processor. The PPT also processes the initial command specified to execute when NetView is initialized and timer request commands scheduled to execute under the PPT.

primary system. A system is a primary system for an application if the application is normally meant to be running there. SA z/OS starts the application on all the primary systems defined for it.

problem determination. The process of determining the source of a problem; for example, a program component, machine failure, telecommunication facilities, user or contractor-installed programs or equipment, environment failure such as a power loss, or user error.

processor. (1) A device for processing data from programmed instructions. It may be part of another unit. (2) In a computer, the part that interprets and executes instructions. Two typical components of a processor are a control unit and an arithmetic logic unit.

processor controller. Hardware that provides support and diagnostic functions for the central processors.

processor operations. The part of SA z/OS that monitors and controls processor (hardware) operations. Processor operations provides a connection from a focal-point system to a target system. Through NetView on the focal-point system, processor operations automates operator and system consoles for monitoring and recovering target systems. Also known as ProcOps.

Processor Resource/Systems Manager (PR/SM). The feature that allows the processor to use several operating system images simultaneously and provides logical partitioning capability. See also logically partitioned mode.

ProcOps. See processor operations
ProcOps Service Machine (PSM). The PSM is a CMS user on a VM host system. It runs a CMS multitasking application that serves as “virtual hardware” for ProcOps. ProcOps communicates via the PSM with the VM guest systems that are defined as target systems within ProcOps.

Product Automation. Automation integrated into the base of z/OS for the products CICS, DB2, IMS, TWS (formerly called features).

Program Operator Interface (POI). A NetView facility for receiving VTAM messages.

Program to Program Interface (PPI). A NetView function that allows user programs to send or receive data buffers from other user programs and to send alerts to the NetView hardware monitor from system and application programs.

Protocol. In SNA, the meanings of, and the sequencing rules for, requests and responses used for managing the network, transferring data, and synchronizing the states of network components.

Proxy Resource. A resource defined like an entry type APL representing a processor operations target system.

PSM. See ProcOps Service Machine.

PU. See physical unit.

R

RACF. See Resource Access Control Facility.

Remote System. A system that receives resource status information from an SA z/OS focal-point system. An SA z/OS remote system is defined as part of the same SA z/OS enterprise as the SA z/OS focal-point system to which it is related.

Requester. A workstation from that user can log on to a domain from, that is, to the servers belonging to the domain, and use network resources. Users can access the shared resources and use the processing capability of the servers, thus reducing hardware investment.

Resource. (1) Any facility of the computing system or operating system required by a job or task, and including main storage, input/output devices, the processing unit, data sets, and control or processing programs. (2) In NetView, any hardware or software that provides function to the network. (3) In SA z/OS, any z/OS application, z/OS component, job, device, or target system capable of being monitored or automated through SA z/OS.

Resource Access Control Facility (RACF). A program that can provide data security for all your resources. RACF protects data from accidental or deliberate unauthorized disclosure, modification, or destruction.

Resource Group. A physically partitionable portion of a processor. Also known as a side.

Resource Measurement Facility (RMF). A feature of z/OS that measures selected areas of system activity and presents the data collected in the format of printed reports, System Management Facility (SMF) records, or display reports.

Resource Object Data Manager (RODM). In NetView for z/OS, a component that provides an in-memory cache for maintaining real-time data in an address space that is accessible by multiple applications. RODM also allows an application to query an object and receive a rapid response and act on it.

Resource Token. A unique internal identifier of an ESCON resource or resource number of the object in the IODF.

Restart Automation. Automation provided by SA z/OS that monitors subsystems to ensure that they are running. If a subsystem fails, SA z/OS attempts to restart it according to the policy in the automation configuration file.

Restructured Extended Executor (REXX). A general-purpose, high-level, programming language, particularly suitable for EXEC procedures or programs for personal computing, used to write command lists.

Return Code. A code returned from a program used to influence the issuing of subsequent instructions.

REXX. See Restructured Extended Executor.

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

RMF. See Resource Measurement Facility.

RODM. See Resource Object Data Manager.

S


SAPlex. SAplex or "SA z/OS Subplex" is a term used in conjunction with a sysplex. In fact, a SAplex is a subset of a sysplex. However, it can also be a sysplex. For a detailed description, refer to "Using SA z/OS Subplexes" in IBM Tool: System Automation for z/OS Planning and Installation.

SA z/OS. See System Automation for z/OS.

SA z/OS Customization Dialogs. An ISPF application through which the SA z/OS policy administrator
defines policy for individual z/OS systems and builds automation control data and RODM load function files.

**SA z/OS customization focal point system.** See [focal point system](#).

**SA z/OS data model.** The set of objects, classes and entity relationships necessary to support the function of SA z/OS and the NetView automation platform.

**SA z/OS enterprise.** The group of systems and resources defined in the customization dialogs under one enterprise name. An SA z/OS enterprise consists of connected z/OS systems running SA z/OS.

**SA z/OS focal point system.** See [focal point system](#).

**SA z/OS policy.** The description of the systems and resources that make up an SA z/OS enterprise, together with their monitoring and automation definitions.

**SA z/OS policy administrator.** The member of the operations staff who is responsible for defining SA z/OS policy.

**SA z/OS satellite.** If you are running two NetViews on an z/OS system to split the automation and networking functions of NetView, it is common to route alerts to the Networking NetView. For SA z/OS to process alerts properly on the Networking NetView, you must install a subset of SA z/OS code, called an SA z/OS satellite on the Networking NetView.

**SA z/OS SDF focal point system.** See [focal point system](#).

**SCA.** In SA z/OS, system console A, the active system console for a target hardware. Contrast with **SCB**.

**SCB.** In SA z/OS, system console B, the backup system console for a target hardware. Contrast with **SCA**.

**screen.** Deprecated term for [panel](#).

**screen handler.** In SA z/OS, software that interprets all data to and from a full-screen image of a target system. The interpretation depends on the format of the data on the full-screen image. Every processor and operating system has its own format for the full-screen image. A screen handler controls one PS/2 connection to a target system.

**SDF.** See [status display facility](#).

**SDLC.** See [synchronous data link control](#).

**SDSF.** See [System Display and Search Facility](#).

**secondary system.** A system is a secondary system for an application if it is defined to automation on that system, but the application is not normally meant to be running there. Secondary systems are systems to which an application can be moved in the event that one or more of its primary systems are unavailable. SA z/OS does not start the application on its secondary systems.

**Security Authorization Facility (SAF).** An MVS interface with which programs can communicate with an external security manager, such as RACF.

**server.** A server is a workstation that shares resources, which include directories, printers, serial devices, and computing powers.

**service language command (SLC).** The line-oriented command language of processor controllers or service processors.

**service period.** Service periods allow the users to schedule the availability of applications. A service period is a set of time intervals (service windows), during which an application should be active.

**service processor (SVP).** The name given to a processor controller on smaller System/370 processors.

**service threshold.** An SA z/OS policy setting that determines when to notify the operator of deteriorating service for a resource. See also [alert threshold](#) and [warning threshold](#).

**session.** In SNA, a logical connection between two network addressable units (NAUs) that can be activated, tailored to provide various protocols, and deactivated, as requested. Each session is uniquely identified in a transmission header by a pair of network addresses identifying the origin and destination NAUs of any transmissions exchanged during the session.

**session monitor.** The component of the NetView program that collects and correlates session-related data and provides online access to this information. The successor to NLDM.

**shutdown automation.** SA z/OS-provided automation that manages the shutdown process for subsystems by issuing shutdown commands and responding to prompts for additional information.

**side.** A part of a partitionable CPC that can run as a physical partition and is typically referred to as the A-side or the B-side.

**Simple Network Management Protocol (SNMP).** A set of protocols for monitoring systems and devices in complex networks. Information about managed devices is defined and stored in a Management Information Base (MIB).

**single image.** A processor system capable of being physically partitioned that has not been physically partitioned. Single-image systems can be target hardware processors.
single-MVS environment. An environment that supports one MVS image. See also \[MVS image\]

single-image (SI) mode. A mode of operation for a multiprocessor (MP) system that allows it to function as one CPC. By definition, a uniprocessor (UP) operates in single-image mode. Contrast with \[physically partitioned (PP) configuration\]

SLC. See \[service language command\]

SMP/E. See \[System Modification Program/Extended\]

SNA. See \[Systems Network Architecture\]

SNA network. In SNA, the part of a user-application network that conforms to the formats and protocols of systems network architecture. It enables reliable transfer of data among end users and provides protocols for controlling the resources of various network configurations. The SNA network consists of network addressable units (NAUs), boundary function components, and the path control network.

SNMP. See \[Simple Network Management Protocol\]

solicited message. An SA \(z/OS\) message that directly responds to a command. Contrast with \[unsolicited message\]

SSCP. See \[system services control point\]

SSI. See \[subsystem interface\]

start automation. \(z/OS\)-provided automation that manages and completes the startup process for subsystems. During this process, \(z/OS\) replies to prompts for additional information, ensures that the startup process completes within specified time limits, notifies the operator of problems, if necessary, and brings subsystems to an UP (or ready) state.

startup. The point in time that a subsystem or application is started.

status. The measure of the condition or availability of the resource.

status display facility (SDF). The system operations part of \(z/OS\) that displays status of resources such as applications, gateways, and write-to-operator messages (WTORs) on dynamic color-coded panels. SDF shows spool usage problems and resource data from multiple systems.

status focal-point system. See \[local point system\]

steady state automation. The routine monitoring, both for presence and performance, of subsystems, applications, volumes and systems. Steady state automation may respond to messages, performance exceptions and discrepancies between its model of the system and reality.

structure. A construct used by \(z/OS\) to map and manage storage on a coupling facility.

subgroup. A named set of systems. A subgroup is part of an SA \(z/OS\) enterprise definition and is used for monitoring purposes.

SubGroup entry. A construct, created with the customization dialogs, used to represent and contain policy for a subgroup.

subplex. See \[SAplex\]

subsystem. (1) A secondary or subordinate system, usually capable of operating independent of, or asynchronously with, a controlling system. (2) In \(z/OS\), an \(z/OS\) application or subsystem defined to \(z/OS\).

subsystem interface (SSI). The \(z/OS\) interface over which all messages sent to the \(z/OS\) console are broadcast.

support element. A hardware unit that provides communications, monitoring, and diagnostic functions to a central processor complex (CPC).

support processor. Another name given to a processor controller on smaller System/370 processors. See \[service processor\]

SVP. See \[service processor\]

switch identifier. The switch device number (swchdevn), the logical switch number (LSN) and the switch name

switches. ESCON directors are electronic units with ports that dynamically switch to route data to I/O devices. The switches are controlled by I/O operations commands that you enter on a workstation.

symbolic destination name (SDN). Used locally at the workstation to relate to the VTAM application name.

synchronous data link control (SDLC). A discipline for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or nonswitched links. The configuration of the link connection may be point-to-point, multipoint, or loop. SDLC conforms to subsets of the Advanced Data Communication Control Procedures (ADCCP) of the American National Standards Institute and High-Level Data Link Control (HDLC) of the International Standards Organization.

SYSINFO Report. An RMF report that presents an overview of the system, its workload, and the total number of jobs using resources or delayed for resources.

SysOps. See \[system operations\]
sysplex. A set of z/OS systems communicating and cooperating with each other through certain multisystem hardware components (coupling devices and timers) and software services (couple data sets). In a sysplex, z/OS provides the coupling services that handle the messages, data, and status for the parts of a multisystem application that has its workload spread across two or more of the connected processors, sysplex timers, coupling facilities, and couple data sets (which contains policy and states for automation).

A Parallel Sysplex® is a sysplex that includes a coupling facility.

sysplex application group. A sysplex application group is a grouping of applications that can run on any system in a sysplex.

sysplex couple data set. A couple data set that contains sysplex-wide data about systems, groups, and members that use XCF services. All z/OS systems in a sysplex must have connectivity to the sysplex couple data set. See also couple data set.

Sysplex Timer. An IBM unit that synchronizes the time-of-day (TOD) clocks in multiple processors or processor sides. External Time Reference (ETR) is the z/OS generic name for the IBM Sysplex Timer (9037).

system. In SA z/OS, system means a focal point system (z/OS) or a target system (MVS, VM, VSE, LINUX, or CF).

System Automation for Integrated Operations Management. (1) An outboard automation solution for secure remote access to mainframe/distributed systems. Tivoli System Automation for Integrated Operations Management, previously Tivoli AF/REMOTE, allows users to manage mainframe and distributed systems from any location. (2) The full name for SA IOM.

System Automation for OS/390. The full name for SA OS/390, the predecessor to System Automation for z/OS.

System Automation for z/OS. The full name for SA z/OS.

system console. (1) A console, usually having a keyboard and a display screen, that is used by an operator to control and communicate with a system. (2) A logical device used for the operation and control of hardware functions (for example, IPL, ALTER/DISPLAY, and reconfiguration). The system console can be assigned to any of the physical displays attached to a processor controller or support processor. (3) In SA z/OS, the hardware system console for processor controllers or service processors of processors connected using SA z/OS. In the SA z/OS operator commands and configuration dialogs, SC is used to designate the system console for a target hardware processor.

System Display and Search Facility (SDSF). An IBM licensed program that provides information about jobs, queues, and printers running under JES2 on a series of panels. Under SA z/OS you can select SDSF from a pull-down menu to see the resources’ status, view the z/OS system log, see WTOR messages, and see active jobs on the system.

System entry. A construct, created with the customization dialogs, used to represent and contain policy for a system.

System Modification Program/Extended (SMP/E). An IBM licensed program that facilitates the process of installing and servicing an z/OS system.

system operations. The part of SA z/OS that monitors and controls system operations applications and subsystems such as NetView, SDSF, JES, RMF, TSO, RODM, ACF/VTAM, CICS, IMS, and OPC. Also known as SysOps.

system services control point (SSCP). In SNA, the focal point within an SNA network for managing the configuration, coordinating network operator and problem determination requests, and providing directory support and other session services for end users of the network. Multiple SSCP’s, cooperating as peers, can divide the network into domains of control, with each SSCP having a hierarchical control relationship to the physical units and logical units within its domain.

System/390 microprocessor cluster. A configuration that consists of central processor complexes (CPCs) and may have one or more integrated coupling facilities.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks.

T

TAF. See terminal access facility.

target. A processor or system monitored and controlled by a focal-point system.

target control task. In SA z/OS, target control tasks process commands and send data to target systems and workstations through communications tasks. A target control task (a NetView autotask) is assigned to a target system when the target system is initialized.

target hardware. In SA z/OS, the physical hardware on which a target system runs. It can be a single-image or physically partitioned processor. Contrast with target system.
**target system.** (1) In a distributed system environment, a system that is monitored and controlled by the focal-point system. Multiple target systems can be controlled by a single focal-point system. (2) In SA z/OS, a computer system attached to the focal-point system for monitoring and control. The definition of a target system includes how remote sessions are established, what hardware is used, and what operating system is used.

**task.** (1) A basic unit of work to be accomplished by a computer. (2) In the NetView environment, an operator station task (logged-on operator), automation operator (autotask), application task, or user task. A NetView task performs work in the NetView environment. All SA z/OS tasks are NetView tasks. See also message monitor task and target control task.

**telecommunication line.** Any physical medium, such as a wire or microwave beam, that is used to transmit data.

**terminal access facility (TAF).** (1) A NetView function that allows you to log onto multiple applications either on your system or other systems. You can define TAF sessions in the SA z/OS customization panels so you don’t have to set them up each time you want to use them. (2) In NetView, a facility that allows a network operator to control a number of subsystems. In a full-screen or operator control session, operators can control any combination of subsystems simultaneously.

**terminal emulation.** The capability of a microcomputer or personal computer to operate as if it were a particular type of terminal linked to a processing unit to access data.

**threshold.** A value that determines the point at which SA z/OS automation performs a predefined action. See alert threshold, warning threshold, and error threshold.

**time of day (TOD).** Typically refers to the time-of-day clock.

**Time Sharing Option (TSO).** An optional configuration of the operating system that provides conversational time sharing from remote stations. It is an interactive service on z/OS, MVS/ESA, and MVS/XA.

**Time-Sharing Option/Extended (TSO/E).** An option of z/OS that provides conversational timesharing from remote terminals. TSO/E allows a wide variety of users to perform many different kinds of tasks. It can handle short-running applications that use fewer sources as well as long-running applications that require large amounts of resources.

**times.** A NetView command that issues a command or command processor (list of commands) at a specified time or time interval.

**Tivoli Workload Scheduler (TWS).** A family of IBM licensed products that plan, execute and track jobs on several platforms and environments. The successor to OPC/A.

**TOD.** Time of day.

**token ring.** A network with a ring topology that passes tokens from one attaching device to another; for example, the IBM Token-Ring Network product.

**TP.** See transaction program.

**transaction program.** In the VTAM program, a program that performs services related to the processing of a transaction. One or more transaction programs may operate within a VTAM application program that is using the VTAM application program interface (API). In that situation, the transaction program would request services from the applications program using protocols defined by that application program. The application program, in turn, could request services from the VTAM program by issuing the APPCCMD macro instruction.

**transitional automation.** The actions involved in starting and stopping subsystems and applications that have been defined to SA z/OS. This can include issuing commands and responding to messages.

**translating host.** Role played by a host that turns a resource number into a token during a unification process.

**trigger.** Triggers, in combination with events and service periods, are used to control the starting and stopping of applications in a single system or a parallel sysplex.

**TSO.** See Time Sharing Option.

**TSO console.** From this 3270-type console you are logged onto TSO or ISPF to use the runtime panels for I/O operations and SA z/OS customization panels.

**TSO/E.** See Time-Sharing Option/Extended.

**TWS.** See Tivoli Workload Scheduler.

**U**

**UCB.** See unit control block.

**unit control block (UCB).** A control block in common storage that describes the characteristics of a particular I/O device on the operating system and that is used for allocating devices and controlling I/O operations.

**unsolicited message.** An SA z/OS message that is not a direct response to a command.

**user task.** An application of the NetView program defined in a NetView TASK definition statement.
**Using.** An RMF Monitor III definition. Jobs getting service from hardware resources (processors or devices) are using these resources. The use of a resource by an address space can vary from 0% to 100% where 0% indicates no use during a Range period, and 100% indicates that the address space was found using the resource in every sample during that period.

**V**

**view.** In the NetView Graphic Monitor Facility, a graphical picture of a network or part of a network. A view consists of nodes connected by links and may also include text and background lines. A view can be displayed, edited, and monitored for status information about network resources.

**Virtual Server.** A logical construct that appears to comprise processor, memory, and I/O resources conforming to a particular architecture. A virtual server can support an operating system, associated middleware, and applications. A hypervisor creates and manages virtual servers.

**Virtual Server Collection.** A set of virtual servers that supports a workload. This set is not necessarily static. The constituents of the collection at any given point are determined by virtual servers involved in supporting the workload at that time.

**Virtual Server Image.** A package containing metadata that describes the system requirements, virtual storage drives, and any goals and constraints for the virtual machine (for example, isolation and availability). The Open Virtual Machine Format (OVF) is a Distributed Management Task Force (DMTF) standard that describes a packaging format for virtual server images.

**Virtual Server Image Capture.** The ability to store metadata and disk images of an existing virtual server. The metadata describes the virtual server storage, network needs, goals and constraints. The captured information is stored as a virtual server image that can be referenced and used to create and deploy other similar images.

**Virtual Server Image Clone.** The ability to create an identical copy (clone) of a virtual server image that can be used to create a new similar virtual server.

**Virtual Storage Extended (VSE).** A system that consists of a basic operating system (VSE/Advanced Functions), and any IBM supplied and user-written programs required to meet the data processing needs of a user. VSE and the hardware that it controls form a complete computing system. Its current version is called VSE/ESA.

**Virtual Telecommunications Access Method (VTAM).** An IBM licensed program that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability. Its full name is Advanced Communications Function for the Virtual Telecommunications Access Method. Synonymous with ACF/VTAM.

**VM Second Level Systems Support.** With this function, Processor Operations is able to control VM second level systems (VM guest systems) in the same way that it controls systems running on real hardware.

**VM/ESA®.** Virtual Machine/Enterprise Systems Architecture. Its current version is called z/VM.

**volume.** A direct access storage device (DASD) volume or a tape volume that serves a system in an SA z/OS enterprise.

**VSE.** See Virtual Storage Extended.

**VTAM.** See Virtual Telecommunications Access Method.

**W**

**warning threshold.** An application or volume service value that determines the level at which SA z/OS changes the associated icon in the graphical interface to the warning color. See alert threshold.

**workstation.** In SA z/OS workstation means the graphic workstation that an operator uses for day-to-day operations.

**write-to-operator (WTO).** A request to send a message to an operator at the z/OS operator console. This request is made by an application and is handled by the WTO processor, which is part of the z/OS supervisor program.

**write-to-operator-with-reply (WTOR).** A request to send a message to an operator at the z/OS operator console that requires a response from the operator. This request is made by an application and is handled by the WTO processor, which is part of the z/OS supervisor program.

**WTO.** See write-to-operator.

**WTOR.** See write-to-operator-with-reply.

**WWV.** The US National Institute of Standards and Technology (NIST) radio station that provides standard time information. A second station, known as WWVB, provides standard time information at a different frequency.

**X**

**XCF.** See cross-system coupling facility.
XCF couple data set. The name for the sysplex couple data set prior to MVS/ESA System Product Version 5 Release 1. See also sysplex couple data set.

XCF group. A set of related members that a multisystem application defines to XCF. A member is a specific function, or instance, of the application. A member resides on one system and can communicate with other members of the same group across the sysplex.

XRF. See extended recovery facility.

Z

z/OS. An IBM mainframe operating system that uses 64-bit real storage. See also Base Control Program.

z/OS component. A part of z/OS that performs a specific z/OS function. In SA z/OS, component refers to entities that are managed by SA z/OS automation.

z/OS subsystem. Software products that augment the z/OS operating system. JES and TSO/E are examples of z/OS subsystems. SA z/OS includes automation for some z/OS subsystems.

z/OS system. A z/OS image together with its associated hardware, which collectively are often referred to simply as a system, or z/OS system.

z196. See IBM Enterprise 196 (z196).

zAAP. See IBM System z Application Assist Processor (zAAP).

zBX. See IBM zEnterprise BladeCenter Extension (zBX).

zBX blade. See IBM zEnterprise BladeCenter Extension (zBX) blade.

zCPC. The physical collection of main storage, central processors, timers, and channels within a zEnterprise mainframe. Although this collection of hardware resources is part of the larger zEnterprise central processor complex, you can apply energy management policies to zCPC that are different from those that you apply to any attached IBM zEnterprise BladeCenter Extension (zBX) or blades. See also central processor complex.

zIIP. See IBM System z Integrated Information Processor (zIIP).

zEnterprise. See IBM zEnterprise System (zEnterprise).

Numerics

390-CMOS. Processor family group designator used in the SA z/OS processor operations documentation and in the online help to identify any of the following S/390 CMOS processor machine types: 9672, 9674, 2003, 3000, or 7060. SA z/OS processor operations uses the OCF facility of these processors to perform operations management functions. See OCF-based processor.
Index

Special characters
*BASE sample policy 303
+SA_PREDEFINED_MSGS MVS
Component entry 330

A
accessibility xiii
ACF 258
ACFFQRY 68
actions on policy databases 297
add-on sample policies
importing 341
listed 304
alternate configuration support 370
alternate configurations
building 371
defining 371
restrictions 372
alternate data sets, monitoring 91
AMC file 317
distributing 324
anchors
heartbeat 92
in RODM 92
Lost&Found 92
specifying 92
sysplex 92
AOCCLONE
See system automation symbols
AOCCLONE IDs 8
AOFSMARTMAT advanced automation
option, to enable AT updates 328
APPLGROUP INFO policy item 122
application
scenario, creating 38
application class 142
Application Defaults entry type 271
Application Defaults entry type, short
description 4
Application entry type
creating 140
introduction 140
policy items 145
application entry type, short
description 3
application group
nature 118
nature of 41
scenario, creating 39
Sysplex Application Group 40
System Application Group 40
application group entry type, short
description 3
application group resource name 41
APPLICATION GROUPS policy item
for systems 106
APPLICATION INFO policy item
for application defaults 272
for applications 152
application instance 142
application monitor routine 144, 158
application move, resource aware 134
application resource name 39
application resources
automatically linking 44
generating 43
APPLICATION SYMBOLS policy
item 162
application type 142
ApplicationGroup entry type
creating 117
introducing 116
policy items 120
APPLICATIONS policy item 125
applying service updates 332
ASCB chaining
MVS job type 143
with application automation 157
assigning
clone IDs 8
system automation symbols 8
AT
See NetView automation table
AT/MRT override syntax checking,
disabling 362
AT/MRT override syntax checking,
enabling 362
ATTEND TIME policy item 262
AUTHENTICATION policy item 286
AUTO MSG CLASSES policy item 82
automated function 168, 184, 188, 189
Automatic Restart Manager
defining element names 159
using clone values for element
names 159
automation
considerations 7
table policy definition 109
automation configuration
ACF 324
AMC file 324
automation control file 324
automation manager configuration
file 324
introducing 15
AUTOMATION CONSOLE policy
item 110
automation control file
building 317
distributing 324
automation flags
setting application defaults 272
AUTOMATION FLAGS policy item
for Application Defaults 272
for applications 160
for MVSCOMP Defaults 268
for System Defaults 269
automation manager configuration
file 317
building 317
automation manager configuration file
(continued)
distributing 324
automation name 41, 119
automation networks
definition 279
automation operators 272
Automation Operators entry type 272
Automation Operators entry type, short
description 4
AUTOMATION OPTIONS policy
item 269
automation policy
defining, outlined 1
overview of defining 1
automation procedure
functions managed by 7
introducing 7
recovery of components and data
sets 8
automation resource name 119
AUTOMATION SYMBOLS policy
item 109
autotask 4
availability target 132

B
BASIC 118
BASIC application group 41
batch build
INGEBCON 326
job 326
return codes 326
batch build processing
defining data sets 367
defining ISPF temporary data
sets 369
batch conversion 356
batch update
INGEBFLT 354
job 354
return codes 354
behavior of application groups
ACTIVE 119
PASSIVE 119
build
job for batch build 326, 354
BUILD 318
automation control file 317
automation manager configuration
file 317
description 317
MPFLSTSA member 317
NetView automation table 317
Policy Database Selection panel 298
system operations configuration
files 317
BUILD error messages 322
build processing
with BUILD command 318

© Copyright IBM Corp. 1996, 2012

397
BUILD command 318
bulk deletion 22
bulk update 22
See updating

C
canceling input 19
CDS 90
chaining 130
changing links
  concurrent user access 359
CICS automation 77
CICS Connection policy item 203
CICS CONNECTION policy item 250
CICS connections 249
CICS CONTROL policy item 203
CICS link definitions 77
policy items 250
CICS link monitor periods 77, 251
CICS subsystems
  policy items 203
classes, nested
  See nested classes
cleanup delay policy definition 157
CLISTS policy item 291
clustarch nodes 8
clone IDs
  assigning 8
  clone values 159
CMD processing 167
code processing 170
COMMAND DEFINITIONS policy item 95
COMMAND FLOODING policy item 97
command prefix 157
command routing mechanism 80
common routine, introduced 7
components
  recovery of with automation
    procedures 8
components, SA z/OS enterprise 75
concurrent customization by multiple users 357
concurrent user access 357
  changing links 359
  entry name selection 358
  entry type selection 358
  introducing 19
  policy database selection 358
  policy definitions 357
condition
  relationship 131
  shutdown 218
  startup 218
  unset condition 214
CONDITION policy item
  for a trigger 218
  for an event 214
configuration
  of automation 15
configuration files
  distributing 324
configuration focal point system 15
configuration support, alternate 370
connections (CICS) 249
considerations
  automation 7
  graphical interface 6
  console name 111
  controller details
    OPC 245
conversion
  job for batch conversion 356
  of initial policy database 356
  selective policy database
    conversion 356
conversion function 355
COPY policy item 27
copying policy
  information that is copied 28
  process 27
couple data set 90
creating
  application groups 117
  application groups, scenario 39
  applications 140
  applications, scenario 38
  events 212
  groups 84
  MOVE sysplex application group,
    scenario 49
  policy databases 299
  policy objects 19
  policy objects using text file
    updates 351
  policy objects, scenario 35
  processors 220
  SERVER sysplex application groups,
    scenario 56
  subgroups 100
  sysplex application groups,
    scenario 40, 49
  sysplexes, scenario 36
  system application groups,
    scenario 45, 48
  systems 102
  systems, scenario 36
  critical threshold 193
CRM 80
customization dialog 1
  canceling input 19
  changing environment 361
  features 17
  help 17
  invocation 16
  primary menu 16
  primary menu, introducing 16
  starting 16
  using 15, 17

D
data
  finding in a panel 23
  management 337
  moved and restructured 355
data sets
  (continued)
  defining for batch processing 367
  ISPF temporary, defining for batch
    processing 369
  monitoring 91
data sets
  (continued)
  recovery of with automation
    procedures 8
  DB2 CONTROL policy item 205
  DB2 subsystems
    policy items 204
Define Relationship panel 126
defining
  alternate configurations for
    systems 371
  automation policy, outlined 1
  automation policy, overview 1
  data sets for batch processing 367
  ISPF temporary data sets for batch
    processing 369
  relationships 127
deleting
  policy object 21
  policy objects using text file
    updates 354
  several policy objects 22
  single policy object 22
dependent resource 127
Description panel 27
DESCRIPTION policy item 26
distributed status, setting 123, 154, 233, 270,
  289
DesiredStatus vote 123, 154, 233, 270,
  289
disability xiii
disabling AT/MRT override syntax
  checking 362
DISPACF 68
displaying panel ID 5
distributing
  automation control file 324
  configuration files 324
DOCUMENTATION policy item 78
DOWNWARD CLASS/INST policy item 150
drain, JES2 198

E
E-T DATA policy item 295
element name, Automatic Restart
  Manager
    cloned 159
    constant 159
    defining 159
enabling AT/MRT override syntax
  checking 362
enterprise definition 15, 77
Enterprise entry type 77, 212
policy items 77
enterprise entry type, short
description 2
entry name selection
  concurrent user access 358
ten entry type
  Application 140
  Application Defaults 271
  ApplicationGroup 116
  Automation Operators 272
  defined 2
  Enterprise 77, 212
  fast path 18
job type (continued)
TRANSIENT 144
JOB/ASID DEFINITIONS policy item 94

K
keyboard xiii

L
labels, system, in overrides 329
link definitions (CICS) 77
link monitor periods (CICS) 77, 251
linking application resources
automatically 44
linking policy objects 29
load profile name 113
LOCAL PAGE DATA SET policy item 92
LOCATE 299
logging policy database
modifications 306
LookAt message retrieval tool xviii
Lost&Found anchor 92
LPARS AND SYSTEMS policy item 225

M
maintenance, policy database 297
MAKEAVAILABLE relationship 128
MAKEUNAVAILABLE relationship 128
MANAGE 298
MCSFLAGS policy definition 111
MEMBER OF policy item
for application groups 139
for applications 196
MESSAGE AUTOMATION policy item
for enterprises 253
message class 277
message classes 278
message prefix 158
message processing 165, 167
code processing 170
reply processing 169
scenario 69
user defined data 173
with user-defined data 69
message retrieval tool, LookAt xviii
message revision table
introducing 327
messages
predefined 330
predefined, modifying 331
response 115, 116
messages entry type, short description 4
MESSAGES/USER DATA policy item
for applications 165
for monitor resources 235
for MVS components 266
minor resources 267
MINOR RESOURCES policy item, for
applications 193
MINOR RESOURCES policy item, for
MVS components 267
missing heartbeat 103
processing 104
missing heartbeat delay 103
missing heartbeat processing 104
modifications to policy database,
logging 306
MONITOR INFO policy item 231
Monitor Resource
entry type 230
monitor resource entry type, short
description 3
monitoring
alternate data sets 91
primary data sets 91
tape mounts 261
MONITORING PERIOD policy item 252
Monitoring, Sysplex Timer 90
MOVE 118
MOVE application group 41
MOVE group
introduced 50
moved and restructured data 355
MOVED status
using 195
MPF DEFINITIONS policy item 256
MPFLSTSA member
building 317
distributing 324
MVS Component
+SA_PREDEFINED_MSGS 330
MVS Component entry type
using 263
MVS Component entry type, short
description 4
MVS job 143
MVS Target System IPL Information
panel 115
MVSCOMP Defaults entry type 267
MVSCOMP Defaults entry type, short
description 4
MVSDUMP
MVS Component entry type 263
MVSESA INFO policy item
for MVSCOMP Defaults 268
NetView
enabling updates 328
INGMSGSA 330
introduction 327
scenario 71
Network entry type 279
Network entry type, short description 4
NMC DEFINITIONS policy item 91
NONMVS job 143
NORM shutdown 186
notify operators entry type 276
NOTIFY OPERATORS policy item 276

O
OMEGAMON sessions policy
definition 283
OMEGAMON SESSIONS policy
item 283
OPC CONTROL policy item 209
OPC controller details 76, 245
policy items 245
OPC special resources 77, 247
OPC subsystems
policy items 208
OPC system details 76, 244
OPC workstation domains 77, 248
OPCA DOMAINDID policy item 248
OPCA PCS policy item 246
OPCA ENVIRON policy item 244
operator console
interested operator 114
output data set 321
overrides, application labels 329
overrides, system labels 329
overview of defining automation
policy 1

P
panel ID, displaying 5
PEEROF relationship 129
policy
add-on sample 304
basic sample 303
samples 303
policy database 15
adding to index 299
available actions 297
batch import 341
creating 299
creating policy objects using text file
updates 351
data, importing 337
data, updating policy text files 346
data, updating via file 344
initial conversion of 356
logging modifications 306
maintenance 297
reports 309
reports, batch creation 311
reports, job for batch creation 311
sending to remote nodes 370

400 System Automation for z/OS: Defining Automation Policy
Index 401
timers entry type, short description 4
TIMERS policy item 258
TRANSIENT job 144
trigger
link to an application group 121
TRIGGER policy item
for application groups 121, 125
for applications 163
for monitor resources 240
triggers
scenario 60
Triggers entry type 218
triggers entry type, short description 3
TWS automation 76

U
unset condition 213, 214
update
job for batch update 354
updating
policy database data policy text files 346
upgrading (continued)
policy database data via file 344
policy database data, block
replacement 347
policy database data, block
replacement examples 348
policy database data, field
replacement 347
policy database data, policy
replacement 349
policy database data, policy
replacement examples 349
user defined data processing 173
User E-T Pairs
entry type 294
scenario 67
User E-T Pairs entry type, short
description 5
user-written functions 369
USG CONTROL policy item 210
USS subsystems
policy items 210

V
VIEW 299
VM Target System IPL Information
panel 116
vote, DesiredStatus 123, 154, 233, 270,
289
VTAMLST 282

W
WHERE USED policy item 28
WLM resource name 160
workstation domains
OPC 248

Z
z/OS job name defined 144
zEnterprise Ensemble Entry Type
entry type 241
zEnterprise Ensemble policy item 242