



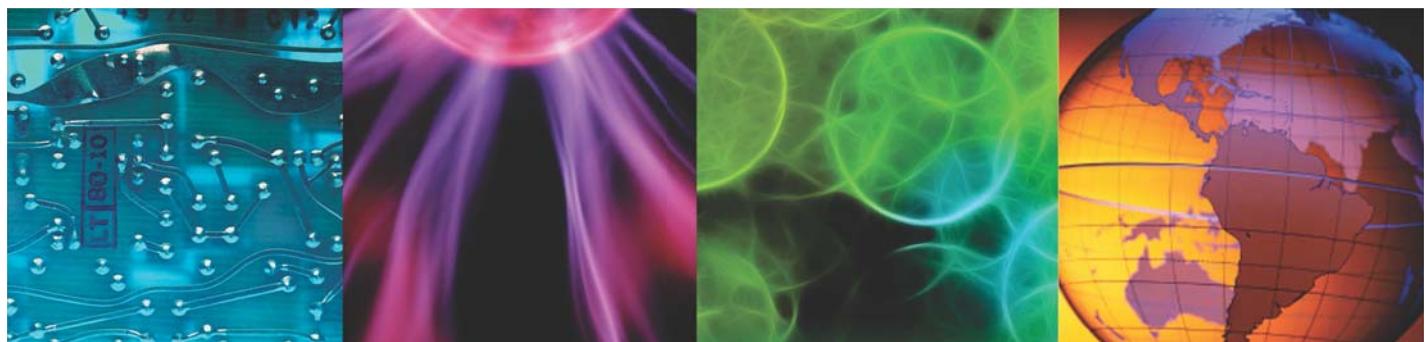
# IBM Training

## System Automation for z/OS 4.1 Implementation and Administration

### Student Notebook

Course code SM937 ERC 1.0

June 2019



All files and material for this course are IBM copyright property covered by the following copyright notice.

© Copyright IBM Corp. 2019. All Rights Reserved.

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

IBM, the IBM logo, and ibm.com are 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 "Copyright and trademark information" at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

References in this publication to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth, savings or other results.

# Contents

|  |           |
|--|-----------|
| About the student .....                                      | 14        |
| Learning objectives .....                                    | 15        |
| Course agenda .....  | 15        |
| Typographical conventions .....                              | 17        |
| Naming conventions .....                                     | 23        |
| <b>1 System Automation for z/OS overview.....</b>            | <b>23</b> |
| Lesson 1 An introduction to System Automation for z/OS ..... | 25        |
| Lesson agenda .....  | 26        |
| System Automation for z/OS: Components .....                 | 27        |
| System operations component .....                            | 29        |
| What can you automate? .....                                 | 31        |
| Summary of key automation technologies .....                 | 33        |
| Architecture .....   | 35        |
| Key automation features .....                                | 37        |
| Goal driven automation .....                                 | 40        |
| Lesson 2 Key operations features and user interfaces .....   | 42        |
| Key operations features .....                                | 43        |
| User interfaces .....  | 45        |
| Single Point of Control and Single System Image .....        | 46        |
| Lesson 3 Policy structure and resource types .....           | 47        |
| Automation policy structure and activation .....             | 48        |
| Resource Types (partial list) .....                          | 49        |
| <b>2 Installation and customization .....</b>                | <b>55</b> |
| Lesson 1. Planning .....                                     | 58        |
| Prerequisites for SA z/OS 4.1 .....                          | 61        |
| Lesson 2. z/OS and OMVS configuration .....                  | 62        |
| Customize PARMLIB members .....                              | 64        |
| Customize other PARMLIB members .....                        | 66        |
| Update COMMNDxx member .....                                 | 68        |
| Customizing the system logger .....                          | 69        |
| Defining an SA z/OS agent to VTAM .....                      | 70        |
| Automatic restart manager considerations .....               | 72        |
| Customizing OMVS .....                                       | 73        |
| Lesson 3. Configuring automation manager .....               | 74        |
| Automation Manager start procedure .....                     | 75        |
| Automation manager start procedure (continued) .....         | 77        |
| Automation manager HSAPRMxx member .....                     | 78        |
| Running HSAPIPLC at IPL .....                                | 80        |
| Lesson 4. Automation agent and NetView configuration .....   | 81        |
| NetView start procedure example .....                        | 83        |

|   |            |
|---|------------|
| NetView STEPLIB .....   | 85         |
| NetView DSICLD and DSIPARM .....  | 86         |
| NetView data sets .....   | 88         |
| Additional SA z/OS data sets .....  | 89         |
| NetView SSI start procedure example .....   | 90         |
| Subsystem name table .....  | 92         |
| DSIPARM definitions .....   | 93         |
| Customizing the communications task .....   | 95         |
| XCF overview .....  | 97         |
| Display XCF groups .....  | 98         |
| Customizing SA z/OS operations .....  | 99         |
| Control load of AT and MRT .....  | 101        |
| EMCS consoles for operators .....   | 102        |
| More CNMSTYLE customization .....   | 103        |
| Switch off network related functions .....  | 104        |
| NetView enable RMTCMD .....   | 105        |
| Lesson 5 REXX function packages .....   | 107        |
| INGTXFPG .....  | 109        |
| Lesson 6 Security and Authorization .....   | 110        |
| Authorization of the started procedures .....   | 112        |
| Setting up security .....   | 113        |
| Security considerations for automation .....  | 114        |
| SECOPTS statements and options in CNMSTYLE .....  | 115        |
| Recommended stylesheet options .....  | 117        |
| Configuration assistant generates the INGESAF member .....                                  | 119        |
| Security definitions migration to SAF and backup .....                                      | 120        |
| Roles .....   | 121        |
| Lesson 7 Configuration assistant .....  | 122        |
| Traditional approach .....  | 124        |
| Configuration assistant .....   | 125        |
| Configuration assistant: Concept .....  | 126        |
| INGDOPT Configuration Options file .....  | 127        |
| INGDOPT Configuration Options file for jobcard, ..... names, SMS, HLQs... .....             | 129        |
| INGDOPT Configuration Options file for SA items:XCF, PDB, procedure and job names .....     | 131        |
| Optional configuration options for Workload Scheduler, CICS, Event/Automation Service ..... | 133        |
| Optional configuration options for Security Server RACF .....                               | 134        |
| Follow instructions in CONFLIB(\$INGREAD) 1 of 2 .....                                      | 135        |
| Follow instructions in CONFLIB(\$INGREAD) 2 of 2 .....                                      | 137        |
| Review questions .....  | 139        |
| Review answers .....  | 140        |
| <b>Unit 3. Customization dialog and Policy Data Base.....</b>                               | <b>144</b> |
| Lesson 1 Customization dialog .....   | 146        |
| Customization dialog .....  | 147        |
| Policy process .....  | 148        |
| Policy contains policy objects linked by WHERE USED .....                                   | 149        |
| Policy Data Base .....  | 150        |
| Customization dialog primary menu .....   | 151        |

|  |            |
|--|------------|
| Customization dialog settings .....  | 152        |
| PDB samples and add-on policies .....  | 153        |
| Add-on policies .....  | 154        |
| Lesson 2 Creating a Policy Data Base .....                                       | 155        |
| Creating a Policy Data Base .....  | 155        |
| Creating a Policy Data Base .....  | 156        |
| Allocate the Policy Data Base (1 of 2) .....                                     | 157        |
| Allocate the Policy Data Base (2 of 2) .....                                     | 158        |
| Entry Type Selection panel .....   | 159        |
| PDB documentation .....  | 161        |
| PDB documentation details .....  | 162        |
| Policy PDF files .....   | 163        |
| New PDB contents .....   | 164        |
| Lesson 3 Introduction to defining automation policy .....                        | 165        |
| Introduction to defining automation policy .....                                 | 165        |
| Modify a system entry .....  | 167        |
| Rename system entry .....  | 168        |
| SYSTEM INFO policy item .....  | 169        |
| Modify an Application .....  | 171        |
| Rename SYSVSSI .....   | 172        |
| Policy Selection panel: Application .....  | 173        |
| APPLICATION INFO panel .....   | 174        |
| Application SHUTDOWN policy .....  | 176        |
| Create HasParent relationship .....  | 177        |
| WHERE USED policy .....  | 178        |
| Lesson 4 Other policy customization features .....                               | 179        |
| Other policy customization features .....  | 179        |
| Query policy statistics .....  | 180        |
| Copy Application policy example .....  | 181        |
| Data Management: Import from PDB .....   | 182        |
| Import from PDB: Select one or more entries .....                                | 183        |
| Data management: Import add-on policy .....                                      | 184        |
| Example add-on import: *ITM .....  | 185        |
| Remove (M) or rename (R) (duplicate) entries .....                               | 186        |
| Lesson 5 Policy Data Base reports .....  | 187        |
| Policy Data Base reports .....   | 187        |
| Report Selection Menu .....  | 188        |
| PDB report files .....   | 190        |
| Reports: FLAT versus HTML .....  | 191        |
| Policy Data Base Report Parameters panel .....                                   | 192        |
| HTML report example .....  | 193        |
| <b>4 Defining applications and building automation configuration files .....</b> | <b>197</b> |
| Lesson 1. Defining Application policy .....                                      | 200        |
| New Application creation .....   | 201        |
| APPLICATION INFO policy .....  | 204        |
| APPLICATION INFO: Desired Available .....  | 207        |
| APPLICATION INFO: Restart after IPL .....  | 209        |

|   |     |
|---|-----|
| APPLICATION INFO: Restart option .....                | 211 |
| APPLICATION INFO: Inform List .....                   | 213 |
| APPLICATION INFO: Additional options .....            | 215 |
| APPLICATION INFO: Transient resources .....           | 216 |
| APPLICATION INFO: NonMVS job type .....               | 217 |
| RELATIONSHIPS policy .....                            | 219 |
| Relationship conditions .....                         | 221 |
| STARTUP policy .....                                  | 222 |
| Start processing .....                                | 223 |
| STARTUP policy item .....                             | 224 |
| PRESTART phase commands .....                         | 226 |
| STARTUP phase commands .....                          | 227 |
| POSTSTART phase commands .....                        | 229 |
| External start considerations .....                   | 230 |
| SHUTDOWN policy .....                                 | 231 |
| Stop processing .....                                 | 232 |
| SHUTDOWN phases .....                                 | 233 |
| TSO SHUTINIT and SHUTNORM commands .....              | 234 |
| SHUTNORM: Reply and command example for NetView ..... | 236 |
| SHUTFINAL commands .....                              | 238 |
| JES2 shutdown example .....                           | 239 |
| Stop considerations .....                             | 240 |
| Stop by parent .....                                  | 241 |
| MESSAGES/USER DATA policy .....                       | 242 |
| Default message entries .....                         | 243 |
| Specifying a WTOR response in the policy .....        | 245 |
| MESSAGES/USER DATA for RV01 .....                     | 246 |
| Set automation status based on message ID .....       | 247 |
| Issue commands when status changes .....              | 248 |
| Automation Table actions .....                        | 249 |
| Message override automatic syntax check .....         | 250 |
| Automation flags .....                                | 251 |
| Threshold processing .....                            | 252 |
| Threshold processing .....                            | 254 |
| Major and minor resources .....                       | 256 |
| Example: Minor resource threshold for RV05 .....      | 258 |
| Example: RV05 MINOR RESOURCES thresholds .....        | 259 |
| Example: RV05 MESSAGES/USER DATA policy .....         | 260 |
| Example: VAPL25020I commands .....                    | 261 |
| Policy inheritance for Applications .....             | 262 |
| Minor resource inheritance example .....              | 263 |
| Application default specifications .....              | 264 |
| Lesson 2. Automation configuration file .....         | 265 |
| Customization dialog: Process .....                   | 267 |
| Automation configuration file format .....            | 268 |
| Automation configuration file members example .....   | 269 |
| Automation agent files .....                          | 270 |
| Automation manager files .....                        | 271 |

|  |            |
|--|------------|
| Map member: AOFACFMP .....                             | 272        |
| Automation agent main member example .....             | 273        |
| Lesson 3. Build automation configuration files .....   | 274        |
| Invoking a build for the first time .....              | 276        |
| Tailoring the message stream .....                     | 278        |
| Lesson 4. Bulk policy updates .....                    | 279        |
| Flat file update .....                                 | 280        |
| Supported policy definitions .....                     | 281        |
| Write selected data from PDB to file .....             | 282        |
| Write selected data from PDB to file (continued) ..... | 284        |
| Flat file format: Header information .....             | 285        |
| Flat file format example: RV01 policies .....          | 286        |
| Flat file syntax .....                                 | 287        |
| PDB update from flat file .....                        | 289        |
| Flat file update serialization .....                   | 290        |
| Creating new entries from the flat file .....          | 291        |
| Example: Creating a new application .....              | 292        |
| Deleting multiple entries .....                        | 294        |
| Deleting multiple entries example .....                | 295        |
| Perform the exercises for this unit. ....              | 296        |
| <b>5 Additional customization.....</b>                 | <b>298</b> |
| Lesson 1. Common Application definitions .....         | 301        |
| SA z/OS symbols .....                                  | 303        |
| Defining SA z/OS symbols .....                         | 304        |
| Example: Symbols in the job name .....                 | 305        |
| Example: Symbols in STARTUP command .....              | 306        |
| Application classes and instances .....                | 307        |
| Nested Application classes .....                       | 309        |
| Nested Application classes example .....               | 310        |
| Data inheritance rules .....                           | 311        |
| Nested Application classes: Notes .....                | 312        |
| Create an Application class .....                      | 313        |
| UPWARD CLASS: Upward link to class .....               | 314        |
| DOWNWARD CLASS/INST: Downward link .....               | 315        |
| Implement planned suspend capability .....             | 316        |
| Suspended resources .....                              | 317        |
| Lesson 2. Additional policy definitions .....          | 319        |
| Notify operators .....                                 | 320        |
| Defining Notify Operators .....                        | 321        |
| Define a timer .....                                   | 322        |
| JES2 spool monitoring and control .....                | 324        |
| SPOOLSHORT processing .....                            | 326        |
| MVS automation .....                                   | 329        |
| MVSDUMP thresholds specification .....                 | 330        |
| MVS component: MVSDUMPTAKEN message .....              | 331        |
| MVSDUMPTAKEN automation flags .....                    | 332        |
| MVS console buffer recovery .....                      | 333        |

|  |            |
|--|------------|
| Automation operators policy .....  | 335        |
| Base automation operators policy .....   | 336        |
| Important operators .....  | 338        |
| Work operators (AOFWRKxx) .....  | 339        |
| Message assignment .....   | 341        |
| Routing specific messages .....  | 342        |
| Notifying the MVS console .....  | 343        |
| MVSCONS automated operator .....   | 344        |
| Commands from MVS consoles .....   | 345        |
| Sysplex with subplex example .....   | 347        |
| Sysplex policy items .....   | 348        |
| <b>Lesson 3. Defining Service Periods, Events, Triggers and Pacing Gates .....</b> | <b>349</b> |
| Define a Service Period .....  | 351        |
| Service Period resources .....   | 353        |
| Example: One Trigger with several Events .....                                     | 354        |
| Define an Event .....  | 355        |
| Define a Trigger .....   | 356        |
| Set Event for BKP1 .....   | 358        |
| Pacing Gates .....   | 359        |
| Create Pacing Gates .....  | 361        |
| Link application to one Pacing Gate .....  | 362        |
| <b>Lesson 4. Defining USS and MTR resources .....</b>                              | <b>363</b> |
| UNIX System Services (USS) automation .....  | 365        |
| Defining a USS Application .....   | 367        |
| USS CONTROL policy .....   | 368        |
| USS Application STARTUP .....  | 370        |
| Monitor resources overview .....   | 371        |
| Active monitor versus passive monitor .....  | 372        |
| Monitor resource definition .....  | 373        |
| Connecting APIs or APGs to MTRs .....  | 375        |
| System administrator and operator tasks .....                                      | 376        |
| Overview of monitor processes .....  | 377        |
| Monitor command basics .....   | 378        |
| Health Status aggregation .....  | 379        |
| HEALTHSTATE policy item .....  | 381        |
| HasMonitor relationship .....  | 382        |
| Passive monitor: MESSAGES/USER DATA .....  | 383        |
| Passive monitor: Set Health Status .....   | 384        |
| INGLIST JES* example .....   | 385        |
| DISPMTR details example: Active monitor .....                                      | 386        |
| DISPMTR details example: Passive monitor .....                                     | 387        |
| <b>Lesson 5. Automation Tables .....</b>   | <b>388</b> |
| MPFLSTSA and INGEMPF .....   | 390        |
| INGMSG01 .....   | 391        |
| INGMSGU1 and INGMSGU2 .....  | 392        |
| INGMSGSA .....   | 393        |
| <b>Lesson 6. Dynamic AT and MRT build .....</b>                                    | <b>394</b> |
| INGMSG02 dynamic loading .....   | 396        |

|   |            |
|---|------------|
| INGMSG02 content .....  | 397        |
| MESSAGES/USER DATA: AT statements .....                       | 398        |
| MESSAGES/USER DATA: AT and MRT actions .....                  | 399        |
| MESSAGES/USER DATA: AT and MRT overrides .....                | 400        |
| Message override syntax check .....                           | 401        |
| Status message reports .....                                  | 402        |
| Status message report example .....                           | 403        |
| Lesson 7. Advanced customization topics .....                 | 404        |
| SA z/OS exits .....   | 405        |
| AT load exits: Example .....                                  | 409        |
| Initialization exits example .....                            | 410        |
| Example command exit: AOFEXC01 .....                          | 411        |
| Advanced automation options (AAOs) .....                      | 412        |
| AAO examples .....  | 413        |
| AAO example: AOFINITREPLY .....                               | 414        |
| Example generic routines .....                                | 415        |
| <b>6 Initialization and runmodes .....</b>                    | <b>419</b> |
| Lesson 1. Structure and availability .....                    | 423        |
| The automation manager environment .....                      | 424        |
| The automation agent environment .....                        | 426        |
| Sysplex environment model .....                               | 427        |
| Automation manager availability .....                         | 428        |
| Lesson 2. Initialization options .....                        | 429        |
| Manager initialization .....                                  | 430        |
| Manager start types .....                                     | 431        |
| Agent initialization sequence .....                           | 432        |
| Agent initialization options .....                            | 434        |
| Lesson 3. Runmodes .....                                      | 436        |
| Characteristics of runmodes .....                             | 438        |
| Defining runmodes .....                                       | 440        |
| Defining runmodes (continued) .....                           | 441        |
| Defining runmodes (continued) .....                           | 442        |
| Assigning an APL or MTR to a runmode .....                    | 443        |
| INGRUN command .....  | 444        |
| <b>7 Gateways and Status Display Facility .....</b>           | <b>447</b> |
| Lesson 1. Gateways .....                                      | 450        |
| An overview of gateways .....                                 | 451        |
| Definition of gateways .....                                  | 452        |
| GATEWAY Policy item .....                                     | 453        |
| Gateway autotasks .....                                       | 454        |
| Lesson 2. Status Display Facility overview .....              | 455        |
| Introduction to SDF .....                                     | 456        |
| SDF displays .....  | 457        |
| Status Display Facility focal point .....                     | 459        |
| Gateways, focal point, and SPOC .....                         | 460        |
| Customized root component (SINGNPRM SDF sample INGPTOP) ..... | 461        |
| Summary status for a system .....                             | 462        |

|   |            |
|---|------------|
| Subsystem status . . . . .                              | 463        |
| Application group status . . . . .                      | 464        |
| Monitor status . . . . .                                | 465        |
| Detailed status component . . . . .                     | 466        |
| Gateway status . . . . .                                | 467        |
| WTOR status . . . . .                                   | 468        |
| Lesson 3. How SDF works . . . . .                       | 469        |
| SDF tree structure . . . . .                            | 471        |
| SDF tree structure example . . . . .                    | 472        |
| Status management in SDF . . . . .                      | 473        |
| Lesson 4. Customizing SDF panels . . . . .              | 475        |
| SDF features . . . . .                                  | 476        |
| SDF panel definition statements . . . . .               | 477        |
| SDF panel definition statements (continued) . . . . .   | 478        |
| PANEL definition statement . . . . .                    | 479        |
| Defining text fields: TF and TT . . . . .               | 480        |
| Defining text fields: Example . . . . .                 | 481        |
| Defining fixed text status fields: SF and ST . . . . .  | 482        |
| Defining dynamic status fields: SF and ST . . . . .     | 484        |
| Status descriptor use . . . . .                         | 485        |
| Defining PF keys . . . . .                              | 486        |
| Example of SDF panel and field definitions . . . . .    | 487        |
| SDF multiple body sections example . . . . .            | 489        |
| Body statements . . . . .                               | 490        |
| Lesson 5. Customizing SDF parameters . . . . .          | 492        |
| DSIPARM customization and members . . . . .             | 494        |
| AOFTREE definition . . . . .                            | 496        |
| AOFPNLS definition . . . . .                            | 497        |
| AOFINIT definition . . . . .                            | 498        |
| Using a PANEL definition . . . . .                      | 500        |
| Extending the tree . . . . .                            | 501        |
| Extending the tree: Config Refresh Monitoring . . . . . | 502        |
| Config Refresh Monitoring Detail Status . . . . .       | 503        |
| Refreshing SDF . . . . .                                | 504        |
| Primary and backup focal points . . . . .               | 506        |
| Lesson 6. SDF policy customization . . . . .            | 508        |
| Customize SDF color and priority . . . . .              | 509        |
| Setting the WTOR colors: Messages policy . . . . .      | 510        |
| Setting the WTOR colors: Code processing . . . . .      | 511        |
| Perform the exercises for this unit. . . . .            | 513        |
| <b>8 Defining Application Groups . . . . .</b>          | <b>515</b> |
| Lesson 1. Overview of Application Groups . . . . .      | 518        |
| Why application groups? . . . . .                       | 520        |
| Group type and nature . . . . .                         | 521        |
| Characteristics of an application group . . . . .       | 523        |
| System application group . . . . .                      | 525        |
| Sysplex application group . . . . .                     | 527        |

|   |            |
|---|------------|
| Notes on the automation name .....  | 528        |
| Group behavior: Active versus passive .....   | 529        |
| Server group .....  | 530        |
| Preference values and their effect on group policy .....  | 532        |
| Automation manager bonus values .....   | 534        |
| Group modes .....   | 535        |
| Basic steps to define a group .....   | 536        |
| Lesson 2. Define a sysplex server group .....   | 537        |
| Step 1: Create the server group .....   | 539        |
| TSOGROUP policy panel .....   | 541        |
| Step 2: Link group to SAplex .....  | 542        |
| Step 3: Link application to group .....   | 543        |
| Step 4: Define owner and behavior .....   | 544        |
| Step 5: Set availability target and preferences .....   | 546        |
| INGGROUP for TSO server group .....   | 548        |
| Lesson 3. Groups with move nature .....   | 549        |
| Initiating application moves .....  | 551        |
| Move APG policy: Move mode .....  | 552        |
| Move APG policy: Prepare move=YES .....   | 553        |
| INGMOVE example in a SAplex .....   | 554        |
| Lesson 4. Define a nested Basic System Group .....  | 555        |
| Step 1: Create the basic group .....  | 556        |
| NETGROUP policy panel .....   | 557        |
| Step 2: Link groups to systems .....  | 558        |
| Step 3: Link Applications to group .....  | 559        |
| Step 4: Modify resource links .....   | 560        |
| Modify resource links: Add IPGROUP .....  | 561        |
| INGGROUP for NETGROUP APG .....   | 562        |
| <b>9 End-to-end automation .....</b>  | <b>565</b> |
| Lesson 1. End-to-end automation overview .....  | 568        |
| .....   | 568        |
| End-to-end automation goals and functions .....   | 569        |
| End-to-end automation architecture .....  | 570        |
| End-to-end automation concepts .....  | 571        |
| Invokes start or stop commands for remote resources on behalf of the E2E automation manager ..... | 572        |
| End-to-end automation concepts (cont.) .....  | 573        |
| End-to-end manager, agent, and adapter roles .....  | 575        |
| End-to-end automation architecture details .....  | 576        |
| Lesson 2. Prerequisites, planning, and activation .....   | 578        |
| End-to-end automation prerequisites .....   | 580        |
| End-to-end automation planning .....  | 581        |
| End-to-end automation configuration assistant - OA52610 .....                                     | 582        |
| End-to-end automation activation steps .....  | 583        |
| Lesson 3. End-to-end agent .....  | 584        |
| End-to-end agent .....  | 586        |
| Environment setup for E2E agent Java component .....  | 587        |
| Default install-root directory: /usr/lpp/ing/adapter/ .....                                       | 588        |

|   |  |
|---|--|
| Example custom-root directory: /u/in/adapter .....  | 589  |
| E2E agent environment configuration file .....  | 590  |
| E2E agent JCL procedure .....   | 591  |
| E2E agent credential file .....   | 592  |
| Start the E2E agent .....   | 593  |
| Starting the E2E agent .....  | 594  |
| E2E agent terminates .....  | 595  |
| E2E agent terminates again .....  | 596  |
| Check if E2E automation is active .....   | 597  |
| Stop the E2E agent .....  | 598  |
| Authentication to a remote domain .....   | 600  |
| <b>Lesson 4. End-to-end adapter .....</b>   | <b>601</b>   |
| End-to-end adapter .....  | 602  |
| End-to-end adapter events .....   | 604  |
| End-to-end adapter configuration parameters: Requests .....                                   | 606  |
| End-to-end adapter customization steps .....  | 607  |
| Enable NetView Event Automation Service (E/AS) .....  | 608  |
| Customize SA and NetView .....  | 609  |
| End-to-end adapter: Successful start .....  | 610  |
| INGE2E command provides the operational API .....   | 611  |
| <b>Lesson 5. End-to-end automation policy .....</b>   | <b>612</b>   |
| *E2E Best Practices Policy .....  | 613  |
| Summary DMN and REF resources .....   | 614  |
| Define E2E Policy: DMN Remote Domains entry type .....  | 615  |
| DMN Remote Domains entry type: Policies and linkage to SAplex .....                           | 616  |
| Define E2E Policy: REF Resource Reference entry type .....                                    | 617  |
| Define E2E Policy: REF Resource Reference entry type .....                                    | 618  |
| <b>Lesson 6. End-to-end user interface .....</b>  | <b>619</b>   |
| End-to-end automation User Interface .....  | 620  |
| INGLIST showing DMN and REF resources .....   | 621  |
| INGLIST showing DMN and REF resources. Scrolled to the right .....                            | 622  |
| INGINFO showing remote domain .....   | 623  |
| INGINFO showing connected remote domain .....   | 624  |
| E2E agent can not connect to remote E2E adapter .....   | 625  |
| INGDMN showing remote domain .....  | 626  |
| Use INGMOVE to “move” member from MVSA into sysplex PLEX56 .....                              | 627  |
| REF resources on SMU .....  | 628  |
| <b>Lesson 7. End-to-end automation scenarios .....</b>  | <b>629</b>   |
| Cross sysplex automation example: Monitor an end-to-end application .....                     | 630  |
| Extending automation scope across sysplexes .....   | 631  |
| Extending automation scope from sysplexes to distributed platforms & cloud applications ..... | 632  |
| Sample scenario: Stop „MyPortal“ for maintenance .....  | 633  |
| Log management with Beta 92 .....   | 634  |
| Cross sysplex scenario Provider – Clients setup .....   | 635  |
| <b>10 .....</b>   | <b>Performance-based automation with OMEGAMON637</b> |
| <b>Lesson 1. Overview of OMEGAMON integration .....</b>                                       | <b>640</b>   |
| Example command flow .....  | 642  |

|  |     |
|--|-----|
| Exception-based monitoring overview .....                          | 644 |
| Example provisioning scenario .....                                | 645 |
| Event-based monitoring overview .....                              | 646 |
| Event-based automation example .....                               | 647 |
| Lesson 2. Implement *ITM add-on policy .....                       | 649 |
| Import *ITM add-on policy .....                                    | 651 |
| Import *ITM add-on policy (continued) .....                        | 652 |
| Define session with OMEGAMON classic .....                         | 654 |
| Define automation operators .....                                  | 656 |
| Lesson 3. Commands and programming interfaces .....                | 657 |
| INGESS example .....   | 659 |
| INGESS details .....   | 661 |
| Recovering a failed session .....                                  | 663 |
| INGOMX overview .....  | 664 |
| OMEGAMON command examples .....                                    | 666 |
| INGOMX TRAP XTYPE examples .....                                   | 667 |
| Lesson 4. MTR policy definitions .....                             | 668 |
| Exceptions and Health Status .....                                 | 670 |
| MTR policy definitions: Overview .....                             | 672 |
| INGMTRAP as a monitor command .....                                | 673 |
| INGMTRAP .....   | 674 |
| Exception-based automation .....                                   | 676 |
| MESSAGES/USER DATA example .....                                   | 677 |
| Map OMEGAMON exception to Health Status .....                      | 678 |
| Issue a command based on an exception .....                        | 679 |
| Issue a command based on an health state .....                     | 680 |
| Using PASS counts for escalation .....                             | 681 |
| Suspending exception analysis .....                                | 682 |
| MESSAGES/USER DATA summary of PASSes .....                         | 683 |
| Automation Table fragment for ING080I .....                        | 684 |
| CODEs for selective processing .....                               | 685 |
| Example: ING080I tokens for XREP .....                             | 686 |
| Example: Define CODEs for XREP .....                               | 687 |
| Example: Define commands for XREP .....                            | 688 |
| Example: AT overrides for XREP .....                               | 689 |
| MESSAGES/USER DATA summary of CODEs .....                          | 690 |
| Example: XREP automation .....                                     | 691 |
| Define relationships .....   | 692 |
| Clear monitor status .....   | 693 |
| Lesson 5. Security considerations .....                            | 694 |
| OMEGAMON security and SA z/OS .....                                | 695 |
| NetView command authorization .....                                | 697 |
| CAT example .....  | 699 |
| Session passwords .....  | 700 |
| Lesson 6. SA z/OS monitoring agent .....                           | 702 |
| Tivoli Enterprise Portal architecture .....                        | 703 |
| Tivoli Enterprise Portal topology view .....                       | 704 |
| Tivoli Enterprise Portal topology view nodes and connections ..... | 705 |

|  |            |
|--|------------|
| Installation and customization overview .....                                | 706        |
| Supported platforms .....  | 707        |
| Runtime environment examples .....   | 708        |
| Choose a configuration .....   | 709        |
| Installing with z/OS Tivoli Enterprise Monitoring Server: Overview .....     | 711        |
| Install prerequisites .....  | 713        |
| Define SA z/OS policy .....  | 714        |
| Configure SA z/OS monitoring agent .....                                     | 715        |
| SA z/OS monitoring agent with z/OS Tivoli Enterprise Monitoring Server ..... | 717        |
| Resource Overview workspace .....  | 718        |
| Resource Requests workspace .....  | 719        |
| Other Tivoli Enterprise Portal customization .....                           | 720        |
| Event-based automation: Setup tasks .....                                    | 722        |
| MONITOR INFO policy .....  | 723        |
| SOAP server requests .....   | 724        |
| SOAP SERVER policy item .....  | 725        |
| SOAP server request example .....  | 726        |
| <b>11 Migration and coexistence.....</b>                                     | <b>729</b> |
| Lesson 1. Migration .....  | 732        |
| Apply maintenance and convert PDBs .....                                     | 733        |
| Review release-related changes .....   | 734        |
| Lesson 2. Coexistence .....  | 736        |
| Restrictions concerning suspend and resume functionality (1 of 2) .....      | 738        |
| Restrictions concerning suspend and resume functionality (2 of 2) .....      | 739        |
| ACF and takeover file coexistence .....                                      | 740        |
| Review questions .....   | 741        |
| Review answers .....   | 742        |



# About this course

---



## IBM System Automation for z/OS 4.1 Implementation and Administration



© Copyright IBM Corporation 2019

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this course you learn how to define automation policy for IBM System Automation for z/OS® (SA z/OS). You learn, for example, how to create policy definitions for systems, Applications, Application Groups, and monitor resources.

This training class is delivered in an environment with multiple opportunities for hands-on lab exercises.

You define automation policy for several environments: single system and multiple system within a basic sysplex.

In this course, the System Automation for z/OS automation manager and automation agent run in a z/OS 2.2 environment.

The automation platform, Tivoli® NetView for z/OS is at version 6 release 2.

For information about other related courses, see the IBM education training paths website:

[ibm.com/training/](http://ibm.com/training/)

| Details                    |  |
|----------------------------|--|
| <b>Delivery method</b>     | Classroom  |
| <b>Course level</b>        | ERC 1.0<br><br>This course is an update of SM936: IBM Tivoli System Automation for z/OS 3.4 Implementation and Administration. |
| <b>Product and version</b> | IBM System Automation for z/OS 4.1<br><br>Including updates June 2018: APAR OA55386 and Service Management Unite 1.1.4         |
| <b>Duration</b>            | 5 days   |
| <b>Skill level</b>         | Intermediate / Advanced  |



**Note:** This course applies to IBM System Automation for z/OS (System Automation), but can be also used for IBM Automation Control for z/OS. Functions not provided by IBM Automation Control for z/OS are in *underlined italics*. The terms automation and the automation product are used when there are no differences. Alternatively the product name is used to describe its functions.

## About the student

This course is designed for users of IBM System Automation for z/OS, especially administrators, and system programmers who are responsible for defining the automation policy that is used by System Automation for z/OS.

Before taking this course, make sure that you have attended the following course or have equivalent skills:

- *SM917 - IBM System Automation for z/OS 4.1 Architecture*

Before taking this course, make sure that you have the following skills:

- Good knowledge of System Automation for z/OS architecture and concepts
- Basic knowledge of System Automation for z/OS operations
- Basic z/OS operations skills and started task concepts
- Basic NetView skills
- Basic ISPF use
- JCL coding and z/OS data set allocation

Recommended courses before this course are:

- *TZ203 - IBM Tivoli NetView for z/OS 6.1 Fundamentals*
- *TZ213 - IBM Tivoli NetView for z/OS 6.1 Automation*

## Learning objectives

The learning objectives are to get the required knowledge to use System Automation for z/OS or IBM Automation Control for z/OS to perform the following tasks:

- Describe the customization of the z/OS system and NetView to prepare for installation and customization
- Customize the automation policy to automate z/OS applications
- Build and activate the required automation configuration files from the Policy Data Base (PDB)
- Monitor and control automated resources
- Describe the coexistence and migration issues for current releases
- Customize the SDF component
  - Navigate and verify the customized SDF panels

## Course agenda

The course contains the following units:

1. [System Automation for z/OS overview](#)

This unit provides an introduction to using System Automation for z/OS and NetView. This unit repeats some of the contents from the architecture course like introduction to System Automation and its components, component System Operations, its architecture and its key features.

2. [Installation and customization](#)

This unit introduces the installation and customization of System Automation for z/OS, including the automation manager, automation agent, and other required z/OS components either using the

traditional approach or by using the configuration assistant.

### [3. Customization dialog and Policy Data Base](#)

This unit introduces the SA z/OS customization dialog and Policy Data Base (PDB). 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. You learn about simple policy definitions for systems and applications, importing policies from add-on policies, and the types of reports that can be generated.

### [4. Defining applications and building automation configuration files](#)

This unit focuses on how to define an Application by using the SA z/OS Customization dialog and how to build the configuration files that are used by the automation manager and automation agent. You also learn about the format of the configuration files and how to update the Policy Data Base (PDB) from flat files.

### [5. Additional customization](#)

In this unit, you explore additional customization features and functions that are available with SA z/OS. Topics include use of symbolics in the policy, Application classes, suspend file, Notify Operators, Service Periods, Events, Triggers, and monitor resources.

### [6. Initialization and runmodes](#)

In this unit, you learn about the initialization and synchronization of the automation manager and automation agent. You have several options for initializing each component. You also learn about refreshing the automation configuration file and managing the primary automation manager. The concept of runmodes, as it relates to system initialization, is also covered. In this unit, you learn about the initialization and synchronization of the automation manager and automation agent. You have several options for initializing each component. You also learn about refreshing the automation configuration file and managing the primary automation manager. The concept of runmodes, as it relates to system initialization, is also covered.

### [7. Gateways and Status Display Facility](#)

In this unit, you learn how to define Gateway sessions and autotasks and how to use and customize the Status Display Facility (SDF) component of SA z/OS. SDF displays the status of your SA z/OS resources from 3270 panels. You can customize the SDF panels to aggregate status or to group resources based on business function, for example.

### [8. Defining Application Groups](#)

This unit provides in-depth information about Application Groups. This information includes the types of Application Groups, ways to define them in the automation policy, and ways to manage them within your enterprise.

### [9. End-to-end automation](#)

This unit covers cross sysplex and end-to-end automation, its architecture, customization, policy definition, operation user interfaces, and scenarios.

## [10. Performance-based automation with OMEGAMON](#)

This unit contains information about using SA z/OS and the OMEGAMON products for performance-based automation: exception-based monitoring and event-based monitoring. You learn how to define the OMEGAMON Applications, sessions between SA z/OS and the OMEGAMON Classic monitors, and monitor resources to support performance-based automation. The unit also contains information about the SA z/OS monitoring agent and the Tivoli Enterprise Portal workspaces.

## [11. Migration and coexistence](#)

This unit describes the migration from and coexistence with previous releases of SA z/OS.

# Typographical conventions

In this course, the following typographical conventions are used.

| Convention                       | Usage   |
|----------------------------------|---|
| <b>Bold</b>                      | Important text is emphasized in <b>bold</b> .   |
| <i>Italics</i>                   | New terms appear in <i>italics</i> when they are defined in the text.                             |
| <b><i>Bold Italics</i></b>       | Important new terms appear in <b><i>bold italics</i></b> when they are defined in the text.       |
| Monospace                        | Code examples, output, and system messages appear in a monospace font.                            |
| UPPERCASE                        | Mainframe commands and parameters can be shown in UPPERCASE letters.                              |
| <u><i>Italics underlined</i></u> | Functions not provided by IBM Automation Control for z/OS are in <u><i>underlined italics</i></u> |

## Commonly used acronyms

Acronyms written in bold are frequently used. Acronyms written in italics are from SA z/OS or NetView.

- **Application (APL, SA z/OS resource)**
- **Application Group (APG, SA z/OS resource)**
- **Automation Manager Configuration (AMC)**
- **Automated operator (AOP, SA z/OS resource, same as Automated Operator Task, autotasks. SA uses automated function to define the autotasks for execution)**
- Automatic restart manager (ARM)
- **Automation configuration file:** (has no acronym, contains AMC, ACF, AT, MRT, MPF)
- **Automation control file (ACF):** (used by agent)
- **Automation Status File (ASF)** (used by agent)
- **Automation Table (AT,** previously also MAT)
- Availability Target (AVT) (of an APG, SA z/OS resource)
- **BCP Internal Interface (BCPi)**
- **Command Authorization Table (CAT)**
- Couple data set (CDS)
- Coupling facility (CF)
- Coupling Facility Resource Management (CFRM)
- Central processor complex (CPC)
- **Cross-system coupling facility (XCF)**
- Customer Information Control System (CICS)
- Data definition (DD) JCL statement
- Data set name (DSN)
- **Extended Multiple Console Support (EMCS)**
- Extensible Markup Language (XML)
- Family of IBM licensed programs for relational database management (Db2®)
- Generation data group (GDG)
- **Geographically Dispersed Parallel Sysplex (GDPS)**
- **Hardware Management Console (HMC)**
- IBM Tivoli Monitoring (ITM)
- **IBM Tivoli NetView for z/OS (NetView)**

- *IBM System Automation for z/OS (**SA z/OS**)*
- IBM WebSphere MQ (*formerly MQ Series*) (MQ)
- IBM Workload Scheduler (IWS or previously also TWS, OPC)
- Information Management System (IMS)
- **Initial program load (IPL)**
- **Interactive System Productivity Facility (ISPF)**
  - Job control language (JCL)
  - Job Entry Subsystem (JES)
  - Library lookaside (LLA)
  - Lifecycle reporting (LCR)
  - Logically partitioned (LPAR) mode
- **Message processing facility (MPF)**
- **Message revision table (MRT)**
- **Monitor resource (MTR, SA z/OS resource)**
  - Multi-access spool (MAS)
  - Multiple Console Support (MCS)
  - Multiple Virtual Storage (MVS)
  - *NetView-to-NetView Task (NNT)*
  - *Network resource (NTW, SA z/OS resource)*
  - OpenEdition for MVS (OMVS)
  - Partitioned data set (PDS)
  - Partitioned data set extended (PDSE)
- **Policy Data Base (PDB)**
- **Primary Automation Manager (PAM)**
  - Primary POI task (PPT)
  - Processor operations (ProcOps for short)
  - ProcOps Service Machine (PSM)
  - Program operator interface (POI) of VTAM
  - Program temporary fix (PTF)
- **Program to program interface (PPI)**
  - Resource Access Control Facility (RACF)
  - Resource Measurement Facility (RMF)

- Runtime environment (RTE) of TEMS, TEMA, OMEGAMON...
- **SA z/OS Subplex (SAplex)**
- *Satisfactory Target (SVT) (of an APG, SA z/OS resource)*
- *Secondary automation manager (SAM)*
- Short Message Services (SMS)
- Simple Object Access Protocol (SOAP), provided by TEMS
- **Status display facility (SDF) of System operations**
- **Subsystem interface (SSI)**
- Supervisor Call (SVC)
- System Authorization Facility or Security Authorization Facility (**SAF**)
- System Automation for Integrated Operations Management (SA IOM)
- System Display and Search Facility (SDSF)
- System Management Facility (SMF)
- **System operations (SysOps for short)**
- *Terminal Access Facility (TAF)*
- Time Sharing Option (TSO)
- Tivoli Enterprise Monitoring Agent (TEMA)
- Tivoli Enterprise Monitoring Server (TEMS)
- Tivoli Enterprise Portal (TEP)
- Tivoli Enterprise Portal Server (TEPS)
- UNIX® System Services (USS)
- Virtual Operator Station Task (VOST)
- Virtual Storage Access Method (VSAM)
- Virtual Telecommunications Access Method (VTAM)
- **Write to operator (WTO)**
- **Write to operator with reply (WTOR)**

# References

## **IBM System Automation for z/OS** manuals

- Customizing and Programming
- Planning and Installation
- Defining Automation Policy
- User's Guide
- Messages and Codes
- Operator's Commands
- Programmer's Reference
- Product Automation Programmer's Reference
- TWS Automation Programmer's Reference and Operator's Guide
- Service Management Unite Automation Installation and Configuration Guide
- End-to-end Automation
- Monitoring Agent Configuration and User's Guide

## **IBM Automation Control for z/OS** manual:

- Getting Started

## **IBM Tivoli NetView for z/OS** selected manuals:

- Administration Reference, SC27-2869
- Automation Guide, SC27-2846
- Command Reference Volume 1 (A-N), SC27-2847
- Command Reference Volume 2 (O-Z), SC27-2848
- Customization Guide, SC27-2849
- Installation: Configuring Additional Components, GC27-2851
- Installation: Getting Started, GI11-94
- Messages and Codes Volume 1 (AAU-DSI), GC27-2856
- Messages and Codes Volume 2 (DUI-IHS), GC27-2857
- Programming: Pipes, SC27-2859
- Programming: REXX and the NetView Command List Language, SC27-2861
- Security Reference, SC27-2863
- Troubleshooting Guide, GC27-2865
- Tuning Guide, SC27-2874
- User's Guide: NetView, SC27-2867

# Naming conventions

- This course is about the System Automation for z/OS functions as they apply to a sysplex. System Automation for z/OS supports dividing a sysplex into multiple subplexes, called **SAplex**. Whenever you see reference to a sysplex, it also applies to a SAplex.
- Whenever you see a command, its is an automation product command. Other commands are specified explicitly, like MVS command.



# 1 System Automation for z/OS overview

IBM System Automation for z/OS 4.1



## Unit 1 System Automation for z/OS overview



Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

This unit provides an introduction to using System Automation for z/OS and NetView. This unit repeats some of the contents from the architecture course like introduction to System Automation and its components, component System Operations, its architecture and its key features.

# Objectives

When you complete this unit, you can perform these tasks:

- Describe the basic components of System Automation for z/OS
  - Describe the automation architecture
  - Describe key automation and operations features
  - Describe policy structure and resource types
- Notes:
    - This unit repeats some of the contents from the architecture course
    - This course applies to IBM System Automation for z/OS (System Automation), but can be also used for IBM Automation Control for z/OS. Functions not provided by IBM Automation Control for z/OS are in *underlined italic*. The terms automation and “automation product” are used when there are no differences

## Objectives

When you complete this unit, you can perform the following tasks:

- Describe the basic components of System Automation for z/OS
- Describe the automation architecture
- Describe key automation and operations features
- Describe policy structure and resource types



**Note:** This unit repeats some of the contents from the architecture course



**Note:** This course applies to IBM System Automation for z/OS (System Automation), but can be also used for IBM Automation Control for z/OS. Functions not provided by IBM Automation Control for z/OS are in *underlined italic*. The terms automation and “automation product” are used when there are no differences

# Lesson 1 An introduction to System Automation for z/OS



## Lesson 1. Overview



Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

This lesson provides the base for the course by repeating some of the contents from the architecture course to provide introductions to System Automation and to component System Operations, its architecture its key features and to goal driven automation.

## Lesson agenda

This is a repeat from the Architecture course and covers

IBM System Automation for z/OS:

- Overview
- System Operations component
- Architecture
- Goal driven automation

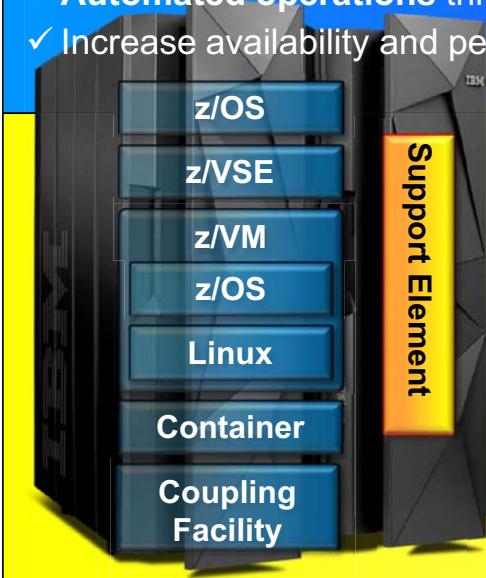
### Lesson agenda

- Overview
- System Operations component
- Architecture
- Goal driven automation

# System Automation for z/OS components

## System Operations

- ✓ Automate repetitive and complex tasks in **z/OS systems and end-to-end**
  - ✓ Helps operators to perform their tasks in a more robust way
  - ✓ Reduces z/OS specific skill requirements
- ✓ **Automated operations** through monitoring of applications and messages
- ✓ Increase availability and performance through pro-active automation



## Processor Operations

- ✓ z Systems external automation and single point of control
- ✓ **Faster LPAR startup**, recovery, shutdown
- ✓ **Save software costs** through LPAR capacity management
- ✓ **Higher availability**: Don't miss any hardware alerts
- ✓ **Save energy** with power management

© Copyright IBM Corporation 2019

1-5

*System Automation for z/OS: Components*

System Automation for z/OS has two key components that are designed to automate processor and system operation. These components are called processor operations and system operations.

## ***The system operations component***

The **system operations** component and NetView as the prerequisite automation platform, can help operators by automating many system console operations and selected operator tasks such as startup, monitoring, recovery, and shutdown of z/OS subsystems, components, applications, UNIX (USS), and sysplex resources. The system operations component can also automate operator console messages, messages in the job log, initiate timer-based actions, and prevent shortages of critical z/OS resources.

The automation product integrates with **OMEGAMON XE** to help you to increase availability and performance through proactive automation.

## ***The processor operations component***

**Processor operations** helps operators manage more systems with greater efficiency. This means that one operator, even from a remote location, can configure, initialize, monitor, shut down, and

recover a central processor complex (CPC), Logical Partitions (LPAR), and multiple systems in parallel and respond to various detected conditions. An operator, using one standard interface, can do all that across multiple types of systems such as z/OS, z/VSE, z/VM, and Linux.

Clients can save software costs through LPAR capacity management, for instance by adjusting capacity across LPARs and WLM capacity groups automatically.

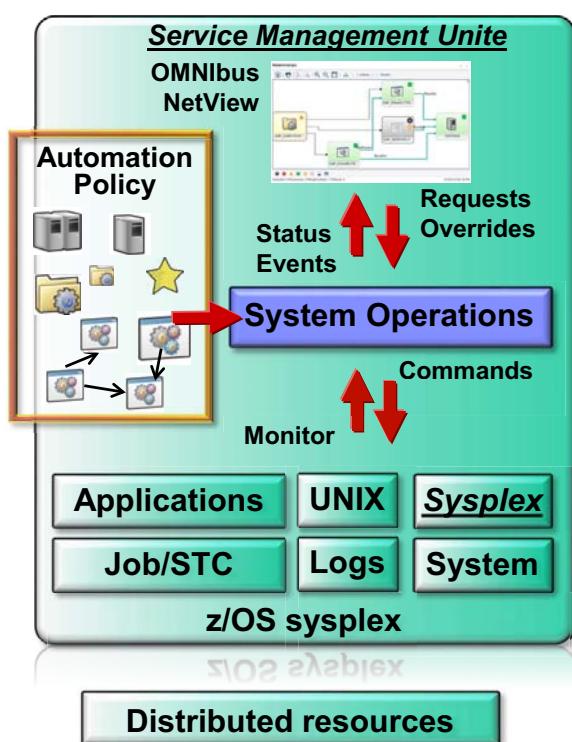
Hardware alerts and priority messages that must be responded to in a matter of seconds can be automated to increase availability.

Save energy with power management based on time of day or system load.



**Note:** The I/O Operations component is not covered as V3.5 was the last release supporting it.

# Component System Operations overview



## Policy-based and goal driven automation

- Start, recover and shutdown z/OS and distributed applications and systems using:

- Timers, events, triggers, service periods
- Dependencies and groups
- Thresholds, active and external monitors

- JES, CICS, IMS, DB2, WebSphere, OMEGAMON, UNIX, SAP, GDPS...

## Message monitoring and response

- WTO, WTOR, joblog, NetView, hardware

- Escalation using OMNIbus, email, SMS...

## Prevent outages of critical resources:

- WTO and AMRF buffers, spool, sysplex
- SYSLOG, LOGREC, SMF, dump data sets

## Easier operations at the application level

- 3270 or Service Management Unit

- Single point of control, single system image

© Copyright IBM Corporation 2019

1-6

### System operations component

The system operations component, which exploits NetView, automates many system console operations and selected operator tasks such as startup, monitoring, recovery, and shutdown of z/OS subsystems, components, applications, UNIX (USS), and sysplex resources and entire systems and sysplexes and even distributed resources (SA z/OS only).

The system operations component provides comprehensive, Plug and Play automation for mission-critical software like JES, CICS, IMS, DB2, IBM Workload Scheduler, WebSphere, OMEGAMON, UNIX, SAP, and GDPS (SA z/OS only), and many more.

Applications can be started or stopped with **timers, events, triggers, or service periods**.

**Dependencies** can be automatically established, like starting TSO after VTAM.

Resources can be members of **groups**, which can greatly reduce the complexity of automation definition and operations.

**Server groups** control which and how many group members are started. Group members can represent, for instance, application servers.

**Thresholds** can be defined to alert operators or stop recovery.

Various **active monitors** including a fast control block scanner and a monitor for UNIX resources are available.

Clients can specify their own monitor and interval. Monitor resources can determine the **health status** using external monitors or OMEGAMON.

**Messages** like write-to-operator (WTO) or WTOR, or even messages from job logs or internal CICS and IMS messages can be automated without writing scripts. Automation can be different depending on codes in the message or depending on how often the message appeared.

**Escalation** to various notification targets can be set up easily using pre-defined or user-defined alert points on a resource level to complete the following tasks:

- Create incidents in IBM Tivoli Service Request Manager
- Create events in IBM Tivoli Netcool OMNIbus
- Alarm staff using email, pager, or SMS using IBM System Automation for Integrated Operations Management (SA IOM)

The system operations component 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 spool
- z/OS dump data sets
- Action Message Retention Facility (AMRF) buffers
- Sysplex resources, for example, coupling facility (SA z/OS only)

All that you have to do is specify to system operations in an ISPF dialog what resources you want to automate and monitor, and what your policies are for automation and monitoring. The automation engine, which is identical on all systems, starts, monitors, recovers, and shuts down resources and complex applications according to your configuration and goals.

At any time, the operator can monitor and control from an enterprise-wide single point of control at the application level. You have the choice of using a 3270-based NetView console or the graphical user interfaces Service Management Unite (part of SA z/OS) or the Tivoli Enterprise Portal, which is monitoring only. When exceptions occur or goals must be changed, an operator just issues requests or overrides. SA z/OS offers a sysplex single system image, system boundaries are removed.

**Goal driven automation** greatly simplifies operations. Operators just request what they want, and automation takes care of any dependencies and resolution of affected or even conflicting goals.

# What can you automate?

- Automate messages from z/OS, applications, logs, hardware
- Prevent outages of critical resources
- Start, recover, and stop z/OS and cross platform resources
  - Started tasks (STCs) and jobs independent of scheduler
  - UNIX System Services (USS) resources
  - Cross sysplex and cross platform
- IPL, startup, and shutdown pacing
- Change system configuration like day or night shift
- Automatic or manual application move or switch in sysplex
- DB2, CICS, and IMS automation
- Escalation of problems
- Pro-active automation using integration with monitors and OMEGAMON
- Integration with scheduling
- Problem determination

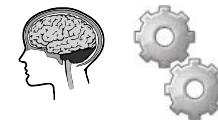
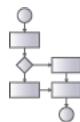
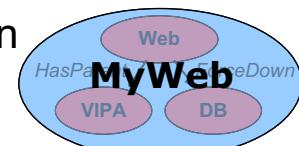
## What can you automate?

- Basic automation scenarios like automating messages from many sources, starting, recovering, and stopping of z/OS, UNIX System Services (USS) resources and even any resources across sysplexes and platforms
- Prevent outages of critical resources like MVS consoles, JES spool, logrec, SMF, page or dump data sets, couple data sets, or syslog
- Advanced automation scenarios like:
  - IPL, startup, and shutdown pacing
  - Different system configurations
  - Pro-active automation using integration with monitors
  - Automatic or manual application move or switch inside SAplex
- Escalation of problems to a status console or to an alert console
- DB2 automation
- CICS and IMS automation, even console access and internal messages and transactions
- Integration with scheduling: IBM Workload Scheduler has an automation workstation that allows

to send any command to automation, synchronously and asynchronously

## Summary of key automation technologies

- **Policy-based automation** can replace scripts
- Plug and play automation using best practices
- Powerful **sysplex-wide automation** and operation
- **Goal driven automation** to keep applications in line with business goals
- **Grouping** of resources for reduced complexity and management at business application level
- **Relationships** between resources for accelerated startup and shutdown, and correct recovery
- Manage by resource state, not by message
  - Status change triggers automation
- Monitor resources with health status and actions



Animated

© Copyright IBM Corporation 2019

1-8

### Summary of key automation technologies

**Policy-based automation** is a sophisticated methodology that allows to easily incorporate business goals into an automation framework. As the term implies, policy-based automation uses simple policy definitions specified in fill in the blanks ISPF-based panels, with no program scripting or special education required.

Policy-based automation includes resource information, groups of resources, and relationships in the decision-making process before taking action. Resource information defines resource class and name; how to start, stop, and monitor the resource; and what the preferred systems are (SA z/OS only). Resources can be members of cluster-wide (SA z/OS only) groups and can have relationships.

IBM ships **Plug and Play automation** that includes best practices automation for many z/OS applications. This can help to complete the following tasks:

- Reduce time and effort in creating a policy or updating one
- Improve automation quality

**Goal driven automation** greatly simplifies operations. Goal driven automation tries to keep applications in line with business goals specified in the policy or by the operator. Automation can take care of any dependencies by issuing start or stop commands to put other resources into the required status. It can resolve conflicting goals and even can remember goals.

**Grouping of resources** and definition of aggregate or business applications can greatly reduce the complexity of automation definition and operations.

Exploiting application groups is beneficial in many ways:

- Automation definition and operations can be greatly simplified through the grouping of resources and even business-application definitions.
- Application groups let you monitor important business applications and help verify that everything they require is available.
- Groups can make operations easier by showing the aggregated status of resources and by group actions such as startup or shutdown.
- Through groups, operators are freed from knowing the various pieces that make up an application, their dependencies, how to start or stop them, and so on.

**Relationships** between resources to achieve accelerated startup and shutdown, and correct recovery. Resources can have complex dependencies of different kinds to other resources inside and outside of the application (group) they belong to. This gives you the power to define these dependencies, so that resources get what they need, are started in the right order as quickly as possible and are shut down fast without interference.

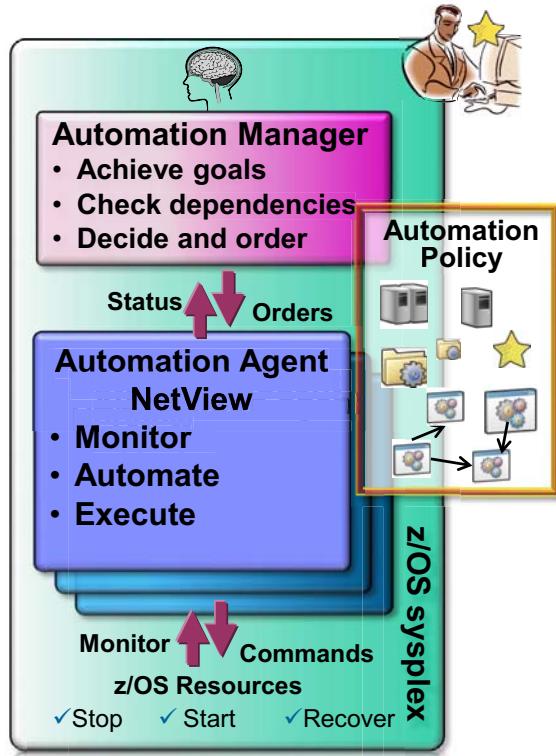
### Manage by state, not by message

Automation uses monitoring to keep track of the resource status. Although automation can use messages to update the status and also can automate messages, it is the status change that triggers automation. For example, child resources are started after the parent is up or a dependent resource is forced down when the resource it depends on terminates.

Also, messages are forgotten, but the status is preserved even across restarts and IPLs.

**Monitor resources** enable integration with any monitor and include easy to exploit OMEGAMON access to update the health status of monitor resources, which can trigger **pro-active automation**. The **health status** of monitor resources is propagated along monitor relationships to applications and is aggregated into the overall resource status.

# Architecture



- The automation policy defines the resource automation and goals ★
- Automation manager (AM)
  - Keep resource status in line with goals and dependencies by sending start and stop orders to agent
- NetView-based automation agent role:
  - Monitor and automate resources
  - Inform AM and execute AM orders
  - Operator console
- Operators
  - Control automation
  - Detect and manage problem statuses
  - Override goals

© Copyright IBM Corporation 2019

1-9

Architecture

## The automation manager

The automation manager orders the agent to load the same policy.

In the automation manager's sphere of control, it maintains status information about each resource, it controls resources, their dependencies to other resources and their desired status (goal). The automation manager uses all of this information to determine which applications to start and stop, and when to do so. The manager does not issue start and stop commands; it sends orders to automation agents which issue the start and stop commands.

## The automation agent

The automation agent runs in the NetView address space which provides the operator console. Access to the agent is through the NetView command facility. Agent commands display and allow some modification of the resource status, automation goals and status, and policy.

The automation agent uses the NetView automation table message-processing function for automation purposes. Messages are generated as the result of some event or situation. Many automation policy definitions use messages to trigger actions. A NetView automation table is the

detection mechanism for messages. The automation product creates entries in the NetView automation table for specific messages. The messages are assigned to a NetView **Autotask** to distribute workload and to ensure serialization of the messages as they are automated. Based on the type of message an action is routed to an automation operator. The action can include update of the agent status for an application or z/OS commands.

The automation agent receives orders from the automation manager and issues commands that are based on defined automation policy.

The agent sends the status changes for each application to the primary automation manager. The primary automation manager sends orders to the automation agent, requesting the start or stop of applications.

## ***The operator***

The operator role is to control automation, to detect and react on problem statuses, messages, and alerts as well as problem analysis, execution of automation change requests by managing automation, policy and resources.

Operators can use requests to override any goals.

# Key automation features

- Comprehensive automation for z/OS and UNIX applications, for example:
  - User definable **start types** and three **stop types**
  - Desired available status, startup and restart options
  - Warning and alert **thresholds** to stop recovery
  - Exception and **captured messages**
  - Built-in ASCB scan monitor reduces CPU overhead for monitoring
  - Monitoring of **UNIX** processes, files, and ports
- **Server group** with *availability* and *satisfactory targets* and member *preferences*
- Operator notification and **alerting** to SDF, SA IOM, OMNIbus, problem management, and so on
- **Runmodes** for selective startup like base, online, databases, and so on
- **Pacing gates** can prevent startup or stop of too many applications
- Non-disruptive cluster-wide policy activation
  - Synchronized with NetView automation table load
- **Automation reports** and availability and recovery time reporting
- Comprehensive **automation infrastructure** makes extension easy

## Key automation features

On this slide are the key automation features.

At the core is comprehensive automation for z/OS and UNIX applications.

Highlights include:

- User definable **start types** with multiple passes and stages that also affect message automation as well as prestart and poststart policy
- Three **stop types** with multiple passes and an INIT and FINAL phase
- **Desired available status** to initially start or stop an application or to accept its current observed status
- **Startup and restart options** like whether the application is started after any failure or only for specific ABEND codes.
- **Thresholds** can be set that let the operator know if certain errors are occurring infrequently, frequently, or have reached a critical level where the recovery process must end 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.
- **Exception and captured messages**

Special support exists for **UNIX resources** running in z/OS UNIX System Services: Monitoring of UNIX processes including UNIX user ID, command path, and filter as well as files, and ports.

Automatic or manual move of an entire application inside a system or Parallel Sysplex (SA z/OS only) is provided by **move groups** that have member preference values.

**Server groups** can represent, for instance, application servers, and control which and how many group members are started through preferences and availability and satisfactory targets. System Automation supports sysplex-wide group members.

Move groups and server groups allow to define preferred locations of a resource and what should happen when a system is unavailable or available again.

Various active monitors including a fast control block scanner and a monitor for UNIX resources are available.

**Escalation** to various notification targets can be set up easily using pre-defined or user-defined alert points on a resource level to:

- Create incidents in IBM Tivoli Service Request Manager
- Create events in IBM Tivoli Netcool OMNIbus
- Alarm staff using email, pager, or SMS using IBM System Automation for Integrated Operations Management (SA IOM)

**Runmodes** can be defined to partially start the system or to start backup applications that are normally started on another system.

The policy management can be incremental and is non-disruptive during policy activation. Policy activation is synchronized with the NetView automation table and message revision table load.

As it was covered previously, it is not mentioned here is that the system operations component provides automation procedures that enable recovery of many z/OS components and data sets.

If the resource consumption of too many applications during the start and stop phases becomes a problem, **pacing gates** can be defined to limit the number of applications of a kind that can be started or stopped at the same time.

**Automation reports** display statistical information about the automation agent and some basic information about the automation manager.

For availability and recovery time reporting automation collects and records job-related information, and writes **System Management Facility** (SMF) records at specific events in the lifetime of a resource.

Comprehensive **automation infrastructure** makes extension easy. Automation supplies commands that provide your automation procedures with a simple, standard way of interfacing with the automation control file, the automation status file, and the NetView log file.

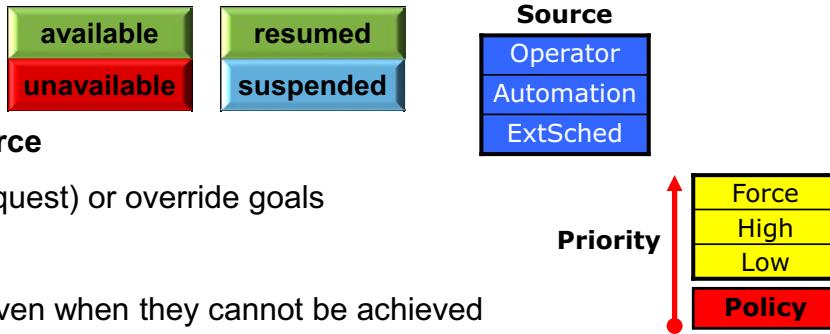
Using these commands in automation procedures provides you with the following advantages:

- Reduced development time: Less code must be written.
- Portable code: Automation policy information that is unique to an enterprise can be kept in the automation control file rather than distributed among many automation procedures. The automation procedures implement a number of different rules for handling a situation and the automation control file is used to select which rules are applicable to the current situation.

A consistent, documented interface. Messages like Write-to-operator (WTO) or WTOR or even from job logs or UNIX can be automated without writing scripts. Automation can be different depending on codes in the message or depending on how often the message appeared.

# Goal driven automation

- Availability goal
- Suspended goal
- Goals come from a **source**
- Operator can create (request) or override goals
- Goals have a **priority**
- Goals are **persistent**, even when they cannot be achieved
  - Unlike commands that are entered and forgotten
  - Cancel requests when you no longer need them
- Automation manager tries to achieve goals while taking care of dependencies
- Easier and safer operations with a single action at application group level
  - **Goals can be propagated** to group members and along active dependencies as votes putting all required resources in the desired status
  - **Conflicting votes are resolved** by priorities and request sources
- Resource status adjusted to goals, dependencies, configuration, and status



## Goal driven automation

Goal driven automation is very different from the command driven automation of other products:

- Goal driven automation can keep applications in line with business goals, dependencies, configuration, and status
- Goals are defined in the policy, can be overridden or created by the operator
- Goals are persistent
- Goals can be propagated

The administrator defines the goals for the application according to business requirements in the policy. Goals are either available (up) or unavailable (down). Application group goals control, which and how many members are started.

The operator can change or override goals and create start or stop goals by entering requests. Operator requests can have a higher priority than the Desired Available state from the policy.

Goals are persistent. They are not like commands that are run and forgotten. You must cancel requests when you no longer need them.

Operations at the application level means starting an application having several components and dependencies with one request.

- The request is propagated as votes to group members and along the dependency tree
- When multiple dependencies exist, conflicting goals can be the result. Conflicting goals are resolved. Requests have a **priority** and have a **source**. Automation uses priorities and sources when determining which request to accept. Sources might be, for example, automation scripts (**AUTOOPS**), human operators (**OPERATOR**), or **EXTERNAL**, like IBM Workload Scheduler

This process can prevent operations errors, like shutting down a resource that is needed by another application. Operators can use goal driven automation to simplify operations by entering or removing a request at the application level. They can use the automation product to manage dependencies and goals that are affected and to resolve conflicting goals.

The automation product automatically adjusts resource status to business goals, dependencies, configuration, and status.

# Lesson 2 Key operations features and user interfaces



## Lesson 2. Key operations features and user interfaces



Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

This lesson introduces the key operations features and the user interfaces available in NetView with 3270-based operator dialogs and the dynamic, highly customizable Status Display Facility as well as an overview of the graphical user interfaces are available:

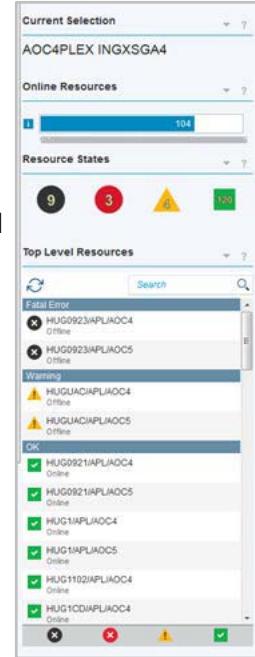
- Service Management Unite (SMU): A new way to operate your mainframe.
- The Tivoli Enterprise Portal to monitor the status of automation on z/OS systems from a browser-based interface.



**Note:** The announcement of IBM Service Management Suite for z/OS, V1.5.0 contains a statement of general direction that IBM intends to discontinue System Automation for z/OS support for collecting automation data using the Monitoring Agent and feeding that data into the Tivoli Enterprise Portal (TEP).

# Key operations features

- **Operations at the application level** lowers complexity
  - Automation takes care of dependencies and components
  - Status aggregation that includes health status
- **Goal driven automation** can reduce errors
  - Goals are not changed by IPL
- Replace consoles with a **single NetView console**
  - Dynamic **Status Display Facility** and powerful operator commands and dialogs
  - Event and status history and logs
  - **Single point of control**
  - Parallel sysplex-wide **single system image**
  - Convenient panel tailoring, sorting, and filtering capabilities
  - Automation flags to switch automation on or off
    - Scope can be system-wide down to single messages or transactions
- *Suspend and resume automation of applications*
- *System IPL complete notification*
- Web-based user interfaces
  - **Service Management Unite** (SA z/OS only)
  - Tivoli Enterprise Portal



© Copyright IBM Corporation 2019

1-13

## Key operations features

On this slide is a selection of key operations features in the automation product.

**Operations at the application level** means starting an application consisting of several components with one request.

**Goal driven automation** simplifies operations. System operators enter or remove a request at the application level. The automation product manages any dependencies and goals that are affected. It can resolve conflicting goals and handle goals even over restarts and IPLs.

## Replace your consoles with a single NetView console:

At any time, the operator can monitor and control from an enterprise-wide (SA z/OS only) single point of control at the application level. You have the choice of using a 3270-based NetView console or the graphical user interface of the Tivoli Enterprise Portal. Both are easy to use, powerful, and provide a common interface for all types of resources. When exceptions occur or goals must be changed, an operator just issues requests and overrides. SA z/OS offers a sysplex single system image, system boundaries are removed.

The single point of control provided by Automation Control for z/OS is limited to three systems in the sysplex.

The 3270-based NetView console includes the dynamic Status Display Facility (SDF) and powerful operator commands and dialogs as well as the netlog that shows all messages and WTOs that are sent to NetView.

Operators can tailor the layout of most display panels that are horizontally scrollable to pin or hide to determine order of columns and to sort by key in different directions and columns.

**Automation flags** allow to switch automation on or off, from a system-wide scope down to single messages or transactions.

**System IPL complete** notification.

**Suspend and resume resources** is a new and easy way to temporarily suspend automation for specific resources and their dependents without impacting the operations team by generating false alarms.

Operators can take resource in and out of automation easily.

# User interfaces

- Service Management Unite**
- 3270 based operator dialogs
- Status display facility (SDF)
- Tivoli Enterprise Portal

The image contains three screenshots:

- MVSAMAIN - MVSA - SYSTEM STATUS SUMMARY:** A hierarchical menu with sections: Resources (>APPLS, >GROUPS, >MONITORS), Messages (>WTOR, >MESSAGES), and Special Items (>GATEWAY). Below is a table for the INGKYST0 command, showing resource details like Type (APL/APG), System (S1/S2), and Status (PROBLEM, INHIBITED, SATISFACTORY).

| CMD Name | Type | System | Sus | Compound | Desired      | Observed    |
|----------|------|--------|-----|----------|--------------|-------------|
| A01B     | APL  | S1     |     |          | AVAILABLE    | HARDDOWN    |
| A02G     | APL  | S1     |     |          | INHIBITED    | SOFTDOWN    |
| A03B     | APL  | S1     |     |          | SATISFACTORY | AVAILABLE   |
| A04G     | APG  | S1     |     |          | INHIBITED    | AVAILABLE   |
| A05B     | APL  | S1     |     |          | DENIED       | AVAILABLE   |
| A06B     | APL  | S1     |     |          | SATISFACTORY | UNAVAILABLE |
| B01B     | APL  | S1     |     |          | INHIBITED    | AVAILABLE   |
| B02B     | APG  | S1     |     |          | SATISFACTORY | UNAVAILABLE |

- Relationships:** A diagram showing relationships between automation scripts (EAP\_BERR/APUT, EAP\_BIAPO/TEST, EAP\_BCH/APL/TE, EAP\_BPA/APL/TE) and programs (VTAM/APL/TESTM, TESTMVS) using 'Runs On' and 'Has Parent' connections.

## User interfaces

With system operations commands, you can control and maintain all of the resources sysplex-wide from a single point of control.

IBM Service Management Unite at the top is a new customizable dashboard user interface that can increase productivity of day to day operations even for expert users.

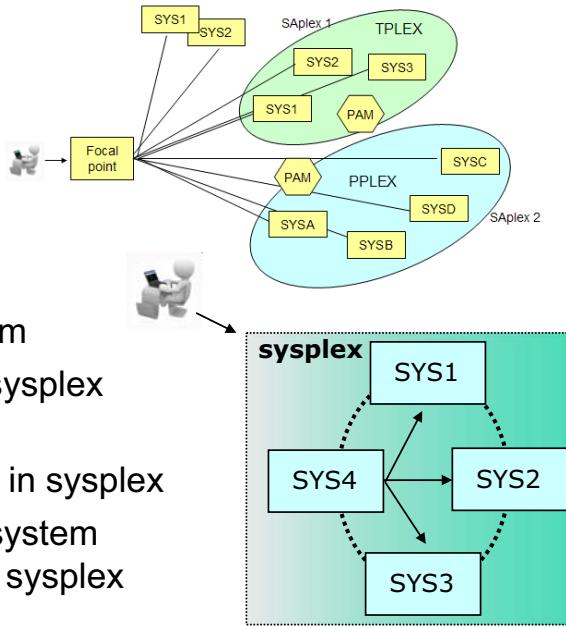
**SDF** to the left is the abbreviation of **Status Display Facility**. It consists of a set of hierarchical panels in NetView which are delivered as samples. The system administrator can also define user-specific panel layouts. If the status of an object changes, it is dynamically reflected on the panels and is propagated up in the hierarchy.

The **INGLIST command** at the bottom is available in NetView and provides details about all resources from the point of view of the automation manager. It displays information about a resource, such as statuses, flags, and schedules.

The **Tivoli Enterprise Portal** (TEP) support allows you to monitor the status of automation on z/OS systems and z/OS sysplexes (SA z/OS only) using a TEP client. On the TEP workspaces, you get an overview on the resources with all their different states. The graphical summaries are combined with the detailed tabular views.

# Single Point of Control and Single System Image

- Automation commands can be sent to any connected system in the sysplex or enterprise
  - Responses automatically routed back to user



## Single System Image

- Sysplex acts like a large single system
- Allows actions on subsystem within sysplex without specifying system name
- Host system is known on all systems in sysplex
- Selection panel displayed when subsystem runs on more than one system in the sysplex

© Copyright IBM Corporation 2019

1-15

### *Single Point of Control and Single System Image*

You can logon to any NetView and send automation commands to any connected system in the sysplex or enterprise by specifying the TARGET parameter. The responses are automatically routed back to the user. The INGLIST and DISPSTAT commands can even display resources from different systems in one panel.

**Single System Image** means that a sysplex acts like a large single system. It allows actions on subsystems within the sysplex without specifying the system name. Its host system is known on all systems in sysplex.

A selection panel is displayed when subsystem runs on more than one system in the sysplex.

# Lesson 3 Policy structure and resource types



## Lesson 3. Policy structure and resource types

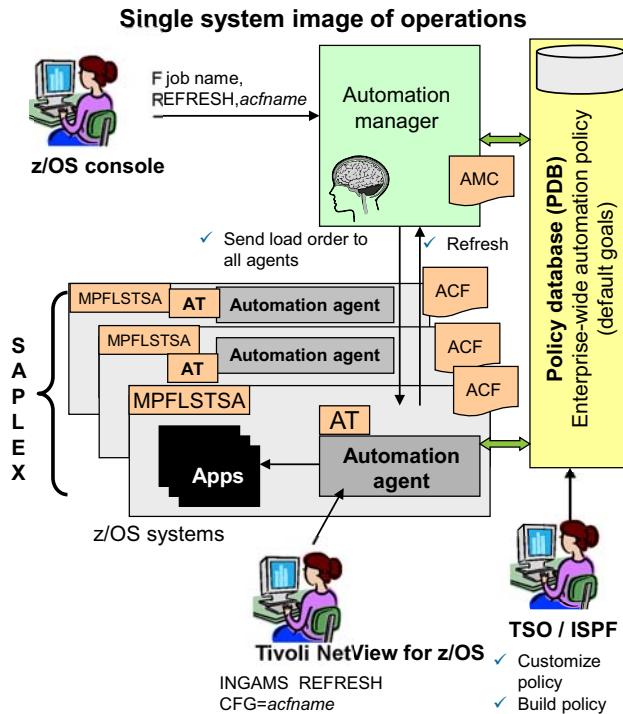


Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

This lesson introduces:

- The process from policy definition to policy build and policy activation
- The policy configuration file structure
- Operated resources

# Automation policy structure and activation



- Use the ISPF-based customization dialog to:
  - Define and customize an automation policy for systems, sysplexes, applications, groups...
- Run the build process to create these items:
  - Automation configuration file
    - Automation manager configuration (AMC) file for managers
    - Automation control file (ACF) for agents
  - Automation table (AT)
  - MPF list member
  - Message Revision Table (MRT)
- Configuration file is loaded or refreshed with these commands:
  - The **INGAMS** command
  - MVS modify command

© Copyright IBM Corporation 2019

1-17

## Automation policy structure and activation

System Automation for z/OS operation is policy-based. Automation policy is defined through the ISPF customization dialog and saved in a policy database (PDB). A single PDB can hold automation policies for all sysplexes in an enterprise.

The policy database provides the input to a build process, which creates the configuration file containing the Automation Configuration Files (ACFs). System Automation for z/OS uses the configuration file to drive automation within the SAplex. There is one configuration file for each SAplex. All control files for a SAplex must be in the same data set. A stand-alone system is treated as a monoplex.

The message processing facility (MPF) list member is generated whenever a new configuration file is built. The list member can be added to the MPF list in the z/OS parmlib data set. Use of the MRT is optional.

The automation table (AT) and message revision table (MRT) can be built for the enterprise, a sysplex, or a single system.

The configuration file is loaded or refreshed with these commands:

- The INGAMS command from a Netview command line
- The MVS modify command against PAM

## Resource Types (partial list)

| Code | Resource Type             | Resource Description  |
|------|---------------------------|---|
| APL  | Application               | A started task or other automatable entity, typically a started task                  |
| APG  | Application Group         | A group of resources that are defined and managed within the automation manager       |
| MTR  | Monitor                   | A resource that represents a monitor, sets and propagates the health status           |
| SYS  | System                    | A resource that represents a z/OS system  |
| GRP  | Group of systems          | A standard or sysplex group of systems or <i>subgroups</i>                            |
| SYG  | System group              | A generated group containing all resources that are running on a system (not an APG!) |
| REF  | <u>Resource reference</u> | Resource reference for End-to-end (E2E) automation                                    |
| DMN  | <u>Remote domains</u>     | SAplex or non-z/OS domain hosting remote resources                                    |

Naming convention is: `resource_name/resource_type/system`  
Sysplex application groups do not specify the system

### Application (APL) Resources

The Application (APL) resources represent the individual address spaces (and other entities) that are to be automated. For example, you might define Application (APL) resources for JES, VTAM, and TSO.

### Application Group (APG)

Application Group (APG) is a group of Applications (APL). It is used to manage the applications and to link them to systems. For an Application to be linked to a system, it must be a member of an Application Group that is linked to the system.

### Monitor (MTR)

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.

## System (SYS)

System (SYS) defines a single system that is to be automated. It includes definitions for automatically responding to events that happen on the system. Connect the Application Groups to it to instruct automation that it must run the resources in the APG on the System.

## Group (GRP) or SAplex

Group (GRP) is a collection of one or more systems (SYS). While Groups are required for most functions to automate systems, they are not required for the single-system automation model used by Automation Control.

For SA z/OS:

- A Group of type **SYSPLEX** contains only systems or subgroups that belong to the same XCF group, representing a logical sysplex, also called **SAplex**.
- A Group of type **STANDARD** is used for all other Groups, for example for documentation purposes or to represent a physical sysplex (a group of systems within different XCF group IDs, such as a GDPS site).
- A SubGroup Entry Type can be used for intermediary grouping of the systems within a group. Typical uses are for ProcOps or for documentation purposes.

## SYG

Indicates the group of all resources of a system. SYG is created automatically. This is the application group that contains all resources of a particular system.

## Resource reference (REF)

This entry type is used for end-to-end automation and defines a reference to a resource running on a remote domain (for example another z/OS sysplex).

## Remote domains (DMN)

This entry type defines a remote domain (for example a z/OS sysplex) where resources are running which are in the scope of end-to-end automation.

## Naming convention

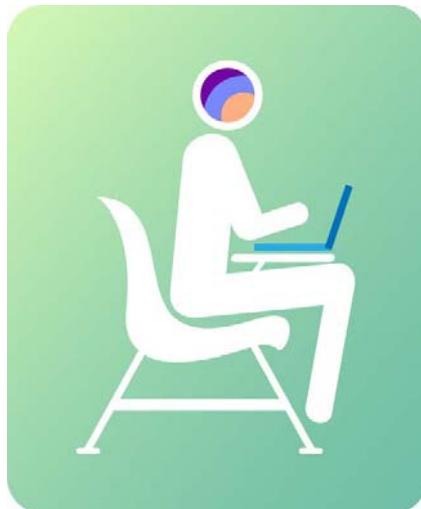
Naming convention is:

`resource_name/resource_type/system`

Sysplex application groups do not specify the system

## Student exercise Unit 1

- **Exercise 1:**  
Introduction to System Automation for z/OS interfaces



© Copyright IBM Corporation 2019

1-19

*Student exercises*

Open your *Student Exercises* book and perform the exercises for this unit.



## **Summary**

---

Now that you completed this unit, you can perform the following tasks:

- Describe the basic components of System Automation for z/OS
- Describe the automation architecture
- Describe key automation and operations features
- Describe policy structure and resource types

### *Summary*

Now that you completed this unit, you can perform the following tasks:

- Describe the basic components of System Automation for z/OS
- Describe the automation architecture
- Describe key automation and operations features
- Describe policy structure and resource types





## 2 Installation and customization

---



### Unit 2 Installation and customization



© Copyright IBM Corporation 2019  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

---

## Overview

This unit introduces the installation and customization of System Automation for z/OS, including the automation manager, automation agent, and other required z/OS components either using the traditional approach or by using the configuration assistant.

## How you will check your progress

You can check your progress in the review questions.

## References

*SC34-2716 System Automation for z/OS Version 4 Release 1 Planning and Installation*  
[https://www.ibm.com/support/knowledgecenter/SSWRCJ\\_4.1.0/com.ibm.safos.doc\\_4.1/manuals.html](https://www.ibm.com/support/knowledgecenter/SSWRCJ_4.1.0/com.ibm.safos.doc_4.1/manuals.html)

# Objectives

After completing this unit, you should be able to perform the following tasks:

- Plan for SysOps customization
- Identify the initial z/OS customization requirements
- Describe the initial configuration requirements for the OS Logger stream, VTAM, and OMVS
- Describe the initial configuration requirements for the automation manager
- Describe the initial configuration requirements for the automation agent and NetView
- Define security and authorization using NetView, System Automation for z/OS and your SAF product
- Use the **configuration assistant** to automate majority of configuration process

© Copyright IBM Corporation 2019

2-2

## Objectives

After completing this unit, you should be able to perform the following tasks:

- Plan for SysOps customization
- Identify the initial z/OS customization requirements
- Describe the initial configuration requirements for the OS Logger stream, VTAM, and OMVS
- Describe the initial configuration requirements for the automation manager
- Describe the initial configuration requirements for the automation agent and NetView
- Define security and authorization using NetView, System Automation for z/OS and your SAF product
- Use the configuration assistant to automate majority of configuration process

# Lesson 1. Planning

## Lesson 1: Planning

- Standards and naming
  - USER data set HLQ must be different from SMPE HLQ
- Upgrade installation or new installation using the configuration assistant?
- Defining the Saplex
- Data set sharing
- MPF or NetView message revision table (MRT) ?
- REXX considerations
- OMVS Setup
- Security and authorization using a System Authorization Facility (SAF) product
- Authorization of the Started Procedures

### What this lesson is about

This lesson covers planning the installation and prerequisites

### Objectives

### References

SC34-2716 System Automation for z/OS Version 4  
Release 1 Planning and Installation

Planning the installation involves:

### Standards and naming:

Decide on data set names for the

- Product libraries. Recommended is release independent
- User libraries
- USER data set HLQ must be different from SMPE HLQ
- Shared runtime libraries
- Domain specific libraries
- Parmlib member suffix

Domain names, XCF Group name, jobnames, major nodes, consoles..

For jobnames etc, maybe just use defaults or a prefix/suffix, like SA, to make it easier to identify product members, procnames, and jobnames

### Configuration assistant

Upgrade installation or new installation using the configuration assistant?

The configuration assistant is recommended.

### Defining the SAplex:

Entire sysplex or subplexes?

You can divide your real sysplex into several logical SA z/OS subplexes. To do this you must define a specific XCF group suffix and a specific group policy object for each subplex. Each SA z/OS subplex must have its own automation manager. 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.

### Data set sharing

The automation manager requires sharing of takeover and other files.

The automation configuration file must be shared SAplex-wide.

### MPF or MRT?

MRT is recommended as no PARMLIB access is required and activation is dynamic. Performance is comparable.

### REXX Considerations

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

## OMVS Setup

Because the automation manager requires OMVS, OMVS must be configured to run without JES.

## Security and Authorization

Use a System Authorization Facility (SAF) product, such as the z/OS Resource Access Control Facility (RACF) to secure your environment as follows:

- All operators, human and automated operators, are defined and authenticated by an SAF product. An OMVS segment must be created if you want to automate UNIX System Services processes.
- Command authorization is done by a SAF product that is based on the issuer of a command
- Resource authorization is done by a SAF product that is based on the issuer of particular commands

SA z/OS facilitates the steps of securing your environment. The Configuration Assistant generates the INGESAF member that is based on the input in your Configuration Options file.

## Authorization of the Started Procedures

The started procedures for the automation manager, the automation agent, the Subsystem Interface, and the IPL Data Gatherer need authority to access SAF-protected resources.

Use the STARTED class.

The user IDs of the SA z/OS started procedures require:

- Read access to product libraries.
- Update access to most user libraries.

# Prerequisites for SA z/OS 4.1

- Mandatory prerequisites
    - z/OS version 2.1 or later
    - IBM Tivoli NetView for z/OS 6.2.1 or later
  - Functional prerequisites Processor Operations
    - IBM Z (IBM z10 or later)
    - Support Element Workplace Version 2.10 (minimum)
    - Hardware Management Console Workplace Version 2.10 (minimum)
- With new IML mode for CPCs, IBM Dynamic Partition Manager (DPM):
- Support Element Workplace Version 2.13.1
  - Hardware Management Console Workplace Version 2.13.1

© Copyright IBM Corporation 2019

2-4

## Prerequisites for SA z/OS 4.1

The minimum base prerequisites for SA z/OS 4.1 are z/OS 2.1 and NetView 6.2.1.

# Lesson 2. z/OS and OMVS configuration

## Lesson 2: z/OS and OMVS configuration

- Occurs after SMP/E installation
- Includes configuration of these items:
  - PARMLIB definitions
  - System logger definitions
  - VTAM definitions
  - ARM instrumentation
  - OMVS considerations

### What this lesson is about

This lesson is an overview of the z/OS configuration requirements like PARMLIB and VTAM definitions as well as OMVS considerations.

### Objectives

After completing this lesson, you should be able to identify the initial z/OS configuration requirements for System Automation for z/OS including VTAM, OMVS, and the system logger.

### References

*SC34-2716 System Automation for z/OS Version 4 Release 1 Planning and Installation*

A good practice might be to provide a single definition that can be transported without additional customization as a set of files to other systems in the enterprise. This common definition can be

created by using z/OS system variables, NetView synonyms, and similar facilities. The data might even be located on shared disks so that several systems can use the same information. This practice assumes some groundwork by the system programmers when setting up the z/OS system. However, it is not extensive, and is probably done for other reasons.

The starting point for configuration is after the SMP/E installation of the product by the system programmer. Some or all of the actions in this section are performed by system programmers or automation administrators, according to local practice. The tasks are typically split between the two groups.

## Customize PARMLIB members

### Add SINGMODn libraries to APF, Linklist, LPA

- Define authorized libraries to the authorized program facility (APF):  
    Authorize SINGMODn (n=1,2,3) data sets
- LNKLST concatenation: Add SINGMOD1 (recommended) and SINGMOD2 (mandatory) to the concatenation
- LPALST: Add SINGMOD3 to the concatenation
- Avoid an IPL by using a PROGxx member for above steps
- Make sure that SINGMODn libraries are catalogued in the master catalog
- Make sure that no other (backlevel) NetView or SA z/OS are in linklist
- NetView requires similar definitions for (remove others):
  - SCNMLNK1, SCNMLPA, CNMLINK, SCNMLNKN, SAQNLINK
- PROGxx exit required for writing preinitialization messages to Canzlog
- Verify that REXX/370 runtime library or REXX alternate library are APF-authorized

© Copyright IBM Corporation 2019

2-6

#### Customize PARMLIB members

SA z/OS is a z/OS subsystem that operates as a privileged application. You must make certain specifications to z/OS to reflect this relationship. The *IBM System Automation for z/OS: Planning and Installation* manual provides more details.



**Note:** Make sure that the SA z/OS load libraries are cataloged in the master catalog or a user catalog identified in the LPALSTxx member, or copy the members to data sets that are in the master catalog. LNKLST data sets processed during IPL must be either cataloged in the master catalog or have their volume serials specified.

Make sure that no other (backlevel) SA z/OS or NetView libraries are in linklist

Define authorized libraries to the authorized program facility (APF). The SA z/OS SINGMODn data sets must be APF authorized. This authorization can be done in member **IEAAPFx** or you can also code a **PROGxx** member to authorize libraries. If you do this, no IPL is required.

To run SA z/OS, you must ensure that program libraries can be found at startup time. Add SINGMOD1 (recommended) and SINGMOD2 (mandatory) to the **LNKLST** concatenation. You can also code a PROGxx member to add libraries to the LNKLST concatenation.

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

NetView requires similar definitions for SCNMLNK1, SCNMLPA, CNMLINK, SCNMLNKN, SAQNLINK. The following data sets are no longer used by NetView V6R2 and can be removed if they are not being used for other reasons: SCNMUXLK, SEKGLNK1, SEKGMOD1, SEKGMOD2, SEKGSMMP1, SEZLLINK.

To enable NetView subsystem *preinitialization messages* to be written to the Canzlog log, add this statement to the PROGxx member:

```
ADD EXITNAME(CNZ_MSGTOSYSLOG) MODNAME(DSI4LCUI)
```

Verify that either the REXX/370 runtime library or the REXX alternate library are APF-authorized.

## Customize other PARMLIB members

- **IEFSSNxx:** Define subsystem name for AA and SSI  
SUBSYS SUBNAME (SYSV) INITRTN (DSI4LSIT)
- **MPFLSTxx:** Use MPFLSTSA generated by build
  - Optionally define header and trailer lines in policy
  - Concatenate with SINGSAMP(INGEMPF)
  - Alternatively use MRT
  - NetView command revision requires MPF exit DSIRVCEX
- **IEASYMxx:** Check that the following system symbols are defined:
  1. netid
  2. tcpname
- **COMMNDxx:** Define start commands at IPL (see next slide)
- **SCHEDxx:** Set NetView nonswappable, special key, and others  
As you run z/OS 2.1 or higher you can skip this step and cleanup
- **SMFPRMxx:** Add type 114 for availability reporting (optional)

© Copyright IBM Corporation 2019

2-7

### Customize other PARMLIB members

To define NetView as a subsystem, you must include an entry in the **IEFSSNxx** member that is selected at IPL time. This entry is a simple definition of a four-character string that identifies the NetView address spaces as subsystems to z/OS.

Compare the contents of the IEFSSNxx member with the INGESSN member, which resides in the SA z/OS sample library.

```
SUBSYS SUBNAME(SYSV) /* NETVIEW-SA SUBSYSTEM NAME */  
INITRTN(DSI4LSIT)
```

The *four-character prefix that you specify must match the four-character prefix of the NetView started task names*. For example, if you specify SYSV, the names of the NetView job name must be SYSVxxxx, where xxxx are any four characters you choose. If you change this four-character prefix, you can dynamically add this entry using the z/OS command SETSSI. Otherwise you must perform an IPL of z/OS to activate the change.

**MPFLSTxx** contains information that the message processing facility (MPF) uses to control message presentation, message management, message suppression, message retention, and message processing. Use MPFLSTxx to identify messages to be processed by using a message automation subsystem like NetView or by using installation-written exits.

SA z/OS builds a set of MPF definitions in member MPFLSTSA. The customization dialog also allows you to define header and trailer lines for the MPF member, thus building a complete MPFLSTxx member.

SA z/OS also provides an MPF member, INGEMPF, for predefined MPF entries. This contains the IDs of all of the messages that occur in the INGMSGSA NetView automation table that is delivered with SA z/OS. Thus if you concatenate both the INGEMPF member and the dynamically-created MPFLSTSA member, you obtain a list of all of the messages that are used in the INGMSGSA and INGMSG01 automation tables.

If you want to use the NetView command revision function, add MPF exit DSIRVCEX:

```
.CMD USEREXIT(DSIRVCEX)
```

**IEASYMxx:** Check that the following system symbols are defined:

- netid
- tcpname

For alternate definition set variable NETID or tcpname in your CNMSTGEN in your user DSIPARM:

```
NETID = xx
```

Check your NETID definition in VTAMLST or use command

```
D NET,VTAMOPTS
```

The **SCHEDxx** member contains definitions of programs that have special requirements in z/OS. If you run z/OS 2.1 or higher then you can skip this step since all automation-related components are already part of the z/OS-delivered Program Property Table.

If you have the following existing entries in SCHEDxx, remove them:

```
PPT PGMNAME(DSIMNT) NOSWAP KEY(8)
PPT PGMNAME(BNJLINTX) NOSWAP KEY(8)
PPT PGMNAME(EKGTC000) NOSWAP NOCANCEL
PPT PGMNAME(DUIFT000) NOSWAP KEY(8)
```

If you plan to use SMF records for availability reporting, you must add SMF type 114 to the **SMFPRMxx** member in PARMLIB.

NetView also can write type 37 (hardware monitor) and type 39 (session monitor) SMF records.

## Update COMMNDxx member

- Add start commands (include SUB=MSTR as appropriate)
  - NetView SSI (optional)
  - NetView (Automation agent)
  - Automation manager
  - HSAPIPLC (optional)
- Remove start commands for other applications that are managed by SA z/OS

```
COM='S AUTOMGR,TYPE=HOT,SUB=MSTR'  
COM='S AUTONETV,SUB=MSTR'  
COM='S AUTOSSI,SUB=MSTR'
```

*Update COMMNDxx member*

© Copyright IBM Corporation 2019

2-8

You start NetView, its subsystem interface (SSI) address space, and an automation manager immediately after initial program load (IPL). Starting NetView this way provides control for the start of other address spaces within the system.

You modify COMMNDxx for these purposes:

- Start NetView, the NetView SSI, and the SA z/OS automation manager
- Add a statement similar to 'S INGEAMSA,JOBNAME=AM,SUB=MSTR'
- Remove any statements for applications that are started by SA z/OS

The examples on the slide match your lab environment. Ensure that the IEFSSNxx member has a NOSTART option for the Job Entry Subsystem (JES) to prevent JESx from starting before SA z/OS during the IPL process. However if you plan to start JESx before NetView, remove the START(NO) option from your definitions in the IEFSSNxx member.

# Customizing the system logger

- (Optional) The automation manager writes history information to the z/OS system logger, and the automation agents read from it
  - Users can use the INGHIST command to view the data
- Customization includes the following steps:
  - Define and activate logger couple data sets (CDSs)
  - Define and add log structure in the CFRM policy for sysplexes
  - Ensure that managers and agents have write access to logger streams

## *Customizing the system logger*

Defining the system logger is optional. If you do not configure the logger, you cannot use INGHIST to see historical data from the automation manager. The automation manager and automation agent address spaces require authorization to update the logger streams.

You must define logger couple data sets if they are not already defined. You also must add the log structure to the z/OS CFRM policy and define the log resources that are used by SA z/OS (HSA.MESSAGE.LOG and HSA.WORKITEM.HISTORY). For buffer size, use the SA z/OS samples.

The NetView agents and all of the automation managers must be able to access the logger data and the log resources. You must define all of them to SAF for update access to CLASS(LOGSTRM) RESOURCE(HSA.MESSAGE.LOG) and CLASS(LOGSTRM) RESOURCE(HSA.WORKITEM.HISTORY). You can also define single systems as MONOPLEX to use the logger.

To disable the logger function, you write **LOGSTREAM=NO** in HSAPRMxx for the AM and in INGXINIT for the agent. If you set up a GDPS controlling system, then you must disable the logger function.

## Defining an SA z/OS agent to VTAM

- VTAM APPL definitions are required for these purposes:
  - Automation agent domain
  - Each operator that is defined in DSIOPF (one for each)
  - TAF full-screen sessions, if using OMEGAMON classic monitors
- To customize, you perform these tasks:
  - Use sample definitions in CNMSAMP(CNMS0013)
  - Set node name to domain name, such as AOFDA, or &SADOMAIN symbolic:

```
&SADOMAIN. APPL AUTH=(VPACE, ACQ, PASS), PRTCT=&SADOMAIN., X
      MODETAB=AMODETAB, DLOGMOD=DSIL6MOD, X
      APPC=YES, PARSESS=YES, X
      DMINWNL=4, DMINWNR=4, DSESPLIT=8, VPACING=10, X
      AUTOSES=2
```

These steps are already done in an existing NetView environment

**Note:** Ensure that the system automation PPT task is defined as SPO if you run a separate network NetView

```
&SADOMAIN.PPT APPL AUTH=(NVPACE, SPO), PRTCT=&SADOMAIN, EAS=1, X
      MODETAB=AMODETAB, DLOGMOD=DSILGMOD
```

© Copyright IBM Corporation 2019

2-10

### Defining an SA z/OS agent to VTAM

The agent NetView is a Virtual Telecommunications Access Method (VTAM) Application. It requires APPL statements to define it to VTAM. You use the CNMS0013 sample in CNMSAMP to create a major node definition to VTAM. Ensure that the node is also activated when VTAM is started by including it in the ATCCONxx member in VTAMLST.

Ensure that there are sufficient APPL statements. Each autotask requires one, as does each logged-on user and each remotely connected RMTCMD user. The NetView sample provides 64 APPL statements. There is no overhead in having too many statements. With VTAM, you can specify unlimited **access control blocks** (ACBs) to provide a dynamic pool so that there are always enough statements available. Use this approach if you can.

You can use system symbols in major node syntax, as shown in the following example.

**&SADOMAIN.** is a *system symbol* that you define in IEASYMxx. It is the SA NetView domain ID. Using symbolics reduces the number of definitions that you must customize.

```
*****
* NETVIEW SUBTASKS *
*****
&SADOMAIN. APPL AUTH= (NVPACE, SPO, ACQ, PASS) , PRTCT=&SADOMAIN. , EAS=4, X
      MODETAB=AMODETAB, DLOGMOD=DSILGMOD
*
      STATOPT= 'NETVIEW 069'
```

You must customize the APPL statement that refers to the domain Primary Program Task (PPT). If you run NetView for network automation, you perform these tasks:

- Define its NetView domain as the Primary Programmable Operator (PPO). The network automation then receives the unsolicited VTAM unsolicited messages.
- Define the system automation NetView domain as SPO (Secondary Programmable Operator).

## Automatic restart manager considerations

- Automatic restart manager (ARM) is a z/OS feature that can restart applications when they fail
  - ARM provides a limited subset of automation
  - ARM does not perform start, stop, or failback
- SA z/OS can restart applications
  - SA z/OS shows relationships and dependencies
  - SA z/OS performs start, shutdown, unplanned outages, and failback
- If an application is enabled for ARM, SA z/OS takes action only if ARM is unsuccessful. This requires:
  - The SA z/OS-supplied element restart exit (ERE) must be available to z/OS. The exit, AOFPERRE, is in the ING.SINGMOD2 data set. No customization is required
  - The AOFARCAT autotask must be created
  - Define MVS automatic restart management element names in policy

© Copyright IBM Corporation 2019

2-11

### Automatic restart manager considerations

Automatic restart manager (ARM) is a feature that is available with z/OS to provide a restart function for batch jobs and started tasks. ARM provides a limited subset of automation functions. If an Application is enabled for ARM, SA z/OS does not take any action against it unless ARM is unsuccessful.

If defined to ARM, the automation manager address space can be restarted by using ARM. However, a much better method of controlling the automation manager is to use a Server Group.

If an application is enabled for ARM, SA z/OS takes action only if ARM is unsuccessful. This requires:

- The SA z/OS-supplied element restart exit (ERE) must be available to z/OS. The exit, AOFPERRE, is in the ING.SINGMOD2 data set. No customization is required.
- The AOFARCAT autotask must be created. The autotask name is included in the AOFOPF member and is created automatically by NetView if you install SA z/OS without changing AOFOPF.
- Define MVS automatic restart management element names in policy.

## Customizing OMVS

- Automation manager must be **superuser** to block OMVS
- Because the automation manager requires OMVS, OMVS must be configured to run **without JES**
  - If z/OS UNIX System Services requires colony address spaces, JES must be started first

Definitions in the BPXPRMxx member must be either

- All FILESYSTYPE specifications with an ASNAME parameter are moved into a separate BPXPRM member activated by automation for message  
`BPXI004I OMVS INITIALIZATION COMPLETE`
- Or add the parameter 'SUB=MSTR' to all ASNAME definitions  
`FILESYSTYPE TYPE (ZFS) ENTRYPOINT (IOEFSCM)  
ASNAME (ZFS, ' SUB=MSTR' )`

### Customizing OMVS

The automation manager is written in C++. You should add an OMVS segment to the RACF profile for the automation manager and make the profile a superuser to allow it to block shutdown of OMVS. z/OS UNIX System Services must be running before automation manager can initialize.

A deadlock can occur during initialization if you have to create colony address spaces for use by a Physical File System (PFS) such as Network File System (NFS). On this slide, you see a solution to this problem.

The definitions in the BPXPRMxx member must match one of the following:

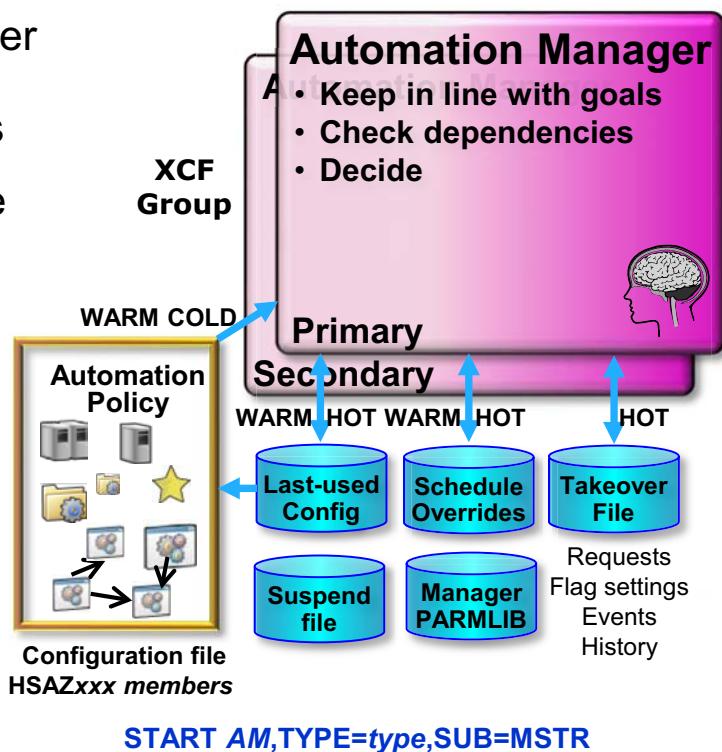
1. Either all FILESYSTYPE specifications with an ASNAME parameter are moved into a separate BPXPRM member. This can be activated via the automation policy by using the SETOMVS command after the message `BPXI004I OMVS INITIALIZATION COMPLETE` has been received.
2. Alternatively, add the parameter 'SUB=MSTR' to all ASNAME definitions that are not being moved to a separate member in the action listed above.

More details are available in the *IBM System Automation for z/OS: Planning and Installation* manual.

# Lesson 3. Configuring automation manager

## Lesson 3: Configuring automation manager

- The automation manager requires sharing of takeover and other files
- Use the start procedure default is INGEAMSA
- Configure the HSAPRMxx automation manager PARMLIB member



© Copyright IBM Corporation 2019

2-13

### What this lesson is about

This lesson provides an overview of the System Automation for z/OS automation manager configuration.

### Objectives

After completing this lesson, you should be able to configure the automation manager started task.

### References

*SC34-2716 System Automation for z/OS Version 4 Release 1 Planning and Installation*

Samples are provided in SINGSAM for members INGEAMSA and HSAPRMxx.

## Automation manager start procedure as generated by configuration assistant

```
//AUTOMGR PROC TYPE=HOT,  
// D=0,  
// Default start delay is none  
// M=SA,  
// Default parmlib suffix is 00  
// P=NO,  
// Default is NO prompting  
// SMP/E installed target lib  
// HLQINST=SA410A,  
// HLQ of shared and unique AM  
// HLQ=SAZOS.USER,  
// System where the Automation  
// SYSNAME=&SYSNAME.,  
// 2nd-level qualifier for  
// automation manager datasets  
// SLQ=AM,  
// Auxiliary qualifier for  
// automation manager data sets.  
// SLQAUX=AM1,  
// Auxiliary qualifier for  
// automation manager trace  
// datasets for debugging purposes.  
// AM1 - for first AM and  
// AM2 - for second AM  
// on the system  
// GRPID=SA,  
// Groupid as part of the name of  
// shared AM datasets  
// HLQCLB=CBC,  
// Automation manager datasets  
// Default C/C++ library HLQ &  
// Default LE/390 library HLQ
```

© Copyright IBM Corporation 2019

2-14

### Automation Manager start procedure

This slide illustrates the sample JCL. All instances of the automation manager can have the same JCL across the sysplex group. Remember that several instances of the automation manager can run within a sysplex group, and even within a single system. Different instances of the automation manager can run on the same system with the same JCL by using a command that specifies different job names:

```
S INGEAMSA, JOBNAME=AM, SUB=MSTR
```

Some changes to the JCL for dump, SYSOUT, and trace data sets are required to support multiple AMs on the same system. For details, see the INGALLCx sample JOBS as described in the *IBM System Automation for z/OS Planning and Installation* manual.

**Always use TYPE=HOT**, except for the first time you start the automation manager with a new Configuration File and empty Takeover File. In this case, HOT is automatically degraded to TYPE=WARM with operator notification. Data sets that are used for restart purposes must be shared among all automation managers.

The automation manager must be active in the sysplex group before any Application can be started, because the automation manager issues all orders to start an Application. Ensure that it is

startable in the first system before JES is running and a SYSOUT data set is defined for any DD statements.

CEEDUMP DD is used by language environment dump services and must be a sequential data set. There are two TRACE DD statements, TRACET0 and TRACET1, that can be used if automation manager traces are required. You define the automation manager jobs in SAF to have these items:

- Access to the OMVS segment as UID0
- Read access for the HSAPLIB data set
- Update access to the log streams
- Update access to the following sets and files:
  - Takeover file
  - SUSPEND file
  - Trace data sets
  - Schedule override file
  - Configuration information file (HSACFGIN statement in start JCL)
  - SYSOUT and dump data sets
- Read access to the automation configuration file

## Automation manager start procedure (continued)

- All three of these data sets must be shared across all systems
  - Also takeover, suspend, and automation configuration files
- HSAPLIB contains HSAPRMxx member
- First start must be a cold start

```
//AMSA      EXEC PGM=HSAPINIT,REGION=0M,TIME=1440,          X
//                  PARM='MEMBER=&M,START=&TYPE,DELAY=&D,PROMPT=&P'
//*
//HSAMODLE DD  DSN=&HLQINST..SINGMOD1,DISP=SHR      SA z/OS
//            DD  DSN=&HLQCLB..SCLBDLL,DISP=SHR      C/C++
//            DD  DSN=&HLQCEE..SCEERUN,DISP=SHR      LE/390
//            DD  DSN=&HLQCEE..SCEERUN2,DISP=SHR      LE/390
//HSAPLIB  DD  DSN=&HLQ..PARMLIB,DISP=SHR
//HSAOVR   DD  DSN=&HLQ..INGXSG&GRPID..HSAAMOVR,DISP=SHR
//HSACFGIN DD  DSN=&HLQ..INGXSG&GRPID..SHSACFGO,DISP=SHR
//SYSOUT   DD  DSN=&HLQ..&SYSNAME..&SLQ..SYSOUT,DISP=OLD
//SYSPRINT DD  DSN=&HLQ..&SYSNAME..&SLQ..SYSPRINT,DISP=SHR
//CEEDUMP  DD  DSN=&HLQ..&SYSNAME..&SLQ..CEEDUMP,DISP=SHR
//TRACETO  DD  DSN=&HLQ..&SYSNAME..&SLQ..TRACETO,DISP=SHR
//TRACET1  DD  DSN=&HLQ..&SYSNAME..&SLQ..TRACET1,DISP=SHR
```

### Automation manager start procedure (continued)

The names of the takeover, automation configuration files is defined in HSAPRMxx. All automation manager instances in the SAplex use the same HSACFGIN, HSAOVR, takeover , and configuration file. All instances might also use the same HSAPLIB definition.

The following three data sets (and more) must be shared:

- The HSACFGIN file is a configuration information file in which the automation manager maintains information about the configuration files that are used. This file is read on automation manager WARM and HOT starts to determine the previously used set of control files. A HOT start reads the data from the Takeover File. A WARM start reads the data from HSACFGIN. A COLD start reads the HSACFGIN only if it is degraded from a WARM or HOT start. Typically, a COLD start is only done the first time that the manager initializes. During a COLD start, the configuration file defined in HSAPRMxx is read.
- The HSAOVR is a schedule override file in which the automation manager maintains information about overrides to Service Periods (schedules). This information is cleared if AM is started TYPE=COLD.
- The HSAPLIB contains the HSAPRMxx members that are read when the automation manager address space is started.

# Automation manager HSAPRMxx member

```
CFGDSN=SAZOS.ACFS.OPSU1EX1
TAKEOVERFILE=SAZOS.USER.INGXSGSA.TAKEOVER
GRPID=SA
BLOCKOMVS=YES
COMM=XCF
DELAY=0
LOGSTREAM=NO
TAKEOVERTIMEOUT=12
SUSPENDFILE=SAZOS.USER.SUSPEND
```

- **DELAY:** Specifies how many seconds the automation manager waits before determining the operational mode
- **GRPID:** Specifies the two-character suffix composing the XCF group name that is used by the automation manager and the various agents when communicating among each other. Default is blank.
- **COMM:** Specifies the communication method between the automation manager and automation agent
- **TAKEOVERFILE:** Is the fully qualified name of the TAKEOVERFILE that contains the persistent data store
- **CFGDSN:** Is the name of the data set that contains the automation configuration file (SOCNTL) that is read by the automation agent and automation manager

## Automation manager HSAPRMxx member

HSAPRMxx is placed in the data set that is defined in the HSAPLIB definition. You copy HSAPRM00 from sample library SINGSAMP to your PARMLIB and edit the values.

All HSAPRM00 parameters are described in *IBM System Automation for z/OS: Planning and Installation Guide*. The following parameters are important:

- **DELAY:** Indicates how long this automation manager waits before attempting to become the PAM. By setting some instances with a long value, you ensure that during a sysplex IPL the Primary automation manager runs on the system you want. A value of zero (0) means that this manager is the primary one, if none is there. This setting can be overridden by a prompting response. A value of zero (0) means it does not wait. If there is a race condition with other AM instances, this one is favored.
- **GRPID:** Is the suffix used to identify the sysplex group. The GRPID defined in HSAPRMxx must match the GRPID defined in INGXINIT for the agent. A null value is the default one.
- **PROMPT:** If set to YES, the operator can override the name of the configuration data set to be used (see CFGDSN).

- **TAKEOVERTIMEOUT:** Defines how long, in seconds, an automation manager waits before it performs a HOT start. If communication is XCF, the AM waits the specified seconds before the takeover is performed from the takeover file. This delay might be required to for VSAM to perform its cleanup activities on the takeover file. The default setting is 12 seconds.
- **TAKEOVERFILE:** Defines the data set name of the Takeover File. It must be fully qualified.
- **BUILDTIMEOUT:** Provides a time limit for the process of building internal data structures when a COLD or WARM start is performed.
- **CFGDSN:** Indicates the data set containing the automation manager configuration files when a COLD start is performed. The data set that is used by the AM can be overridden by a prompt when the AM is started.
- **OVRDELETEDELAY:** Indicates how long schedule overrides are kept after their expiry before being deleted from the shared data set.
- **STOPDELAY:** Provides a delay interval used when a DEFER STOP is issued for the automation manager.
- **LOGSTREAM:** Defines whether or not the automation manager establishes a connection to the system logger at initialization time. This value must match the LOGSTREAM defined in INGXINIT for the agent. The default setting is YES.
- **IOINTERVAL:** Defines the interval that is used to buffer any I/O to the Takeover File. The value can be from zero to ten seconds. The default is zero, which means that no buffering is performed. The maximum is ten seconds. At the end of the interval, any deferred I/O is performed.
- **BLOCKOMVS:** Specifies whether the automation manager blocks the shutdown of OMVS when the automation manager is active. Specify YES or NO.
- **SUSPENDFILE:** Name of the suspend data set which contains the SA resources that should be suspended by the SA automation manager after loading or refreshing the configuration data set.

HSAPRMxx members can use system symbolics; for example, &SYSCLONE. Specifying COMM=XCF indicates that the cross-system coupling facility (XCF) is used for communication. These parameters can be specified by using the start procedure for the automation manager. The start procedure specifications override the HSAPRMxx member parameters.

## Running HSAPIPLC at IPL

- HSAPIPLC procedure stores IPL data in the IPLDATA file
  - Can be viewed by using the INGPLEX command in NetView:  
INGPLEX option 6 (Display IPL information)
  - Can be shared to allow comparing IPL data
  - Requires HSAIPL DD statement in the NetView JCL:  
HSAIPL DD DSN=&VQ2..INGXSG&GRPID..IPLDATA,DISP=SHR
- Runs once per IPL
- Alternatively, you can define HSAPIPLC in your automation policy with Restart after IPL = START
  - See SYSVIPLC in \*BASE PDB for an example
  - Starting from \*BASE is the preferred way
- Start from COMMNDxx member:
  - COM='S HSAPIPLC,SUB=MSTR'

© Copyright IBM Corporation 2019

2-17

### Running HSAPIPLC at IPL

The slide shows an HSAIPL DD, which could be shared among the systems within a SAplex. This will allow comparing IPL data between two systems. There is no requirement to have the file system specific.

The file name uses the XCF group ID. The example shows an HSAIPL DD that can be shared across systems, SAplexes or sysplexes. The HSAIPL DD statement corresponds to the HSAPIPL DDname in the NetView start procedure.

# Lesson 4. Automation agent and NetView configuration

## Lesson 4: Automation agent and NetView configuration

- NetView start procedure for the automation agent address space
- NetView subsystem interface (SSI) start procedure
- DSIPARM definitions
- SA z/OS communications task
- EMCS consoles under NetView
- REXX function packages

### What this lesson is about

This lesson provides an overview of the System Automation for z/OS automation agent started task.

### Objectives

After completing this lesson, you should be able to configure the System Automation for z/OS agent as a NetView started task.

### References

*SC34-2716 System Automation for z/OS Version 4 Release 1 Planning and Installation*

Three address spaces are required by SA z/OS:

- SA z/OS automation manager

- NetView (or SA z/OS automation agent)
- NetView subsystem interface, known as the NetView SSI

This lesson focuses on the two NetView address spaces that are required to run SA z/OS and the related configuration requirements. The SA z/OS agent runs under NetView and uses NetView facilities to capture system events.

## NetView start procedure example as generated by configuration assistant

```
//AUTONETV PROC PROG=DSIMNT,  
//  
//          DOMAIN=&COURSE.,           ** NETVIEW DOMAIN NAME  
//  
//          GRPID=SA,              ** GROUPID AS PART OF THE NAME OF  
//                                SA PLEX SHARED DATASETS  
//  
//          Q1=SAZOS.USER,         ** USER DSN HIGH LEVEL QUALIFIER  
//  
//          SQ1=NETV621B,          ** NETVIEW DSN HIGH LVL QUALIFIER  
//  
//          SQ2=SA410A,            ** SA z/OS DSN HIGH LVL QUALIFIER  
//  
//          VQ1=SAZOS.USER,         ** VSAM DSN     HIGH LVL QUALIFIER  
//  
//          VQ2=SAZOS.USER,         ** SA z/OS DSN HIGH LVL QUALIFIER-VSAM  
//  
//          /* Comment out the following symbolic if starting NetView before JES  
//          SOUTA='*',             ** Default printed output class  
//          REG=0,                  ** Region Size(Im MB) for NetView  
//          SUBSYM='',              ** Symbolic substitution switch  
//          NV2I='',                ** UNIQUE AID  A two character value  
//          TRSIZE=''               ** INTERNAL TRACE TABLE SIZE - Number  
//NETVIEW  EXEC PGM=&PROG,TIME=1440,  
//          REGION=&REG.M,  
//          PARM=(24K,200,  
//                  '&DOMAIN','&SUBSYM','&NV2I','&TRSIZE'),  
//          DPRTY=(13,13)
```

- PROG=DSIMNT: Automation NetView only
- PROG=BNJLINTX: Network and automation NetView
- SUBSYM=\*SUBSYM: z/OS system variables permitted in NetView (def.)

© Copyright IBM Corporation 2019

2-19

### NetView start procedure example

A SA z/OS sample JCL for NetView (INGENVSA) is shown on the slide and described in the following list:

- **PGM:** The program name is DSIMNT (the NetView main task) or BNJLINTX. If you use a single NetView for networking automation and system automation, the program name is BNJLINTX.
- **REG:** You specify a large region by using REGION=0 if you can, or setting as high as possible. A value of 100M is reasonable and provides space for message queuing and REXX programs. You can use the RES, TASKUTIL, and TASKMON commands to monitor the region size.
- **PARM:** The third and fourth operands illustrate the ability to specify the domain name and domain password in the JCL, rather than in the CNMSTGEN member. The JCL can be run on any system.
- The fifth operand defaults to \*NOARM, indicating that NetView is not to be started by the z/OS Automatic Restart Manager (ARM). If you want the restart to be automatic, specify \*ARM. NetView then generates a default ARM token when registering with z/OS. ARM means that if the NetView address space fails, it can be restarted automatically by z/OS to reduce outage time.

- The sixth operand defaults to \*SUBSYM, indicating that symbolic substitution (by using the z/OS system variables) within NetView definitions is supported.

## NetView STEPLIB

- **STEPLIB is not recommended** for performance reasons as each module is searched in STEPLIB first
- All STEPLIB data sets must be APF-authorized
  - Use PROGxx
- For performance reasons, you include the SINGMOD1 and SINGMOD2 data sets in the linklist concatenation of your system

© Copyright IBM Corporation 2019

2-20

### NetView STEPLIB

The SINGMOD $n$  libraries must be included in the linklist by using PROGxx or LNKLSTxx. Data sets that are not included in the linklist can be added to the STEPLIB concatenation (authorized in PROGxx or IEAAPFxx). Any module that is accessed must be authorized in APF (authorized program facility). If a data set is not authorized, NetView fails with an 047 ABEND when it tries to initialize. Use the PROGxx technique, with the SET PROG=xx command for dynamic updates. This technique is useful when you are testing.



**Note:** STEPLIB is not recommended for performance reasons as each module is searched in STEPLIB first generating I/Os

## NetView DSICLD and DSIPARM

- DSICLD points to REXX routines:

|                                       |                 |
|---------------------------------------|-----------------|
| //DSICLD DD DSN=&Q1..CNMCLST,DISP=SHR | USER TARGET     |
| // DD DSN=&SQ2..SINGNREX,DISP=SHR     | SA TARGET       |
| // DD DSN=&SQ1..SAQNCLST,DISP=SHR     | NETVIEW TARGET  |
| // DD DSN=&SQ1..CNMCLST,DISP=SHR      | NETVIEW TARGET  |
| // DD DSN=&SQ1..CNMSAMP,DISP=SHR      | NETVIEW SAMPLIB |

- DSIPARM points to NetView definition members, for example, CNMSTYLE:

|   |                |
|---|----------------|
| //DSIPARM DD DSN=&Q1..&DOMAIN..DSIPARM,DISP=SHR | USER TARGET    |
| // DD DSN=&Q1..DSIPARM,DISP=SHR                 | USER TARGET    |
| // DD DSN=&SQ2..SINGNPARM,DISP=SHR              | SA TARGET      |
| // DD DSN=&SQ1..SAQNPARM,DISP=SHR               | NETVIEW TARGET |
| // DD DSN=&SQ1..DSIPARM,DISP=SHR                | NETVIEW TARGET |

Example: &SQ2 might be SA410A

Example: &SQ1 might be NETV621B

The DSICLD statement defines libraries for REXX EXECs and NetView command list (CLIST) programs. The first data set CNMCLST represents user-written programs that are used in all systems within the enterprise. There might be also a domain specific one for rare exceptions only.

The **SINGNREX** data set contains the REXX EXECs that are used for SA z/OS. It is concatenated ahead of the NetView CNMCLST data set.

SA z/OS uses a large amount of REXX programming. Because of the modular nature of the programming, many levels of called REXX programs are common within the SA z/OS environment. At any time, many REXX environments might be required. You must increase the standard number included with the TSO/E feature to at least 1000. If you do not, REXX programs might fail in the NetView Application address space. The sample is in SYS1.SAMPLIB(IRXTSMPE), and is a simple macro specification. Your system programmer can use SMP/E to manage this change.

If you have the REXX compiler installed in your installation and you have the runtime libraries available on all systems where SA z/OS runs, you can optionally compile the distributed REXX programs for better performance.

The DSIPARM statement defines libraries that contain definitions, which are used by many different parts of NetView. The statement is analogous to the SYS1.PARMLIB library that is used by many different components of z/OS.

Standard definitions for NetView operation are provided in the NetView data set NETVIEW.V5RxM0.DSIPARM. You can find standard definitions for SA z/OS in SINGNPRM. Many of the definitions in the user libraries are overrides of the standard definitions. The user libraries must be first in the list.

You might need domain-specific overrides to the definitions. For example, each domain might use a unique beginning panel for SDF. The panels are located in the domain-specific DSIPARM to identify the initial SDF panel.

Other standard data sets might require similar considerations, like the following ones:

- CNMPNL1 for panels
- CNMMSG for messages
- DSIPRF for operator profiles

In each case, you might want to use one or more user data sets first in the DD concatenation.

## NetView data sets

- Use single extent. Multiple extents are not recommended as NetView can not read beyond the first extent
  - Most installations use a PDSE
  - Use the NetView REACC command to free and reaccess data sets
- Allocate extra directory space if you have many members
- Make the DSILIST data set large, with sample directory space
- Secure SMP/E libraries as read-only
- Do not add the configuration file data set into the DSIPARM concatenation chain

The configuration file is allocated dynamically
- Include the Netview SCNMLNKN data set in the LNKLST concatenation to make the module DSIPHONE available in the TSO/ISPF environment

© Copyright IBM Corporation 2019

2-22

### NetView data sets

All NetView data sets should be a single large extent. If you specify the libraries with multiple extents, members might become inaccessible to the running NetView SA z/OS. Consider creating a regular job to compress all the PDS; however, remember to specify DISP=SHR. Most installations use a PDSE library.

Use the NetView REACC command so that data sets can be freed and reaccessed, thereby accessing any new extents.

Allocate extra directory space if you have many members, like numerous SDF panel definitions or many REXX scripts.

IBM and other vendor data sets must be read-only. Change only your user data sets.

Ensure that the block size of your user data sets is at least as big as the biggest block size of any other data set in the concatenation.



**Note:** Use PDSE in a sysplex environment only.

## Additional SA z/OS data sets

- Data sets are allocated with sample jobs INGALLC $n$  in SINGSAMP
- Automation Status File (ASF) holds records for subsystem status or errors for thresholds
- SINGSAMP(INGREORG) is a sample job for doing a REORG on the status file
- You use a password file for OMEGAMON sessions, if not already allocated to NetView
- The IPLDATA and INGDUMP data sets are included

© Copyright IBM Corporation 2019

2-23

### Additional SA z/OS data sets

For SA z/OS, you must allocate additional data sets by using several sample jobs, INGALLC $n$ , in SINGSAMP. The INGALLC $n$  jobs are documented in the *IBM System Automation for z/OS Planning and Installation* guide.

The Status File is required. The Status File is a VSAM file that stores status of subsystems for use across IPLs. It also contains error statistics, which are used when determining thresholds, up to ten per subsystem.

IPLDATA is a VSAM file used to store IPL information. HSAIPLC populates this data set. INGDUMP is a sequential file used by the automation agent to store dump information. Include these files in your space estimates. Edit the sample jobs to create these files. You can also define the number of collected IPLs.

## NetView SSI start procedure example

```
//AUTOSSI PROC PROG=CNMINIT,
//           REG=16600,          ** REGION SIZE (IN K)
//           MBUF='RETIRED',   ** (Retired) message buffers
//           CBUF='RETIRED',   ** (Retired) command buffers
//           DSIG='RETIRED',   ** (Retired) command designator
///* See MVSPARM.CMD.DESIGNATOR in CNMSTYLE for command designator info
//   MSGIFAC='RETIRED',   ** (Retired) message interface setting
//   PPIOPT='PPI',        ** PPI OPTIONS SWITCH
//   ARM='*NOARM',        ** AUTOMATIC RESTART (ARM) USAGE
...

```

- ARM=\*NOARM: Replace with \*ARM or ARM ID for restart by z/OS
- PPIOPT=PPI: Needed for many options, such as IBM Workload Scheduler, ARM, or EIF

Note: The PPI must belong to the SA z/OS NetView SSI, not the network NetView SSI

### NetView SSI start procedure example

Each NetView can have an optional subsystem address space associated with it. This address space is required for unsolicited messages from z/OS as this address space interfaces to the z/OS subsystem interface (SSI). The Extended MCS interface is used for solicited messages of command responses. The SSI is also required under the following circumstances:

- Entering commands from real MCS consoles by using a command designator such as %DISPSTAT

DSIG specifies up to eight characters to be used as the NetView subsystem command designator. It must precede all NetView commands and command lists that are issued from an MVS console to distinguish them from other z/OS commands. The default value of null causes the subsystem to use the four-character subsystem name as the prefix. Each SSI must have a unique command designator.

You can register the prefix with the z/OS system on which the job runs or with the sysplex with the PFXREG option.

- Completing recovery when a NetView address space is restarted by ARM

Typically, the subsystem address space is always configured. Use the sample JCL CNMSJ010 supplied by NetView. The first two parameters specify the maximum number of messages and commands that are queued. If EMCS consoles are used for the unsolicited message stream, the numbers required do not have to be high, because the message and command traffic is direct to NetView.

The MSGIFAC specified in the start procedure *must match* the MVSPARM.MSGIFAC statement in CNMSTYLE. If it does not, an error message is issued. The default supplied by NetView is MSGIFAC=SSIEXT.

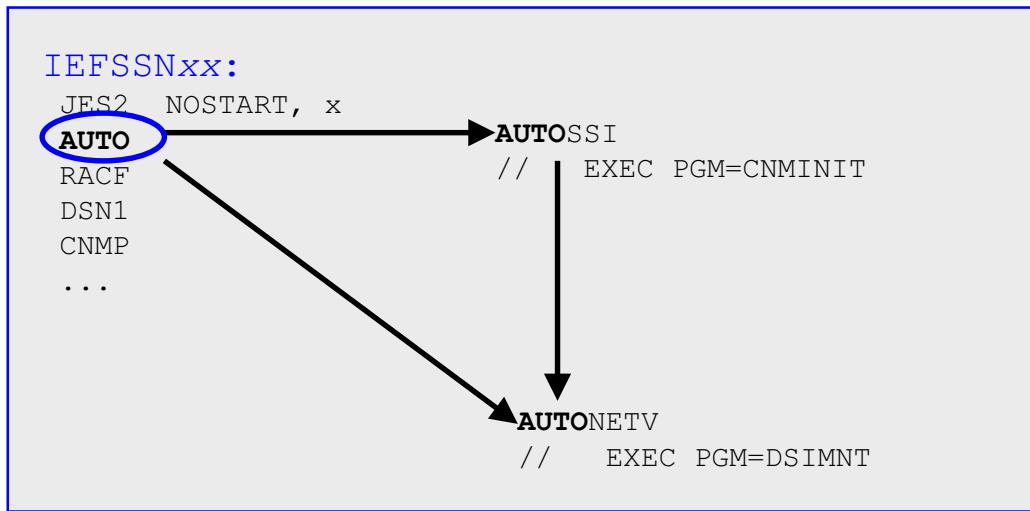


**Note:** In NetView V6.1 and later, several options in SSI JCL are retired such as MSGIFAC, DSIG and others. MSGIFAC and other definitions for the SSI have been moved to the CNMSTYLE member

The PPI option indicates that this NetView subsystem manages the NetView program-to-program interface (PPI). With PPI, communication can occur between NetView programs and other address spaces in the same z/OS image. This PPI is used by several products, such as the Tivoli Workload Scheduler interface and ARM notification. Other vendor software products also use it, and it can be used between programs that are not NetView programs. The PPI is also used by the NetView Event Automation Service (E/AS) which is required for end-to-end automation.

If you have two NetView SSI address spaces in your z/OS image, only one can control the NetView PPI. Ensure that the SSI address space that you choose meets all of your availability requirements. If both address spaces have PPI specified, the first active SSI address space controls it.

## Subsystem name table



- You define a four-character subsystem name, such as AUTO
- Start procedures for NetView and SSI address spaces must begin with the same four characters:  
AUTONETV and AUTOSSI

© Copyright IBM Corporation 2019

2-25

### Subsystem name table

This example shows how the definition of subsystems in the IEFSSNxx member matches the started task names of the two NetView procedures. Each NetView address space recognizes its complementary address space by the four-character prefix. Because the prefix is defined in IEFSSNxx, it can use the SSI and is treated by z/OS as a z/OS subsystem. The automation manager does not rely on subsystem names in any way.



**Note:** The INGESSN sample in SINGSAM uses a subsystem name of INGE, which matches the SA z/OS sample procedures.

## DSIPARM definitions

- Follow instructions in SINGSAMP(INGSTGEN) to enable SA and SYSOPS towers (do not copy INGSTGEN)
  - TOWER = SA
  - TOWER.SA = SYSOPS
- These other members might require changes for SA z/OS
  - CNMSTGEN: Style sheet overrides
  - INGXINIT: Define communications task parameters
  - CNMCMDU: SETTIMER command
  - AOFOOPFGW: Gateway autotasks
  - INGESCAT: Command authorization table
    - Group NVOPS2: Add your NetView administrators
  - User Automation Tables INGMSGU1, INGMSGU2, own synonyms

© Copyright IBM Corporation 2019

2-26

### DSIPARM definitions

The NetView *style sheet* (CNMSTYLE) provides a simplified way of making definitions to NetView. The style sheet reduces the amount of editing of DSIPARM members (and other data sets). SA z/OS provides a sample style sheet member, INGSTGEN, in SINGSAMP. You never update CNMSTYLE directly. Instead, you incorporate the INGSTGEN content in the active CNMSTGEN. Do not copy INGSTGEN.

With style sheets, many of the definitions that are required for SA z/OS are enabled automatically by specifying **TOWER=SA** and **TOWER.SA = SYSOPS** in the CNMSTYLE definition. Be aware that **TOWER.SA = SYSOPS** is specified in the default NetView CNMSTYLE sample.

When editing DSIPARM and SINGNPRM members, copy them from the SMP/E libraries to a user data set and edit the members in the user data set. Concatenate the user data set before the IBM data sets. For example, copy NetView sample CNMSTGEN to *USER.DSIPARM* and customize the TOWER statement.

By default, SETTIMER starts the NetView TIMER command. If you want the SA z/OS timer functions instead, you must customize command definition member CNMCMDU. For more details, see the *IBM System Automation for z/OS Planning and Installation* manual.

SA z/OS uses a set of Automation Table synonyms in member AOFMSGSY. If you change any of your task names, you must update the task names referenced in AOFMSGSY. If you have your own set of AT synonyms, do not place them in AOFMSGSY. Create your own member and include it in INGMSGU1.

## Customizing the communications task

- Communication is between agent and manager
- Task name is INGPXDST
- Definition member is INGXINIT; default values are provided
- If required, e.g. in a GDPS environment, copy INGXINIT from SINGSAMP to DSIPARM and edit typically these items:
  - GRPID=SA 2-character suffix for XCF group for SAplex. Not required
  - LOGSTREAM=NO (Default is YES)
  - PLEXID=00
  - PPI=YES
  - LIFECYCLE=500;MY.AGENT.DATA.SET For extended XCF communication group for AAs in *different* SAplexes within a physical sysplex

Values must match the manager HSAPRMxx values

### Customizing the communications task

The INGPXDST task is used to communicate with other agents in the sysplex and the automation manager. The task reads DSIPARM member INGXINIT when it initializes.

The group ID (GRPID) is a suffix that indicates which group of systems this system communicates with in its sysplex (it might only be a subset). If you do not specify a GRPID, the default group name INGXSG is used. You do not have to modify the defaults unless you are running with a different XCF group ID.

LOGSTREAM=NO is an option indicating that no access to the System Logger is provided.

INGXINIT should ONLY be copied to local DSIPARM if an update to GRPID and / or LOGSTREAM is needed - and that should be the exception - for instance in case GDPS is being used or client wants to have the lpars within the same sysplex belonging to different AM configurations.

The agent must communicate with a manager before the ACF is read. The automation manager provides the location of the ACF and Automation Table. Communication with the agent takes place by using the XCF services feature of z/OS. INGXINIT specifications can use system symbolics such as &SYSCLOSE. For performance reasons, you might have to tune the large and small XCF buffers.

## Defining Extended XCF Communication Groups

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

## XCF overview

- XCF is used for communication between manager and agents and VSAM data sets for backup
- Two XCF groups are defined:
  - INGXSGxx: Communication between manager and agent
    - Group ID (xx) is defined
      - HSAPRMxx GRPID= for manager
      - INGXINIT GRPID= for the agents
    - Default name is INGXSG
  - INGPX\$\$\$: Communication between managers
- You issue the D XCF, GROUP command to display XCF groups
- Multiple logical sysplexes or SAplexes are required if you do not have shared DASD for all systems in the sysplex

### XCF overview

The first six characters of the name of the XCF group used for the managers and agents, INGXSG, are fixed. The last two characters are defined in the start parameters. The XCF group name is used internally to have unique automation manager names. The MVS command **D XCF, GROUP** can be used to display the XCF groups.

By using the GRPID= parameter, multiple PAMs can function within a single sysplex environment by using SAplexes. Each PAM is responsible for its own SAplex resources. The SAplexes are separate XCF group rings within a single sysplex, and they are required if only some systems within the sysplex have shared disks. A practical example is the separation into test and production sysplexes.

Your system administrator can ensure that transport classes for CLASSLEN(956) and CLASSLEN(4028) are defined. The XCF group name must not be assigned to the transport classes. The person who is responsible for the sysplex setup must know that SA z/OS has a maximum XCF message length of 3500 bytes. Either an existing transport class with the appropriate class length can be used, or a new transport class can be defined.

# Display XCF groups

## D XCF, GROUP

**AMs and AAs GRPID=SA defined in INGXINIT and HSAPRMxx**

IXC331I 09.11.04 DISPLAY XCF 090

GROUPS (SIZE) : COFVLFNO (1)

EZBTCPCS (1)

IGG0CAS (2)

INGPX\$\$\$ (2)

INGPX\$00 (2)

INGXSGSA (4)

IOEZFS1 (1)

ISTCFS01 (1)

ISTXCF (1)

MVSA (1)

MVSB (1)

SYSATB01 (2)

SYSBPX (1)

SYSCNZMG (1)

SYSDAE (3)

SYSENF (1)

SYSGRS (1)

SYSIEFTS (1)

SYSIGW00 (2)

SYSIGW01 (2)

SYSIKJBC (1)

SYSIOSPX (1)

SYSIOS01 (1)

SYSJES (1)

SYSJ2\$XD (1)

SYSMCS (6)

SYSMCS2 (42)

SYSRMF (1)

SYSTTRC (1)

SYSWLM (1)

SYSXCF (1)

**Only AMs**

**extended XCF communication group for AAs in different SAplices with PLEXID 00 defined in INGXINIT**

In this example, two managers and two agents are active. All are connected with XCF

## Display XCF groups

On this slide, you see an example of an MVS **D XCF, GROUP** command. The three XCF groups that are used by SA z/OS are highlighted. The **INGAMS** command can also be used to display the XCF group.

## Customizing SA z/OS operations

- CNME1049: Runs when user logs on to set environment variables
- AOFEXDEF: Runs at agent initialization to set AAO global variables
- CNMSTYLE: Can be used to set AAOs instead of AOFEXDEF:

Example:

- COMMON.AOFSMARTMAT=3
- COMMON.AOFCNMASK = 290C0D0E0F101518
- COMMON.INGREQ\_ORIGINATOR = 1
- COMMON.AOFRESTARTALWAYS = 0
- COMMON.AOF\_AAO\_SDFROOT.0=2
- COMMON.AOF\_AAO\_SDFROOT.1=&SYSNAME .
- COMMON.AOF\_AAO\_SDFROOT.2=MVSA MVSB MVSC

Notes:

- AAO documentation is in SINGNPRM(AOFSTYLE)
- RESTYLE COMMON command can be issued to reset common global variables. Avoid overlap of CNMSTYLE common global variables with AOFEXDEF

### Customizing SA z/OS operations

You can customize SA z/OS operations by using Advanced Automation Options (AAO) global variables. You can set these variables in the following ways:

- **AOFEXDEF:** This exit gets control at the start of SA z/OS initialization. You can change, for example, console masks and processing options.
- **CNME1049:** This variable is the initial CLIST that most installations define for their logged-on users. You can use CNME1049 to change items such as screen format defaults.
- **CNMSTYLE:** SA z/OS definitions such as the AAOs can be defined by using CNMSTYLE, rather than coding them in user exit AOFEXDEF. However, CNMSTYLE global variables can also be reset by using the **RESTYLE COMMON** command. Avoid overlap of CNMSTYLE common global variables with AOFEXDEF.



**Hint:** In general, you do not edit CNMSTYLE. Instead, you edit members such as CNMSTGEN. CNMSTYLE is used in these class materials as a reference to the NetView style sheet.

See the *IBM System Automation for z/OS: Customizing and Programming* manual for a detailed list of these Advanced Automation Options (AAOs).

## Control load of AT and MRT

Use COMMON.AOFSMARTMAT AAO to control load of Automation Table and message revision table:

- 3: Refresh at INGAMS REFRESH; use the automatically built AT and MRT
- 2: Refresh at INGAMS REFRESH; use the automatically built AT
- 1: Refresh at INGAMS REFRESH; use ATs from DSIPARM
- 0: Disable Easy Message Management

### Control load of AT and MRT

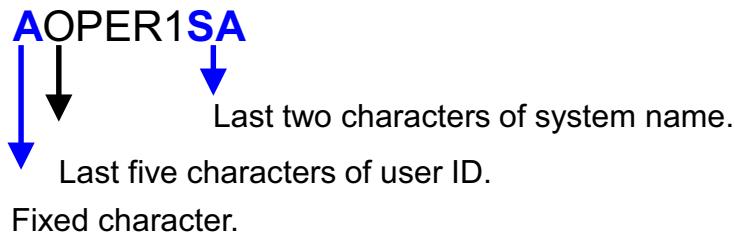
The common global variable AOFSMARTMAT controls whether the AT and MRT fragments generated at Automation Control File build should be used. For compatibility reasons the provided default is 2 indicating that the generated AT fragment is loaded at SA z/OS initialization time or during an INGAMS REFRESH. The recommended value of AOFSMARTMAT is 3 indicating that the generated AT and MRT fragments are loaded at SA z/OS initialization time or during an INGAMS REFRESH.

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.

## EMCS consoles for operators

- Console names must be unique within the sysplex
- AOCGETCN reserves EMCS console with SETCONID command

Example for OPER1 in MVSA:



- AOCGETCN cannot run before SA z/OS is initialized
- Generated name can be modified
  - Change AOFCNMASK in AOFEXDEF
  - Change COMMON.AOFCNMASK AAO in CNMSTYLE

### EMCS consoles for operators

To issue z/OS commands from NetView, the operator must obtain a uniquely named EMCS console. This would include autotask operators. The name must be unique within the system and within the sysplex. The default console name associated with an operator ID is controlled with the *ConsMask* statement in CNMSTYLE.

SA z/OS provides the AOCGETCN command to obtain an EMCS console with a specialized name. AOCGETCN is enabled in NetView CLIST CNME1049 (LOGPROF1) when you enable the SA tower. SA z/OS uses AOCGETCN to obtain a console with a name constructed as shown in the example on this slide. If a z/OS command fails because there is a duplicate console name, AOCGETCN is run automatically to assign an EMCS console with a unique name. This is enabled by a statement in Automation Table INGMSG01. However, the failed command is not automatically retried. AOCGETCN requires the SA z/OS agent to be active. In some cases, you might have to use z/OS commands before the agent initializes. You can use GETCONID or SETCONID for those cases. For more information, see the *IBM System Automation for z/OS Customizing and Programming* manual.

## More CNMSTYLE customization

- Set the banner in NCCF:  
`banner = my SA z/OS`
- Do not write held messages to the Syslog when NetView is shutdown:  
`DEFAULTS.HOLD=LOCAL`
- Do not logoff idle NetView operators :  
`idleParms.exceptOp = OPER1,OPER2`
- Disable memstore if you test your own REXX scripts, otherwise changes are not picked up as your REXX scripts are cached:
  - Use command MEMSTOUT member or define
  - `memStore.stgLimit = 0%`
- Define designator character string to enable system operators to issue NetView commands:  
`MVSPARM.Cmd.Designator=%`

You can use the RESTYLE command to process some of the CNMSTGEN updates

### More CNMSTYLE customization

Set the banner in NCCF:

`banner = my SA z/OS`

Do not write held messages to the Syslog when NetView is shutdown:

`DEFAULTS.HOLD=LOCAL`

Do not logoff idle NetView operators:

`idleParms.exceptOp = OPER1,OPER2`

Disable memstore if you test your own REXX scripts, otherwise changes are not picked up as your REXX scripts are cached:

- Use command MEMSTOUT member or define
- `memStore.stgLimit = 0%`

Define designator character string to enable system operators to issue NetView commands:

`MVSPARM.Cmd.Designator=%`

You can use the RESTYLE command to process some of the CNMSTGEN updates

## Switch off network related functions

- Do not select any of the NetView Network Management towers, only select  
TOWER = SA
- Do not start CNMTAMEL:  
CNMTAMEL INIT=N
- Do not start XCF support task:  
DSIXCFMT INIT=N
- Do not start asynchronous GetHostBy Name/Addr task:  
DUIDGHB INIT=N
- Use the CNMI statement to define whether this NetView® program owns the Communications Network Interface (CNMI)  
CNMI = No
- Indicate that this NetView program is not designated to receive MDS-MUs with the CP name as the destination  
VTAMCP.USE = No

### Switch off network related functions

To switch off network related functions do:

- Do not select any of the NetView Network Management towers, only select  
TOWER = SA
- Do not start CNMTAMEL:  
CNMTAMEL INIT=N
- Do not start XCF support task:  
DSIXCFMT INIT=N
- Do not start asynchronous GetHostBy Name/Addr task:  
DUIDGHB INIT=N
- Use the CNMI statement to define whether this NetView® program owns the Communications Network Interface (CNMI)  
CNMI = No
- Indicate that this NetView program is not designated to receive MDS-MUs with the CP name as the destination  
VTAMCP.USE = No

## NetView enable RMTCMD

If you have more than one system with NetView enable the RMTCMD:

To use RMTCMD over an LU 6.2 session, tasks DSIHPDST and DSIUDST must be active on both NetView systems.

To use RMTCMD over TCP/IP, only the DSIUDST task must be active

- Check definitions in DSIIUNIT initialization member of the DSIUDST RMTCMD support task for NETID and port
  - Start DSIUDST RMTCMD support task: TASK.DSIUDST.INIT=Y
  - Stop an LU 6.2 or IP communication session between a status focal point host and a server workstation  
`function.autotask.NetConv = *NONE*`
  - Use RMTSYN statement to specify the protocol used for communications by the PIPE ROUTE stage and by the RMTCMD and LABEL commands. Specify your domain name and netid:  
`RMTSYN.netid.domain = ip_address`
- Example:** RMTSYN.USIBMES.ING01=10.31.&IPBSUB..&IPBLLQ.

### NetView enable RMTCMD

If you have more than one system with NetView enable the RMTCMD.

The RMTCMD command processor sends system, subsystem, and network commands to a remote NetView system for processing. The responses to these commands are returned to the RMTCMD issuer.

To use RMTCMD over an LU 6.2 session, tasks DSIHPDST and DSIUDST must be active on both NetView systems. To use RMTCMD over TCP/IP, only the DSIUDST task must be active.

The RMTSYN statement specifies the protocol used for communications.

- Check definitions in DSIUINIT initialization member of the DSIUUDST RMTCMD support task for NETID and port

- Start DSIUUDST RMTCMD support task: TASK.DSIUUDST.INIT=Y

- Stop an LU 6.2 or IP communication session between a status focal point host and a server workstation

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

- Use RMTSYN statement to specify the protocol used for communications by the PIPE ROUTE stage and by the RMTCMD and LABEL commands. Specify your domain name and netid:

```
RMTSYN.netid.domain = ip_address
```

```
Example: RMTSYN.USIBMES.ING01=10.31.&IPBSUB..&IPBLLQ.
```

- If you get error messages like:

```
INGC102 AOFDA % 13:37:24 - BNH697I REMOTE OPERATIONS TCP/IP SERVER SET-UP  
FAILED
```

```
DSIUUDST AOFDA 13:37:24 - BNH604I INITAPI REQUEST FAILED, ERRNO = 10218
```

- you have to make sure that tcPIP is started and the port defined in DSIUUDST initialization member is connectable.

- You can use the NETSTAT command to verify your TCP port connections.

```
MVS DISPLAY TCPIP,,NETSTAT,PORTLIST
```

See also the RMTSYN documentation and related Technote about:

Defining NETVIEW RMTSYN statement when using RMTCMD over TCP/IP.

# Lesson 5 REXX function packages

## Lesson 5: REXX function packages

- SA z/OS supplies its own REXX function packages
  - NetView 6.2.1 loads INGRXFPG to the function package table at initialization time
  - INGTXFPG must be defined for the TSO and batch MVS environments
    - Required for the batch command receiver and the customization dialog AT override syntax checking
- You must add TSO REXX Function Package INGTXFPG to the function package table in the MVS IRXTSPRM module
  - Edit the IRXREXX2 sample job in SYS1.SAMPLIB to add the SA for z/OS package name INGTXFPG to the sample IRXTSPRM module code
  - Assemble and link-edit the IRXTSPRM module into a LPALST data set

### What this lesson is about

This lesson provides an overview of the REXX function packages used with System Automation for z/OS.

### Objectives

After completing this lesson, you should be able to identify and describe the REXX function packages used with System Automation for z/OS.

### References

*SC34-2716 System Automation for z/OS Version 4  
Release 1 Planning and Installation*

You must install two REXX function packages: INGRXFPG for the NetView environment and INGTXFPG for the TSO and MVS environment.

All NetView REXX functions of SA z/OS are packaged in module INGRXFPG. This package will be automatically loaded to the function package table in NetView at initialization time when running NetView 6.2.1 or higher.

The SA for z/OS installation and planning manual provide instructions for installing the INGTXFPG package.

INGTXFPG requires that you modify, assemble, and link-edit an assembler code module that contains the names of the REXX function packages:

- Edit the IRXREXX2 sample job in SYS1.SAMPLIB to add the SA for z/OS package name INGTXFPG to the sample IRXTSPRM module code
- Assemble and link-edit the IRXTSPRM module into a LNKLST data set

# INGTXFPG

```
EDIT      INGC200.JAYS.JCL(IRXREXX2) - 01.01          CHARS 'INGTXFPG' found
Command ==> _____ Scroll ==> CSR
000346 *
000347 PACKTB_SYSTEM_FIRST DC A(PACKTB_ENTRIES)
000348 *
000349 PACKTB_SYSTEM_TOTAL DC F'3'                   /* entries in use */
000350 *
000351 PACKTB_SYSTEM_USED DC F'3'                  /* Address of the first*/
000352 *                                         /* System entry */
000353 PACKTB_LENGTH DC F'8'                      /* Total number of @P1C*/
000354 PACKTB_FFFF DC X'FFFFFFFFFFFFFFFFFF'       /* System entries @P1C*/
000355 PACKTB_ENTRIES EQU *                      /*Number of System @P1C*/
000356                                     /* entries in use @P1C*/
000357 PACKTB_ENTRY_MVS EQU *                  /* Length of each PACKTB entry */
000358 PACKTB_NAME_MVS DC CL8'IRXEFMVS'        /* Set the function package name*/
000359 PACKTB_NEXT_MVS DS DC                 /* Point to the next entry */
000360 PACKTB_ENTRY_TSO EQU *                  /* The TSO PACKTB entry @PG10210*/
000361 PACKTB_NAME_TSO DC CL8'IRXEFPCK'        /* Set the function package name*/
000362 PACKTB_NEXT_TSO DS DC                 /* Point to the next entry */
000363 PACKTB_ENTRY_SAM EQU *                  /* The TSO PACKTB entry @PG10210*/
000364 PACKTB_NAME_SAM DC CL8'INGTXFPG'        /* Set the function package name*/
000365 PACKTB_NEXT_SAM DS DC                 /* Point to the next entry */
```

© Copyright IBM Corporation 2019

2-37

INGTXFPG

This slide shows the IRXREXX2 assembler code. You must modify, assemble, and link-edit this code to provide the name of the SA for Z/OS REXX function package INGTXFPG to the TSO and MVS environment. The link edited module should be put in LNKLST. This function package is used by a batch command receiver and the message override syntax checking feature that are new in SA for z/OS version 4.1.

The SINGMOD1 library must be available once the TSO REXX package module has been updated. If it is not available, TSO sessions can not be started.

# Lesson 6 Security and Authorization

## Lesson 6: Security and Authorization

- Authorization of the Started Procedures
- Use System Authorization Facility (SAF), such as RACF, and or NetView
- Secure your automation and environment :
  - User access to the NetView program
    - UserIDs, passwords, and logon attributes
  - Access to commands and data sets
    - NetView and non NetView commands
  - Resource authorization
  - Keyword and value authorization
  - Terminal access restrictions
  - Cross-domain logons

### What this lesson is about

This lesson describes how to secure your System Automation environment.

### Objectives

After completing this lesson, you should be define security and authorization using NetView, System Automation for z/OS and your SAF product

### References

*SC34-2716 System Automation for z/OS Version 4 Release 1 Planning and Installation*  
*IBM Tivoli NetView for z/OS Version 6 Release 2 Security Reference SC27-2863*

You can secure the product. Only authorized personnel are able to access product-specific data sets, find out runtime information about automated resources, or change the status of such resources.

This lesson describes the necessary steps:

- Authorization of the Started Procedures

- Use System Authorization Facility (SAF), such as RACF, and or NetView

An SAF product, such as RACF, is an application that supports the RACROUTE interface and performs functions such as centralized auditing, resource authorization, and user identification and verification. For example, RACF lets you limit data set access. Using an SAF product also helps simplify and centralize your security. All the operator and command authorization syntax is created and managed within one product, rather than handled uniquely for each application.

- Secure your automation and environment :

- User access to the NetView program
  - ◆ UserIDs, passwords, and logon attributes
- Access to commands and data sets
  - ◆ NetView and non NetView commands
- Resource authorization
- Keyword and value authorization
- Terminal access restrictions
- Cross-domain logons

## Authorization of the started procedures

The started procedures for the automation manager, the automation agent , the subsystem interface, and the IPL data gatherer need authority to access SAF-protected resources.

- Use the STARTED class
- Use the real procedure name
- The user IDs of the SA z/OS started procedures require:
  - Read access to product libraries
  - Update access to most user libraries
  - IPL Data Gatherer requires control access to IPLDATA
- The automation manager 's user must have an OMVS segment and access to the BPX.SUPERUSER resource
- Additionally, add libraries SCEERUN, SCEERUN2, CSSLIB, and SCLBDLL as Program Controlled and authorize the automation manager 's user accordingly

### *Authorization of the started procedures*

The started procedures for the automation manager, the automation agent, the Subsystem Interface, and the IPL Data Gatherer need authority to access SAF-protected resources.

Use the STARTED class for the real procedure name. None of the started procedures requires the PRIVILEGED or TRUSTED attribute. You must check with your security administrator for details.

The user IDs of the SA z/OS started procedures require:

- Read access to product libraries
- Update access to most user libraries
- IPL Data Gatherer requires control access to IPLDATA

Ensure that the user ID that is associated with the NetView started task or job is defined to your SAF product with an OMVS segment.This is required so that the NetView program can use z/OS Unix System Services (USS) functions, which include TCP/IP functions.

The automation manager's user must have an OMVS segment and access to the BPX.SUPERUSER resource.

Additionally, add libraries SCEERUN, SCEERUN2, CSSLIB, and SCLBDLL as Program Controlled and authorize the automation manager's user accordingly.

# Setting up security

| Type of Security                  | Using an SAF Product   | Using NetView and SA z/OS                  |
|-----------------------------------|--|--|
| Defining operators and passwords  | APPL class and NETVIEW segment   | Profiles in DSIPRF and operators in DSIOPF |
| NetView command authorization     | NETCMDS class  | NetView command authorization table        |
| RMTCMD                            | RMTOPS class   | Remote security table like DSISECUR        |
| Data set access                   | DATASET class  | Not available                              |
| Cross system commands             | SYSAUTO class  | Not available                              |
| Use of commands from TSO or Batch | Front-end checking: SYSAUTO class<br>Back-end checking: NETCMDS class                        | NetView command authorization table        |
| Resources                         | SYSAUTO class and profiles for SA z/OS special resources and resources controlled by SA z/OS | NetView command authorization table        |
| Restrict logon access             | TERMINAL class, WHEN Keyword   | Not available                              |

NetView also provides security for: Defining resources to spans, Authorizing operators to spans, RUNCMD, RODM

© Copyright IBM Corporation 2019

2-40

## Setting up security

This table provides an overview of the types of security that you can use to ensure the integrity of your system using an SAF Product and/or using NetView and SA z/OS

The initial NetView security settings are defined by the SECOPTS statements in the CNMSTYLE member. These statements define the security method that is used by the NetView product to specify the following items:

- The type of password checking.
- Where the NetView operator logon attributes are defined.
- Where span of control is defined (Only for network).
- The type of command authorization.
- All operators, human and automated operators, can be defined and authenticated by an SAF product.
- Command authorization can be done by a SAF product based on the issuer of a command.
- Resource authorization can be done by a SAF product based on the issuer of a particular command.

# Security considerations for automation

- Restrict data set access to the DSIPARM data sets for automation tables and DSICLD data sets for command lists
- Restrict access to Automated Operator Tasks (autotasks)
  - Specify non-expiring passwords for autotasks to prevent operators from logging on to autotask operator IDs, and to prevent automation from being affected by an expired password
  - Use command authorization to restrict the type of commands which operators can send to autotasks
    - Restrict the NetView EXCMD command or use AUTHCHK=SOURCEID
- Bypass authority-checking of commands from the automation table if:
  - Your automation table is secure enough
  - Authority-checking impacts your system performance
  - Force authority-checking of disruptive commands:
    - Use SEC=CH on the CMDDEF statement in CNMCMD

## Security considerations for automation

- Restrict data set access to the automation table and DSICLD
- Restrict access to Automated Operator Tasks (autotasks)

Specify non-expiring passwords for autotasks to prevent operators from logging on to autotask operator IDs, and to prevent automation from being affected by an expired password  
Use command authorization to restrict the type of commands which operators can send to autotasks. Restrict the NetView EXCMD command or use AUTHCHK=SOURCEID
- Bypass authority-checking of commands from the automation table if:
  - Your automation table is secure enough
  - Authority-checking impacts your system performance
  - Force authority-checking of disruptive commands:
    - ◆ Use SEC=CH on the CMDDEF statement in CNMCMD

## SECOPTS statements and options in CNMSTYLE

```
*SECOPTS.OPERSEC = NETVPW          SECOPTS.OPSPAN = NETV
SECOPTS.OPERSEC = SAFCHECK         *SECOPTS.OPSPAN = SAF
*SECOPTS.OPERSEC = SAFPW           SECOPTS.SPANAUTH = *NONE*
*SECOPTS.OPERSEC = SAFDEF          *SECOPTS.SPANAUTH = TABLE.CNMSPAN2
*SECOPTS.OPERSEC = MINIMAL         *SECOPTS.SPANAUTH = VTAMLST.CNMSPAN1

SECOPTS.SURROGAT = NO              SECOPTS.SPANCHK = SOURCEID
*SECOPTS.CMDAUTH = TABLE.CNMSCAT2 *SECOPTS.SPANCHK = TARGETID
*SECOPTS.CMDAUTH = SAF.CNMSBAK1   SECOPTS.WEBAUTH = PASS
SECOPTS.CMDAUTH = SAF.PASS         *SECOPTS.WEBAUTH = CHECK
*SECOPTS.CMDAUTH = SAF.FAIL        SECOPTS.RMTAUTH = SENDER
*SECOPTS.CMDAUTH = SCOPE.CNMSCOP1 *SECOPTS.RMTAUTH = ORIGIN

SECOPTS.AUTHCHK = SOURCEID
*SECOPTS.AUTHCHK = TARGETID
```

© Copyright IBM Corporation 2019

2-42

### *SECOPTS statements and options in CNMSTYLE*

You can use operands to the SECOPTS statements as follows:

- OPERSEC defines the method used for logging on to the NetView program.
- SURROGAT specifies if the NetView operator ID is checked to determine if it is a surrogate of a TSO user ID.
- CMDAUTH defines the method that NetView uses for protecting command usage.
- AUTHCHK specifies the user ID that is to be used when verifying command authorization.
- OPSPAN defines the method for determining the authority of an operator to start spans of control.
- SPANAUTH specifies the location of the span definitions for resources and views.
- SPANCHK specifies the operator ID that is used for defining span checking.
- WEBAUTH specifies if authorization checking is to be performed for operator access to the NetView Web server.



**Note:** RMTAUTH specifies the method to determine the operator ID that is used as the remote operator for security checks that are performed on RMTCMD and ENDTASK requests. RMTAUTH is used for only incoming requests, not for command security checking.

## Recommended stylesheet options

Modify the CNMSTGEN member, generated by the Configuration Assistant:

- SAF operator identification and password checking:  
`SECOPTS.OPERSEC = SAFDEF`
- SAF command authorization checking:  
`SECOPTS.CMDAUTH = SAF.PASS`
- Check the authority of the original issuer:  
`SECOPTS.AUTHCHK = SOURCEID`
- Commands from the automation table are not authority-checked:  
`DEFAULTS.AUTOSEC = BYPASS`
- Activate resource level security checks:  
`SECOPTS.SARESAUT = ON.PASS`
- Display the current security settings :  
`LIST SECOPTS`

© Copyright IBM Corporation 2019

2-43

### *Recommended stylesheet options*

When you are ready to switch to SAF-based security, in your <sa\_hlq\_user>.DSIPARM data set, edit the CNMSTGEN member and activate the following options:

The first option specifies that operator identification and password or password phrase checking is done with an SAF security product.

`SECOPTS.OPERSEC = SAFDEF`

The second option specifies that the NetView component performs command authorization checking with an SAF security product. Users can issue all commands when the SAF product cannot make a security decision. This option avoids the need to define profiles and permissions for all non-critical NetView component commands explicitly.

`SECOPTS.CMDAUTH = SAF.PASS`

The third option specifies to check the authority of the original issuer or the ID closest to the original issuer. Make sure, you specify each of the options once and you comment out the default settings in this member.

`SECOPTS.AUTHCHK = SOURCEID`

The fourth option specifies that commands routed tasks from the NetView automation table are not authority-checked by a SAF security product, unless SEC=CH was specified on the CMDDEF statement.

```
DEFAULTS.AUTOSEC = BYPASS
```

You activate resource level security checks by setting the following stylesheet option in CNMSTGEN:

```
SECOPTS.SARESAUT = ON.PASS
```

## Configuration assistant generates the INGESAF member, also provided in SINGSAMP

- Profiles that protect commands and other resources
- Definitions of groups that represent roles
- Group membership that contain the individual operators in each role
- Necessary definitions for all the auto operators that are required by the product
- PERMIT statements that grant certain roles access to definitions for commands

© Copyright IBM Corporation 2019

2-44

### *Configuration assistant generates the INGESAF member*

You find the INGESAF member and all the other generated members in the CONFLIB data set. A sample is also provided in SINGSAMP. Refer to the description section of this member and discover the provided security definitions within this member. For establishing the SAF-based security environment it is required to use the Configuration Assistant. The INGESAF member contains:

- Profiles that protect commands and other resources
- Definitions of groups that represent roles
- Group membership that contain the individual operators in each role
- Necessary definitions for all the auto operators that are required by the product
- PERMIT statements that grant certain roles access to definitions for commands

## Security definitions migration to SAF and backup

- Keep your existing definitions and profiles in DSIOPF and DSIPRF for reference
- Keep your NetView command authorization table as a backup
- Using the SEC=BY keyword on the CMDDEF statement in CNMCMU bypasses security checking and can improve system performance
- Immediate commands (TYPE=I or B on the CMDDEF statement) can only be protected by the NetView command authorization table. Use BACKTBL keyword on the REFRESH Code or CNMSTYLE:
  - SECOPTS.CMDAUTH = SAF.backtbl\_name
  - REFRESH CMDAUTH=SAF, BACKTBL=backtbl\_name
- When the SAF product cannot make a security decision because it is not active, NETCMDS class is not active, or is missing definitions:
  - Activate a backup NetView command authorization table (see above) or
  - Disable command authority checking in CNMSTYLE or with REFRESH command
    - SECOPTS.CMDAUTH=SAF.PASS
    - REFRESH CMDAUTH=SAF, SAFNODEC=PASS

### *Security definitions migration to SAF and backup*

- Keep your existing definitions and profiles in DSIOPF and DSIPRF for reference
- Keep your NetView command authorization table as a backup
- Using the SEC=BY keyword on the CMDDEF statement in CNMCMU bypasses security checking and can improve system performance
- Immediate commands (TYPE=I or B on the CMDDEF statement) can only be protected by the NetView command authorization table. Use BACKTBL keyword on the REFRESH Code or CNMSTYLE:

```
SECOPTS.CMDAUTH = SAF.backtbl_name
REFRESH CMDAUTH=SAF, BACKTBL=backtbl_name
```

- When the SAF product cannot make a security decision because it is not active, NETCMDS class is not active, or is missing definitions:
- Activate a backup NetView command authorization table (see above) or
- Disable command authority checking in CNMSTYLE or with REFRESH command

```
SECOPTS.CMDAUTH=SAF.PASS
REFRESH CMDAUTH=SAF, SAFNODEC=PASS
```

## Roles

| Role          | Default Group – USS Group ID      | Description  |
|---------------|-----------------------------------|--|
| User          | INGUSER – 80002                   | Display only   |
| Operator      | INGOPER – 80003                   | Everything to keep the system running  |
| Administrator | INGADMIN – 80004                  | Remote security table like DSISECUR  |
| Auto Operator | INGAUTO – 80001<br>INGWRK – 80006 | Everything to bring resources into their desired status or recover from failures<br>INGWRK: additional permissions |
| Superuser     | INGSUPER – 80005                  | no restrictions  |

**Note:** The mapping of roles and commands in the INGESAF member is only a guideline. Following the recommendations in this member, however, reduces the time to secure the environment

### Roles

To facilitate the definition of command authorizations for human and auto operators, it is recommended to use groups. Each group corresponds to a certain usage profile or role.

The product comes with five predefined roles that are described in this table:

- User
- Operator
- Administrator
- Auto Operator
- Superuser

Refer to the INGESAF member for a complete reference of commands and services and the associated roles as provided by the product.



**Note:** The mapping of roles and commands in the INGESAF member is only a guideline. Following the recommendations in this member, however, reduces the time to secure the environment.

# Lesson 7 Configuration assistant

## Lesson 7: Configuration assistant

Replace traditional customization with the configuration assistant

- Complete options file
- Run configuration assistant batch job
- Use generated jobs to allocate runtime data sets
- PARMLIB, PROCLIB, VTAMLST members generated
- SAF definitions generated

### What this lesson is about

This lesson provides an overview of the configuration assistant of System Automation for z/OS.

### Objectives

After completing this lesson, you should be able to use the configuration assistant to automate the majority of the configuration process

### References

*SC34-2716 System Automation for z/OS Version 4 Release 1 Planning and Installation*

A new configuration assistant function is available.

Instead of manually adapting configuration jobs, start procedures, and initialization files to your environment, this assistant generates these files for you.

- Faster time to value through the Configuration Assistant for the runtime environment.
- Automation comes with five predefined operator roles. It avoids user exits to provide granular security and to exploit z/OS's security access facility and thus leverages existing security governance that is already in place in an enterprise's mainframe environment.
-

# Traditional approach

Read the Installation manual:

- Decide which of the many installation steps apply to your z/OS environment

Perform those steps by:

- Adapting all the identified sample files
  - Completing your environmental data at multiple places spread across the sample files

And do all this in a consistent way

### *Traditional approach*

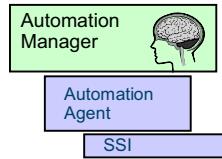
The traditional approach for installing and configuring a product is to read the manuals and then you have to decide which of the many steps apply to your local installation. You perform these step-by-step as described in the thick installation manuals. This means that you adapt the delivered samples at multiple places spread often over several sample files.

The challenge for the system administrator is to do this all in a consistent way. So, normally the system administrator crosses their fingers that this works the right way.

# Configuration assistant

## Overview

- Simple configuration to get base automation components running
  - Automation manager
  - Automation agent
  - Subsystem interface
- Use of **configuration assistant** that automates majority of configuration process
- Keep number of configuration variables down at the necessary minimum
- Benefit from lab **experience** using “standard” option set (stylesheet)
- Post install/**configuration verification** to ensure all necessary steps (in particular steps that must be done by other persona) have been completed



© Copyright IBM Corporation 2019

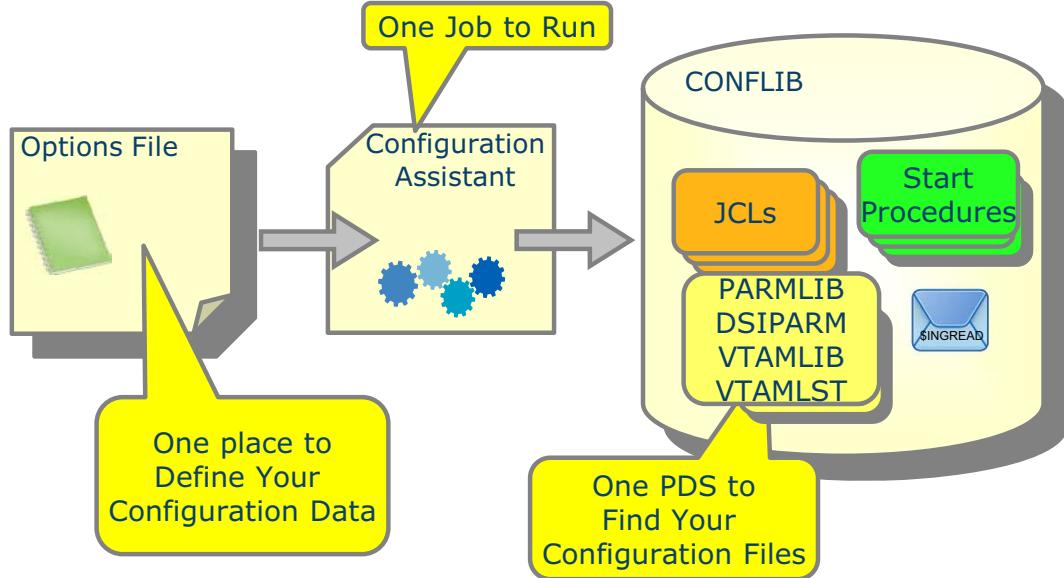
2-49

### Configuration assistant

After you installed the product with SMP/E, the configuration assistant comes into play. It covers the actions to be done when the product target libraries are available to your system environment. The configuration assistant sets up an environment so that the base automation components are running. These are the automation manager, the automation agent, and the subsystem interface.

The configuration assistant automates most of the configuration steps. The concept is that the number of configuration variables is as low as absolutely necessary (approximately 30). When using the configuration assistant, you benefit from the experience of the IBM Böblingen lab. After the configuration assistant has completed, a verification of the generated setup can be done against your system environment.

## Configuration assistant: Concept



Generating Configuration Files supporting system symbols.

© Copyright IBM Corporation 2019

2-50

### Configuration assistant: Concept

The concept of configuration assistant is to have one place to store your configuration data. Then, have one job to process them. Then, have one output PDS where all your JCLs, start procedures, and parameter files are stored. Now the SYSPROG can inspect the members and, if it is required, can adapt them to the local needs.

The CONFLIB data set contains these items:

- Jobs to allocate all data sets and USS paths that are required by SA z/OS during runtime.
- Procedures to start the components of SA z/OS to be copied to your target SYS1.PROCLIB.
- Runtime configuration members for both automation manager and automation agent.
- Parameter files that are ready to be copied to your target SYS1.PARMLIB.
- VTAM definitions that are files ready to be copied to your target VTAMLST.
- Jobs to delete data set files and USS paths in case you must reconfigure or delete SA z/OS again.
- A job to verify the success of the installation and configuration process.

# Configuration assistant steps

## INGDOPT Configuration Options file

1. Allocate a data set where you can maintain working copies of INGDOPT
2. Create a work copy (from SINGSAMP) of the INGDOPT Configuration Options file and the Configuration Assistant sample job (INGDCONF)
3. Look at the INGDOPT Configuration Options file member and prepare some answers especially around naming, SMS, network...
4. Edit the working copy of the INGDOPT Configuration Options file to reflect the parameters of the installed environment. These parameters are then used to build the necessary artefacts to complete the configuration
  - Do use **system symbols** so that the INGDOPT member is enterprise-wide
  - USER data set HLQ must be different from SMPE HLQ
  - Skip optional steps like Workload Scheduler, CICS, Event/Automation Service, RACF... – if not used
5. Check and submit INGDCONF job which will create the CONFLIB
  - You need to update jobcard and variables
6. If you rerun the configuration assistant, the CONFLIB will be deleted and reallocated
  - You need to create a backup copy

### INGDOPT Configuration Options file

Preparation consists of the following steps:

1. Allocate a data set where you can maintain working copies of INGDOPT.  
Compose the name of that library out of a high-level qualifier (HLQ), the SAplex name (SAplex) both of your choice and the low-level qualifier (LLQ) named CONFWRK. This naming scheme must not be changed because it is used by the Configuration Assistant job. For example, if you decide to use the HLQ of 'USER' and you configure SA z/OS on z/OS systems belonging to a sysplex named SYSPLEX1, the recommended name for the work data set is USER.SYSPLEX1.CONFWRK.  
The length of the data set name must not exceed 35 characters
2. Create a work copy (from SINGSAMP) of the INGDOPT Configuration Options file and the Configuration Assistant sample job (INGDCONF)
3. You define various settings that vary from installation to installation in the INGDOPT Configuration Options file. Typical examples are data set high-level qualifiers, system name, and the NetView domain name. These settings are used to build the configuration files in the CONFLIB data set. Look at the INGDOPT Configuration Options file member and prepare some answers especially around naming, SMS, network...

4. Edit the working copy of the INGDOPT Configuration Options file to reflect the parameters of the installed environment. These parameters are then used to build the necessary artefacts to complete the configuration. The INGDOPT Configuration Options file contains comprehensive documentation on the purpose of the parameters.
  - Do use system symbols so that the INGDOPT member is enterprise-wide
  - USER data set HLQ must be different from SMPE HLQ
  - Skip optional steps like Workload Scheduler, CICS, Event/Automation Service, RACF... – if not used
5. Check and submit INGDCONF job which will create the CONFLIB:  
You need to update jobcard and variables within your INGDCONF work copy. When finished, submit the job.

If you rerun the configuration assistant, the CONFLIB will be deleted and reallocated, therefore you need to create a backup copy.

The CONFLIB data set stores the generated JCLs, start procedures, parmlib members, and other initialization and configuration members.

## INGDOPT Configuration Options file for jobcard, names, SMS, HLQs...

| Category | Option                   | Default or Example < Comment   |
|----------|--------------------------|--|
|          | sys_jobcard_1            | //ING<#4:8> JOB 'CONFIGURATION',                                     |
| Item     | sys_sysname              | &SYSNAME. < Use system symbol  |
|          | net_netview_domain_id    | &SADOMAIN. < Use system symbol                                       |
|          | net_netview_majnode      | SA < different from domain. Fixed name for enterprise-wide or cloned |
|          | sa_hlq_smpe=SA           | SA < Prefix for SMP/E target library                                 |
|          | nv_hlq_smpe              | NETVIEW < Prefix for SMP/E target library                            |
|          | sa_hlq_user              | SAZOS.USER < Prefix for user data sets                               |
|          | sys_sms_managed_datasets | YES < SMS managed SA z/OS data sets                                  |
|          | sys_sms_storclass,...    | SMS Data, Management, and Storage Class - or                         |
|          | sys_dataset_volser,...   | UNIT and VOLSER if NOT SMS managed                                   |
|          | sys_hlq_sclbd1           | Installation prefix of the C/C++ runtime library SCLBDLL             |
|          | sys_hlq_sceerun          | Installation prefix of LE/390 runtime libraries SCEERUN, SCEERUN2    |
|          | sys_parmlib_suffix       | SO SA  |

© Copyright IBM Corporation 2019

2-52

### INGDOPT Configuration Options file for jobcard,names, SMS, HLQs...

The INGDOPT Configuration Options follow the following naming structure:

The first part of the option name indicates the category:

- sys: System, contact System Programmer / Storage Administrator
- net: Network, contact Network Administrator
- sa: SA z/OS, contact Automation Programmer
- nv: NetView, contact Automation Programmer
- racf: Security Server RACF, contact Security Administrator

The following parts of the option name indicates the item, sometimes followed by a number if the option requires multiple values, like for jobcard.

In above table the second column shows a Default, if it exists, sometimes followed by an *Example in italic* which can be trailed by a comment separated by a <.

## Syntax:

- Comments must start with character # or \* in column 1
- Each configuration option is defined as a keyword=value pair.  
No comment can be defined in that line.
- Keywords are recommended to start in column 2.  
This allows you to work with the XEDIT 'EXCLUDE' and 'FIND' commands in order to suppress comment lines.

e.g. Issue on the XEDIT command line

```
x all      followed by
f '' 1 all  to only see keyword=value pairs  and
f ....: all  to add additional information:
```

Option .....

Required ...:

Default .....

Example.....:

- leading and trailing blanks are ignored
- Continuation lines must be marked by a plus character '+' at the end of the line.

Options are documented and also specify whether an option is required, has a default or length limitation:

```
*
```

Option .....: net\_netview\_domain\_id

\* Required ....: Yes

\* Default ....: None

\* Max.length...: 5

\* Example.....: ING01 -or-
\* ....: &INGDOM. -or-
\* ....: A&SYMBOL1.B&SYMBOL2.C

\*

net\_netview\_domain\_id=&COURSE.

Here system symbol &COURSE is used. It must be terminated with a .

sys\_parmlib\_suffix=SA is the two character suffix, which should be appended to the generated PARMLIB members. Valid characters are A-Z, 0-9, and the special characters (\$, # and @).

## INGDOPT Configuration Options file for SA items: XCF, PDB, procedure and job names

| Option                | Default or Example < Comment   |
|-----------------------|--|
| sa_xcf_grpid_suffix   | SA < 2-character suffix for XCF group for SA communication in SAplex                     |
| sa_xcf_plexid_suffix  | SA < extended XCF group for communication of multiple SAplexes within a physical sysplex |
| sa_automation_policy  | SAZOS.USER.PDB < Automation Policy Dataset name  |
| sa_am_start_proc      | AUTOMGR < Automation manager procedure name  |
| sa_am_start_job.1     | AM < Job name of the Primary automation manager  |
| sa_am_start_job.2     | AM2 < Job name of the Secondary automation manager                                       |
| sa_saagent_start_proc | INGENVSA < Procedure name of the automation agent  |
| sa_saagent_start_job  | SYSVAPPL < Job name of the automation agent  |
| sa_nvssi_start_proc   | CNMSJ010 < Procedure name of the NetView SSI   |
| sa_nvssi_start_job    | SYSVSSI < job name of the NetView SSI  |
| sa_ipldata_start_proc | HSAPIPLC < Procedure name of the IPL Data Collector                                      |
| sa_ipldata_start_job  | SYSVIPLC < job name of the IPL Data Collector  |

© Copyright IBM Corporation 2019

2-53

INGDOPT Configuration Options file for SA items:XCF, PDB, procedure and job names

### Suffix of the XCF Group ID used by SA z/OS

Specifies the 2-character suffix that composes the XCF group ID that is used by the automation manager and the agents when communicating among each other. If specified, the ID may contain one or two characters. Valid characters are A-Z, 0-9, and the special characters (\$, # and @). The suffix is prefixed with the string INGXSG to construct the XCF Group ID, for example INGXSGxy. You can use any suffix of your choice. If you run multiple automation environments within one physical sysplex, unique suffixes must be defined for each automation environment. If you do not specify a Group ID, the default group name INGXSG is used. This is only valid, if you have only one automation environment in your physical sysplex.

### Suffix of the XCF Sysplex ID (PLEXID) used by SA z/OS: sa\_xcf\_plexid\_suffix

Specifies the 2-character suffix that composes the XCF sysplex ID. It is used to build the name of an extended XCF communication group within a physical sysplex. All SA z/OS automation agents using the same PLEXID belong to the same communication group. This allows you to send commands from one agent to another agent which joins the same group. If specified, the suffix may contain 1 or 2 characters. Valid characters are A-Z, 0-9, and the special characters (\$, # and @). The suffix is prefixed with the string INGPX\$ to construct the XCF Communication Group. You can

use any suffix of your choice except \$\$ . If you do not specify a suffix, then no PLEXID is generated. That means that the agent is not able to communicate with another agent, thus you cannot send commands from or to this system's automation agent

### Start Command of the automation manager

As a result of your input the following syntax is applied when adding start command to the COMMNDxx PARMLIB member:

```
'S sa_am_start_proc,JOBNAM=sa_am_start_job.1,SUB=MSTR'
```

No start command will be added to the COMMNDxx member for 'sa\_am\_start\_job.2'.

A job name sa\_am\_start\_job.2 for the Secondary automation manager is only required when you run it on the same system as the Primary automation manager which is only required in a single system SAplex.

## Optional configuration options for Workload Scheduler, CICS, Event/Automation Service

| Option                   | Default or Example < Comment                                    |
|--------------------------|---|
| sys_tws_hlq_smpe         | The HLQ of your IBM Workload Scheduler installation             |
| sys_tws_mlog_ds          | The message log data set name of the Workload Scheduler tracker |
| sys_cpsm_hlq_smpe        | HLQ of your CICS installation                                   |
| nv_eas_start_proc        | IHSAEVNT < Event/Automation Service (E/AS) procedure name       |
| nv_eas_eif_start_job     | EIFAEVNT < job name of the NetView E/AS started task            |
| nv_eas_eif_init_file     | EIFINIT < E/AS global initialization file member name           |
| nv_eas_eif_messagea_file | EIFMCFG < E/AS message adapter configuration file member name   |
| nv_eas_eif_ppi_name      | INGEVOMN < E/AS Program to Program interface (PPI) name         |
| nv_eas_eif_rec_host      | 0 < Hostname of the event server, typically running OMNIbus     |
| nv_eas_eif_rec_port      | 0 < Port number on which the event server listens               |

All these configuration options are optional.

## Optional configuration options for Security Server RACF

| Option                | Default or Example < Comment                              |
|-----------------------|---|
| racf_group_auto       | INGAUTO < Group Name for SA z/OS AutoOperators            |
| racf_omvs_gid_auto    | 80004 < OMVS GroupID (GID) for above group                |
| racf_omvs_uid         | 90004 < OMVS UID for above group                          |
| racf_omvs_home        | / < OMVS directory for above group                        |
| racf_group_autowrk    | INGWRK < Group Name for SA z/OS work operators (AUTWRKxx) |
| racf_omvs_gid_autowrk | 80006 < OMVS GroupID (GID) for above group                |
| racf_group_user       | INGUSER < Group Name for SA z/OS USER role                |
| racf_omvs_gid_user    | 80001 < OMVS GroupID (GID) for above group                |
| racf_group_oper       | INGOPER < Group Name for SA z/OS OPERATOR role            |
| racf_omvs_gid_oper    | 80002 < OMVS GroupID (GID) for above group                |
| racf_group_admin      | INGADMIN < Group Name for SA z/OS ADMINISTRATOR role      |
| racf_omvs_gid_admin   | 80003 < OMVS GroupID (GID) for above group                |
| racf_group_super      | INGSUPER < Group Name for SA z/OS SUPERUSER role          |
| racf_omvs_gid_super   | 80005 < OMVS GroupID (GID) for above group                |

© Copyright IBM Corporation 2019

2-55

Optional configuration options for Security Server RACF

## Roles

To facilitate the definition of command authorizations for human and auto operators, it is recommended to use groups. Each group corresponds to a certain usage profile or role. The product comes with five predefined roles:

Refer to the INGESAF member for a complete reference of commands and services and the associated roles as provided by the product.



**Note:** The mapping of roles and commands in the INGESAF member is only a guideline. Following the recommendations in this member, however, reduces the time to secure the environment.

## Configuration assistant steps: Follow instructions in CONFLIB(\$INGREAD) 1 of 2

1. Create work data sets
  - Review the first allocation job INGDSALC
    - Increase space for DSILIST data set
    - Check whether DD CNMDVIPP uses the Q1 parameter which creates a JCL error
  - Submit job INGDSALC. For RC>4 a backout is required using job INGDSDEL
2. Configure NetView DSIPARM
  - Copy or merge the listed members from this CONFLIB data set to your user DSIPARM library:
    - CNMSTGEN, INGESCAT, INGPTOP, INGXINIT
3. Copy automation manager parameter member HSAPRMSA
4. Copy JCL startup procedures into your SYS1.PROCLIB concatenation
5. Copy or merge the generated PARMLIB members into your active SYS1.PARMLIB concatenation
  - Implement that automated messages are sent to NetView using MPF or the NetView Message Revision Table
  - Recommended is to use the NetView Message Revision Table (MRT) that is being generated by the SA build. To automatically activate the MRT add to CNMSTGEN:  
COMMON .AOFSMARTMAT=3
  - Alternatively use MPF. In this case check the active MPF member, especially that entry .NOENTRY specifies AUTO(Y), otherwise you might be missing messages in NetView

Follow instructions in CONFLIB(\$INGREAD) 1 of 2

Follow the Instructions as Documented in \$INGREAD documentation member \$INGREAD was tailored to your installation and created in the CONFLIB data set.

Follow the instructions documented there and complete the basic configuration.

### Configuring z/OS PARMLIB

Copy the members or merge the content of the members of the CONFLIB data set into your active SYS1.PARMLIB concatenation: (Suffix used is SA)

- BPXPRMSA, COMMNSA, IEFSSNSA
- Instead of using IEAAPFSA, LNKLSTSA, LPALSTSA consider using PROGSA, if you cannot perform an IPL. You must verify and adapt the member according to your system requirements to avoid destructive actions. When done, copy it to your active SYS1.PARMLIB concatenation.

### MPF

Update your MPF configuration with the following entries

independent of whether you can follow the recommendation below.

AOF603D, SUP (NO) , RETAIN (NO) , AUTO (YES)

AOF\*, SUP (NO) , RETAIN (NO) , AUTO (NO)

It is recommended to use the Message Revision Table (MRT) and Automation Table (AT) technique of this automation product. At a later time, these tables can be generated by this product's customization dialog. This technique does not require further changes of your MPF configuration. If you cannot follow this recommendation, you need to update your MPF configuration to expose all messages to automation which are required to automate your z/OS environment.

For details, refer to "MVS: Initialization and Tuning Reference".

## Configuration assistant steps: Follow instructions in CONFLIB(\$INGREAD) 2 of 2

### 6. Configuring VTAM Connectivity

- Verify the definitions in the AMODETAB, AUSSTAB, ISTSDCOS, and INTERCOS members
- Submit the INGMTAB job to generate tailored VTAM mode tables in .CONFLIB.VTMLIB
- Copy the VTAM Major Node member to the SYS1.VTAMLST concatenation of your active VTAM
- Add the VTAM Major Node member name to your ATCCONxx member
- Copy the .CONFLIB.VTMLIB members into the active SYS1.VTAMLIB concatenation

### 7. Configuring SAF Security

- Submit the INGGENL job which creates .CONFLIB.&SYSNAME. and copies member INGESAF
- For a test system you can also decide to check security only against the NetView STC userID:  
**SECOPTS.OPERSEC = SAFPW**
- Verify the definitions in job INGESAF and submit
- The operator roles are optional

### 8. Activate the new configuration:

- Check that all PARMLIB updates have been done and either perform an IPL or add SSI, LPA, Linklist, and APF dynamically
- Activate VTAM mode table AMODETAB
- Activate the VTAM major node

### 9. There is also a verification job, however it might complain about a problem with non existing majnode

© Copyright IBM Corporation 2019

2-57

Follow instructions in CONFLIB(\$INGREAD) 2 of 2

6. Configuring VTAM Connectivity. Your CONFLIB data set hosts members that contain the VTAM major node and other VTAM definitions. Verify the definitions in the AMODETAB, AUSSTAB, ISTSDCOS, and INTERCOS members and adapt them if required.

- Submit the INGMTAB job. This job allocates a data set with the low level qualifier (LLQ) 'CONFLIB.VTMLIB' and fills it with tailored VTAM mode tables.
- Copy the VTAM Major Node member to the SYS1.VTAMLST concatenation of your active VTAM
- Add the VTAM Major Node member name to your ATCCONxx member
- Copy the .CONFLIB.VTMLIB members into the active SYS1.VTAMLIB concatenation
- Copy the following members from the generated CONFLIB.VTMLIB data set into the SYS1.VTAMLIB concatenation of your active VTAM started task:
  - AMODETAB
  - AUSSTAB
  - ISTSDCOS
  - INTERCOS (optional sample VTAM list - logon mode table for NETA)

7. Configuring SAF Security:

- Submit the INGGENL job which creates .CONFLIB.&SYSNAME. and copies member INGESAF
- For a test system you can also decide to check security only against the NetView STC userID:SECOPTS.OPERSEC = SAFPW
- Verify the definitions in job INGESAF and submit
- The operator roles are optional

8. Activate the new configuration:

- Check that all PARMLIB updates have been done and either perform an IPL or add SSI, LPA, Linklist, and APF dynamically with the following z/OS commands:

```
SETPROG LNKLST,UNALLOCATE  
P LLA  
SET PROG=SA  
S LLA,SUB=MSTR  
SETPROG LNKLST,ALLOCATE  
SETPROG LNKLST,UPDATE,JOBNAME=*
```

- You also need to add NetView as a subsystem to z/OS. Use the z/OS command:

```
SETSSI ADD,SUB=AUTO,INITRTN=DSI4LSIT  
– Activate VTAM mode table AMODETAB  
– Activate the VTAM major node
```

9. Submit the INGDVRFY Configuration Verification job on the target system where SA z/OS was configured. This job is in the CONFLIB library. After the job terminates, investigate the job log for INGVxxxx messages. If required, correct the configuration according to those messages.

# Review questions

1. What are the two major components of System Operations?
2. What three started tasks are used to support the System Automation for z/OS environment?
3. What is the communication method used to communicate between the two major components of System Automation for z/OS?
4. What user interface is used to customize the automation policy data base?
5. Where is the automation policy data base stored?
6. What data set is used to provide for the run time automation policy?
7. What is the recommended start type for the automation manager?
8. Why is STEPLIB not recommended?
9. How should NetView libraries be allocated?
10. What are the advantages of the NetView style sheet?
11. What does the following SECOPTS statements in the CNMSTYLE member do:  
SECOPTS.OPERSEC = SAFCHECK
12. Why use the configuration assistant?
13. How can you make your DSIPARM and the configuration assistant options file enterprise-wide?

# Review answers

1. What are the two major components of System Operations?  
*The automation manager and the automation agent.*
2. What three started tasks are used to support the System Automation for z/OS environment?  
*The automation manager started task, the automation agent started task running under NetView, and the NetView Subsystem Interface (SSI) started task.*
3. What is the communication method used to communicate between the two major components of System Automation for z/OS?  
*Cross System Communications Facility (XCF)*
4. What user interface is used to customize the automation policy data base?  
*The System Automation for z/OS customization dialog*
5. Where is the automation policy data base stored?  
*As multiple members in a partitioned data set called the Policy Data Base (PDB)*
6. What data set is used to provide for the run time automation policy?  
*A partitioned data set referred to as the Automation Configuration File containing the Automation Control file (ACF) and the automation manager configuration (AMC)*
7. What is the recommended start type for the automation manager?  
*A hot start is the recommended start type for the automation manager as service period overrides and current resource states and settings are not lost*
8. Is STEPLIB not recommended?  
*STEPLIB is not recommended for performance reasons as each module is searched in STEPLIB first*
9. How should NetView libraries be allocated?  
*Most installations use a PDSE. Alternatively use single extent only*
10. What are the advantages of the NetView style sheet?  
*The NetView style sheet (CNMSTYLE) provides a simplified way of making definitions to NetView. The style sheet reduces the amount of editing of DSIPARM members (and other data sets). With style sheets, many of the definitions that are required for SA z/OS are enabled automatically by specifying TOWER=SA and TOWER.SA = SYSOPS in the CNMSTYLE definition.*

11. What does the following SECOPTS statements in the CNMSTYLE member do:

SECOPTS.OPERSEC = SAFCHECK

*SAFCHECK defines that operator passwords or password phrases are checked by an SAF product, with operator profiles that are specified in the NetView DSIOPF member and logon attribute values that are defined in the DSIPRF data set. Access to protected data sets and any commands are checked at the individual task level by the SAF product. With SAFDEF, Logon attributes are defined in the NETVIEW segment of an SAF product.*

12. Why use the configuration assistant?

*The configuration assistant gives you one place to store your configuration data. Then, have one job to process them. Then, have one output PDS, the CONFLIB data set, where all your JCLs, start procedures, and parameter files are stored. The number of configuration variables is as low as absolutely necessary and you benefit from the experience of the IBM Böblingen lab.*

13. How can you make your DSIPARM and the configuration assistant options file enterprise-wide?

*By using system symbols*

## Summary

Now that you have completed this unit, you should be able to perform the following tasks:

- Plan for SysOps customization
- Identify the initial z/OS customization requirements
- Describe the initial configuration requirements for the OS Logger stream, VTAM, and OMVS
- Describe the initial configuration requirements for the automation manager
- Describe the initial configuration requirements for the automation agent and NetView
- Define security and authorization using NetView, System Automation for z/OS and your SAF product
- Use the **configuration assistant** to automate majority of configuration process

© Copyright IBM Corporation 2019

### Summary

Now that you have completed this unit, you should be able to perform the following tasks:

- Plan for SysOps customization
- Identify the initial z/OS customization requirements
- Describe the initial configuration requirements for the OS Logger stream, VTAM, and OMVS
- Describe the initial configuration requirements for the automation manager
- Describe the initial configuration requirements for the automation agent and NetView
- Define security and authorization using NetView, System Automation for z/OS and your SAF product
- Use the configuration assistant to automate majority of configuration process





# Unit 3. Customization dialog and Policy Data Base

---

IBM System Automation for z/OS 4.1



## Unit 3 Customization dialog and Policy Data Base



© Copyright IBM Corporation 2019

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

This unit introduces the SA z/OS customization dialog and Policy Data Base (PDB). 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.

You learn about simple policy definitions for systems and applications, importing policies from add-on policies, and the types of reports that can be generated.

### References

*SC34-2717 System Automation for z/OS Version 4 Release 1 Defining Automation Policy*

[https://www.ibm.com/support/knowledgecenter/SSWRCJ\\_4.1.0/com.ibm.safos.doc\\_4.1/manuals.html](https://www.ibm.com/support/knowledgecenter/SSWRCJ_4.1.0/com.ibm.safos.doc_4.1/manuals.html)

## Objectives

---

When you complete this unit, you can perform the following tasks:

- Navigate the ISPF customization dialog
- Create and manage new Policy Data Bases
- Create and link policy objects in the Policy Data Base
- Copy, import, and rename Policy Data Base entries
- Generate Policy Data Base reports

### *Objectives*

This unit introduces the SA z/OS customization dialog.

You learn to:

- Navigate the ISPF customization dialog
- Create and manage new Policy Data Bases
- Create and link policy objects in the Policy Data Base
- Copy, import, and rename Policy Data Base entries
- Generate Policy Data Base reports

# Lesson 1 Customization dialog

## Lesson 1: Customization dialog

- Customization dialog
- Customization dialog: Process
- Policy definition
- Add-on policies

### What this lesson is about

This lesson provides an overview of the System Automation for z/OS version ISPF customization dialog, the policy definition process and add-on policies.

### Objectives

After completing this lesson, you should be able to navigate and describe the System Automation for z/OS version ISPF customization dialog.

### References

*SC34-2717 System Automation for z/OS Version 4 Release 1 Defining Automation Policy*

# Customization dialog

- Is a separate ISPF application that runs under TSO
  - Is used for defining policy definitions for SA z/OS across the enterprise
  - Is used for building configuration files from the policy definitions
- Can build the automation configuration files (SOCNTL) with these items:
  - Automation manager configuration (AMC) for automation managers
  - Automation control file (ACF) for automation agents
  - NetView automation table (AT)
  - NetView message revision table (MRT)
  - z/OS MPF list member
- Can generate several reports
- Can run on any system
- Can be used to automatically convert policy from previous SA z/OS releases

© Copyright IBM Corporation 2019

3-4

## Customization dialog

The customization dialog is used to create and edit the policy for the enterprise. The policies are then used to build the automation configuration files used by the automation manager, automation agent, NetView and z/OS (message processing facility [MPF] member).

To complement the automation specifications in the policy, statements are added to an appropriate Automation Table (AT) and message revision table (MRT). These statements identify messages that indicate changes in status of subsystems, and other events used by additional user automation. The implementation of Automation Table entries is simplified with the dynamic Automation Table build function, also known as Easy Message Management. This topic is covered in Unit 5 “[Automation Tables](#)” on page 388. The dialog does not have to run on the same system where the SA z/OS applications run. They create files, which can then be transported to the systems where they must be available.

# Policy process

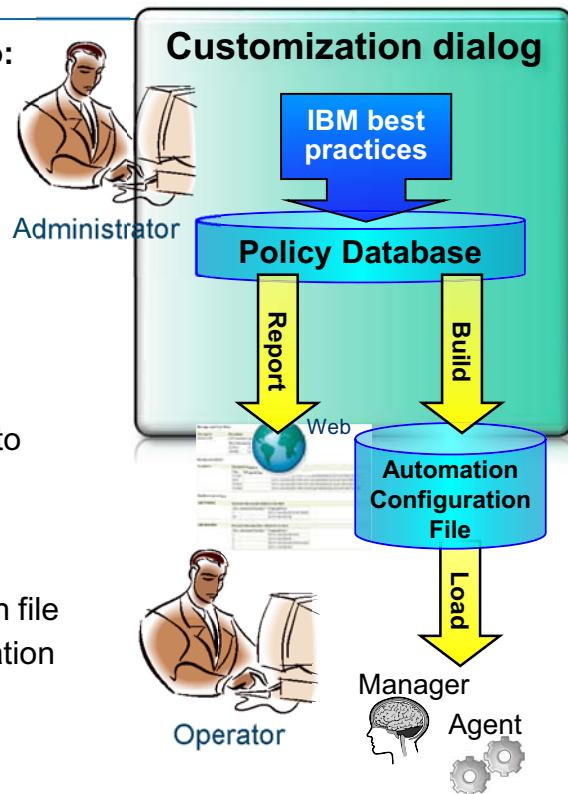
**Administrator uses customization dialog to:**

- Import from IBM best practices policy into **Policy Database**
- Define enterprise-wide shareable and cloneable automation policy
- Create policy reports for documentation
- Build the automation configuration file

**Operator** requests from automation manager to load an automation configuration file

**Automation manager**

- Loads the specified automation configuration file
- Orders the automation agent to load automation configuration file



© Copyright IBM Corporation 2019

3-5

## Policy process

Using the **customization dialog**, the system administrator creates and updates entries in a **Policy Data Base (PDB)**.

The system administrator can import from **IBM best practices policy** into the Policy Database. An **enterprise-wide shareable and cloneable** automation policy is recommended.

The system administrator can create **policy reports** for documentation.

The **build process** takes the entries in the PDB and creates the following items:

Then the operator requests from the automation manager to **load the automation configuration file** using the INGAMS command.

The automation manager loads the specified automation configuration file and orders the automation agent to load automation configuration file.

## Policy contains policy objects linked by WHERE USED

- Policy Data Base contains these policy objects:
  - Enterprise
  - Groups (SAplex)
  - Systems
  - Application Groups
  - Applications and : Sysplex and system
  - Service Periods
  - Triggers and Events
  - Monitor resources
  - ...
- **WHERE USED** links policy objects to the related policy object(s)
- Build process creates only linked policy objects

© Copyright IBM Corporation 2019

3-6

### *Policy contains policy objects linked by WHERE USED*

The policy definition creates many policy objects. A policy object holds the automation information about a discrete part of the automation environment. There are many different types of policy objects. For example, they can include an enterprise, SAplexes, systems, applications, application groups, timers, default specifications, service periods, triggers, or monitor resources.

Each policy object contains a set of attributes that are specific to that type of object. System attributes include system name, console definition, and NetView domain name, for example. Application attributes include job name, dependencies, start commands, shutdown commands, and shutdown delay, for example.

The **WHERE USED** attributes indicate which policy objects are built for which systems and sysplexes.

The build process combines the policy definitions together to create the automation configuration file. The automation configuration members for the AM cover a SAplex; the agent automation configuration members have a system-level scope. The AT and MRT members depend on the scope requested. However, only one MPF list member is generated.

## Policy Data Base

- Is a partitioned data set (PDS) that holds a set of ISPF tables.
- Is created and accessed by using customization dialog.
  - Is started from ISPF panels or INGDLG command.
    - INGDLG is a standard TSO CLIST to invoke SA z/OS Dialog.
    - INGDLG can be used from option 6 with no additional customization.
    - EXEC 'ING.SINGIREX(INGDLG)' 'HLQ(ING)'
- Contains all policy objects for all systems in the enterprise.

### Policy Data Base

The PDB can contain all the policy objects for all systems and SAplices in the enterprise. You can put all definitions for an enterprise into one PDB, although it is not required. All systems controlled by one automation manager (XCF group) must be in one PDB. Where there are common definitions that apply to many systems, a single update can apply to all those systems. However, where there are several systems in a sysplex group, the definitions for each of those systems must be in the same PDB.

The PDB is a set of ISPF tables used only by the customization dialog. The PDB is not used in any way by the operational SA z/OS applications (automation agent or automation manager). For this reason, you can administer the PDB and build your configuration files on a single system in the enterprise. It does not even have to be a system that runs NetView.

INGDLG is a standard TSO CLIST, which is included as part of the SA z/OS product. You can use INGDLG without any further customization from option 6 of ISPF. Alternatively, the system programmer can link the customization dialog (INGDLG) to the standard ISPF menu structure.

## Customization dialog primary menu

```

MENU OPTIONS HELP
System Automation for z/OS 4.1 Customization Dialog
Option ==> _____ More: +
0 Settings      User parameters
BR Browse        Browse the Policy Database
1 Edit           Edit the Policy Database
2 Build          Build functions for Policy Database
3 Report         Generate reports from Policy Database
4 Policies       Maintain Policy Database list
5 Data Management Import policies into a Policy Database
U User           User-defined selections
X Exit          Terminate Customization Dialog

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

```

To open a PDB, choose one of the following options:

- Specify the PDB name in the Current Policy Database field, select option 1, and press Enter.
- Select option 4 to display a list of all PDBs, and select a PDB from the list.

© Copyright IBM Corporation 2019

3-8

### *Customization dialog primary menu*

The SA z/OS 4.1 customization dialog Primary Menu is displayed when the dialogs are started. You can select several options from the list and use the pull-down commands at the top of the panel.

This panel automatically displays the name of the PDB that was last used by the TSO user. The PDB name is stored in member AOFTPDB of the ISPF table output library, so that you can use it again immediately if you want. The first time you access this panel, the **Current Policy Database** field is blank.

# Customization dialog settings

The screenshot shows the SA z/OS customization dialog. A blue arrow points from the 'Color' option in the primary menu to the 'Inherited Data Color Selection' panel.

```

MENU HELP
Option ===> [ ] Settings Menu

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 apply only to your userid:
10 Policy
11 Dsnames
12 Workdsn
13 JobStep

COMMANDS HELP
Command ===> [ ] Inherited Data Color Selection

Select a color to display inherited class values on data panels:
White
S Yellow
Green
Pink
Red
Blue

Display the selected color in reverse video . . . NO (YES NO)

```

**By default, inherited class data is displayed as yellow.**

© Copyright IBM Corporation 2019

3-9

## Customization dialog settings

Option zero (**0**) from the primary menu displays the Settings Menu. From this panel, you can perform these tasks:

- Define aliases for the SA z/OS commands.
- Define the display characteristics of the panels.
- Define the message and logging options.
- Define options for PDB conversion.
- Define the color for inherited values on policy data panels.
- Modify the SA z/OS policy definitions.
- Specify the data sets to be used by SA z/OS as defaults and in the batch build and migrate processes.
- Define temporary work data sets to ISPF for using batch functions such as the build process.

## Sample add-on policies

- \*EMPTY is simply an empty policy database
- Out of the box, sample add-on policies
- Blue ones are customizable add-ons that can be customized to use a subset of the policy
- For information about the contents of the sample add-on policies, use the view (v) action on the Create a New Policy Database panel

\*EMPTY  
\*BASE  
\*CICS  
\*DB2  
\*E2E  
\*GDPS  
\*IBMCOMP  
\*IMS  
\*ITM  
\*IWS  
\*ProcOps  
\*SAPSRV  
\*TBSM

© Copyright IBM Corporation 2019

3-10

### PDB samples and add-on policies

This slide shows the sample PDBs and PDB add-ons that are included with SA z/OS. \*BASE and \*EMPTY are sample PDBs. \*BASE contains many of the base policy definitions. By using the \*BASE PDB, you can add policy definitions that are unique to the Applications by selecting the add-on policies related to the Applications, for example, DB2, IMS, CICS, ITM, or WebSphere. The \*EMPTY PDB does not contain any automation policy.

For information about the contents of the sample add-on policies, use the view (v) action on the Create a New Policy Database panel. \*IBMCOMP contains policy for SA z/OS IO Operations, z/OS Common Information Model (CIM) Server, and the NetView Event/Automation Services (EAS).

Add-on policies can be customized to include a subset of the policy. For example, suppose you are asked to create policies for OMEGAMON XE for z/OS. Using the customization dialog, you can customize the import of the \*ITM add-on policy to only contain OMEGAMON XE for z/OS from a total of 10 selectable components:

| Action | Status   | Component               |
|--------|----------|-------------------------|
|        |          | OMEGAMON - NetView z/OS |
|        | SELECTED | OMEGAMON - z/OS         |

When you customize an add-on policy, you can select only the policies that you require. Later in the import process, you can delete specific entries.

## Add-on policies

- Add-on policies are designed to be added on to the \*EMPTY PDB or an existing user PDB
- Add-on policies contain relationships to \*BASE or other add-on policies
- Add-on policies might contain duplicate entries  
For example, a subset of DB2 is required for SAP and WebSphere
- Some add-on policies are customizable to import only a subset, such as \*BASE, \*E2E, \*IBMCOMP, \*ITM

The add-on policies for DB2, IMS, CICS, ITM, or WebSphere, can be added to the \*BASE PDB or to an existing PDB. Several of the add-on policies are customizable. They contain more than one set of policies. You can select a subset of the policies when you import them.

# Lesson 2 Creating a Policy Data Base

## Lesson 2: Creating a Policy Data Base

- Creating a Policy Data Base
- PDB documentation
- Policy PDF files
- New Policy Data Base contents

© Copyright IBM Corporation 2019

3-12

*Creating a Policy Data Base*

### What this lesson is about

This lesson describes how to create a System Automation for z/OS 4.1 Policy Data Base.

### Objectives

After completing this lesson, you should be able to create a System Automation for z/OS 4.1 Policy Data Base.

### References

*SC34-2717 System Automation for z/OS Version 4 Release 1 Defining Automation Policy*

# Creating a Policy Data Base

| MENU                      | COMMANDS         | ACTIONS                        | VIEW | HELP                               |
|---------------------------|------------------|--------------------------------|------|------------------------------------|
| Policy Database Selection |                  |                                |      |                                    |
| Command ==> new           |                  |                                |      | Row 8 to 15 of 42<br>SCROLL==> CSR |
| Action                    | Policy Database  | Enterprise Name/Data Set Name  |      |                                    |
|                           | OPSU1EX1         | 'SAZOS.PDB.OPSU1EX1'           |      |                                    |
|                           | OPSU1EX1_INSTR   | 'OPSU1EX1_INSTR'               |      |                                    |
|                           | OPSU3EX1         | 'SHZOS.PDB.OPSU3EX1'           |      |                                    |
|                           | OPSU3EX1_INSTR   | 'OPSU3EX1_INSTR'               |      |                                    |
|                           | OPSU3EX1_OLD     | 'SAZOS.PDB.OPSU3EX1_INSTR'     |      |                                    |
|                           | OPSU3EX1_OLD_INS | 'SA340PSU3EX1'                 |      |                                    |
|                           | OPSU4EX1         | 'SAZOS.PDB.OLD.OPSU3EX1'       |      |                                    |
|                           | OPSU4EX1_INSTR   | 'SAZOS.PDB.OLD.OPSU3EX1_INSTR' |      |                                    |
|                           |                  | 'OPSU4EX1'                     |      |                                    |
|                           |                  | 'SAZOS.PDB.OPSU4EX1'           |      |                                    |
|                           |                  | 'OPSU4EX1_INSTR'               |      |                                    |
|                           |                  | 'SAZOS.PDB.OPSU4EX1_INSTR'     |      |                                    |

Enter NEW to define a new PDB

Option 4 from Primary Menu. Initially, there are no PDBs defined.  
Enter **NEW** to define a new PDB, or **ADD** to add an existing PDB.

The first time that you enter the customization dialog, you do not have a current PDB. You can choose to create a PDB or use an existing PDB.

To create a new PDB, use the **new** (abbreviated as **N** or **n**) command. To add an existing PDB to the list, use the **ADD** command. If you add an SA z/OS 3.4 or 3.5 PDB, the back level PDB is automatically converted to the new format. You are prompted to verify that this is your intent.

If you select option 4 to see your list of PDBs, you see the panel shown in this slide. You can see that there are no PDBs that you worked with before. The list grows as you select PDBs to work with. This list is a personal list; it is also saved in your ISPF profile. If you create a new PDB, it is displayed on your list in future TSO logons. Other TSO users do not see this PDB in their list, unless they use the **ADD** command to add the PDB to their personal list.

## Allocate the Policy Data Base (1 of 2)

| COMMANDS  |        | ACTIONS  | HELP         |
|---|--------|--|--------------|
| AOFGPDBN  |        | Create a New Policy Database                     |              |
| Command ==> _____ Row 1 to 8 of 13                                  |        |  |              |
| To define a new Policy Database, specify the following information: |        |  |              |
| Policy Database Name . . . NEW_PDB                                  |        |  |              |
| Enterprise Name. . . . . saplex                                     |        |  |              |
| Data Set Name. . . . . 'sazos.pdb.test'                             |        |  |              |
| Model Policy Database. . . *EMPTY                                   |        | Policy Database name or "?"<br>for list of names |              |
| Add-on policies to be added to a standard SA model policy database: |        |  |              |
| Action  | Status | Add-on Policy                                    | Customizable |
| S   |        | *BASE  | YES          |
|   |        | *CICS  |              |
|   |        | *DB2   |              |
|   |        | *E2E   | YES          |
|   |        | *GDPS  |              |
|   |        | *HYPERSWAP                                       |              |
|   |        | *IBMCOMP   | YES          |
|   |        | *IMS   |              |

Define PDB  
name, data set,  
enterprise

Empty or existing PDB

Select \*BASE  
add-on policy

Select the \*EMPTY PDB plus \*BASE add-on policy, and press Enter.

© Copyright IBM Corporation 2019

3-14

### Allocate the Policy Data Base (1 of 2)

On this panel, you specify the Policy Data Base name (NEW\_PDB), select an enterprise name (SAPLEX), and define the data set name (ING330.PDB.ADMIN). The PDB name is the name by which you access the PDB from your list of PDBs. It is displayed in the **Current Policy Database** field on the primary menu panel, AOFGPRM.

You must select a PDB to use, such as \*BASE or an existing PDB. Additionally, you can select one or more add-on policies to import into the new PDB. You can also import add-on policies later. You can view more descriptive information specific to each add-on by entering V (view) in the Action column.

## Allocate the Policy Data Base (2 of 2)

| COMMANDS HELP   |                |
|---|----------------|
| New Policy Database Dataset Information                   |                |
| Command ==>   |                |
| New Policy Database : NEW_PDB                             |                |
| Attributes to be used for allocation of the new data set: |                |
| Managed storage . . . . .                                 | <u>NO</u>      |
| Management class . . . . .                                |                |
| Storage class . . . . .                                   |                |
| Volume serial . . . . .                                   |                |
| Data class . . . . .                                      |                |
| Space units . . . . .                                     | <u>TRACKS</u>  |
| Primary quantity . . .                                    | <u>50</u>      |
| Secondary quantity . .                                    | <u>50</u>      |
| Directory blocks . . . .                                  | <u>50</u>      |
| Record format . . . . :                                   | FB             |
| Record length . . . . :                                   | 80             |
| Block size . . . . .                                      | <u>32720</u>   |
| Data Set Name type . .                                    | <u>LIBRARY</u> |
| Device Type . . . . .                                     |                |
| More:   | +              |
| YES      NO   |                |
| Blank for default management class                        | *              |
| Blank for default storage class                           | *              |
| Blank for authorized default volume                       |                |
| Blank for default data class                              | *              |
| CYLS TRKS BLKS KB MB                                      |                |
| 1 to 999 - In above units                                 |                |
| 0 to 999 - In above units                                 |                |
| 1 to 999  |                |
| LIBRARY    PDS  |                |

Verify the data set allocation information for NEW\_PDB and press Enter to continue.

### Allocate the Policy Data Base (2 of 2)

This panel contains the data set information that is necessary to allocate the data set for your PDB. Verify the information, including paging forward with PF8 to see additional allocation parameters. At a minimum, specify **LIBRARY** or **PDS** for the **Data Set Name** type field. PDS allocates the PDB as a z/OS data set. LIBRARY allocates the PDB as a PDSE. Press Enter to allocate the PDS.

Typically, a small environment requires five cylinders for the PDB and a large environment requires 20 cylinders. Any values that you enter on this panel are saved for the next PDB that you allocate. Status messages are displayed in the Command Progress Window panel during the creation of the PDB.

# Entry Type Selection panel

| MENU HELP      |                       | Entry Type Selection |                      |
|----------------|-----------------------|----------------------|----------------------|
| Option ==> [ ] |                       | More: -              |                      |
| 1 ENT          | Enterprise            | 30 TMR               | Timers               |
| 2 GRP          | Groups                | 32 TPA               | Tape Attendance      |
| 3 SBG          | SubGroups             | 33 MVC               | MVS Components       |
| 4 SYS          | Systems               | 34 MDF               | MVSCOMP Defaults     |
| 5 APG          | ApplicationGroups     | 35 SDF               | System Defaults      |
| 6 APL          | Applications          | 36 ADF               | Application Defaults |
| 7 EVT          | Events                | 37 AOP               | Automation Operators |
| 8 SVP          | Service Periods       | 38 NFY               | Notify Operators     |
| 9 TRG          | Triggers              | 39 NTW               | Networks             |
| 10 PRO         | Processors            | 40 XDF               | Sysplex Defaults     |
| 11 MTR         | Monitor Resources     | 41 RES               | Resident CLISTS      |
| 12 ENS         | zEnterprise Ensembles | 42 SCR               | Status Display       |
| 13 PAC         | Pacing Gates          | 50 DMN               | Remote Domains       |
| 20 PRD         | Product Automation    | 51 REF               | Resource References  |
| 21 MSG         | Messages              | 99 UET               | User E-T Pairs       |

When PDB creation is complete, the Entry Type Selection panel is displayed. All imported policy, for example, \*BASE is now accessible from this panel.

## Entry Type Selection panel

The Entry Type Selection panel (AOFGETYP) displays a list of the policy object types that you can define within a PDB. You can select a policy object by entering its three-character identifier or numeric value.

Option **20** (PRD, Product Automation) contains additional automation policy for product-specific automation (CICS, IMS, and TWS). Option **99** (UET, User Entry-Type pairs) is used to define free-format user data to be used for a globally dispersed parallel sysplex (GDPS).

The customization dialog allows editing within all 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 a second user tries to update the same policy entry that is being updated by another user, the second user can access the policy on a read-only basis. The Entry Type Selection panel indicates which entry types can be used by multiple users by flagging them with an asterisk (\*).

Multiple users who access the policy entries might affect the bulk updates. Any user who does a build must have exclusive access to the PDB.

Several ISPF commands are helpful when you use the customization dialog:

- **pfshow off:** Turn off the display of PF keys. Displaying the PF key settings can mask some of the options available to you.

- **panelid:** Enable the display of the panel ID, which is useful when navigating between panels.
- **msgid on:** Displays the message ID when you request help (**PF1**) for a message.

# PDB documentation

| Policy Selection           |             |  | Row 1 to 7 of 7          |
|----------------------------|-------------|--|--------------------------|
|                            |             |  | SCROLL==> CSR            |
| Entry Type : Enterprise    |             |  | PolicyDB Name : NEW_PDB  |
| Entry Name : SAPLEX        |             |  | Enterprise Name : SAPLEX |
| Action                     | Policy Name | Policy Description                           |                          |
| DESCRIPTION                |             | Enter description                            |                          |
| SEND COMMAND OPERS         |             | Define Operator Profile for sending commands |                          |
| INGSEND PARMs              |             | Define INGSEND Command ParmS                 |                          |
| PROCOPS FOCAL POINTS       |             | Define processor operations focal point info |                          |
| AUTO MSG CLASSES           |             | Define Auto Msg Classes                      |                          |
| SYMBOL DESCRIPTION         |             | Enter system automation symbol description   |                          |
| DOCUMENTATION              |             | Overall Policy Database documentation        |                          |
| ***** Bottom of data ***** |             |  |                          |



Review the documentation under the ENTerprise policy definition (option 1).

## PDB documentation

The ENTerprise policy (option 1) contains an entry for DOCUMENTATION. Select this option to read help that is related to the \*BASE and add-on policies provided by SA z/OS.



The entries that are displayed on the Policy Selection panel are called *policy items*. You see this panel for several resource types, such as Applications.

## PDB documentation details

```
*BASE Sample Policy  
=====
```

This policy describes a sample computing environment containing a sysplex (SYSPLEX1) with 3 systems (SYS1, SYS2, SYS3) and typical z/OS applications.

This policy allows to select subcomponents. Those are:

- Base z/OS
- Job Entry Subsystem 2 (JES2)
- Job Entry Subsystem 3 (JES3)
- Base Communication Services (USS)
- File Transfer Protocol Daemon (FTPD)
- IBM HTTP Server
- Network File System Server and Client (NFS)
- Secure Post-Processing
- TELNET
- Dynamic

When adding this policy to your PDB you may need to adapt its policy definitions to your company's naming rules.

It is recommended that you create an Enterprise Report when you start to adapt names. This report can help you to find all the sections that must be changed accordingly.

After this policy was added to a new or existing PDB certain adaptation is required.

### Groups:

- SYSPLEX1 : - You may rename the sysplex policy entry name according to your environment.  
- Also change the sysplex name in the SYSPLEX policy to the physical sysplex name.

© Copyright IBM Corporation 2019

3-18

### PDB documentation details

This DOCUMENTATION policy item contains information about the \*BASE PDB and the add-on policies that are available.

## Policy PDF files

- PDF files for base and add-on policies are in SA Wiki only
- In the PDF files, you can view sample resources and relationships

IBM System Automation

Version 4 Release 1

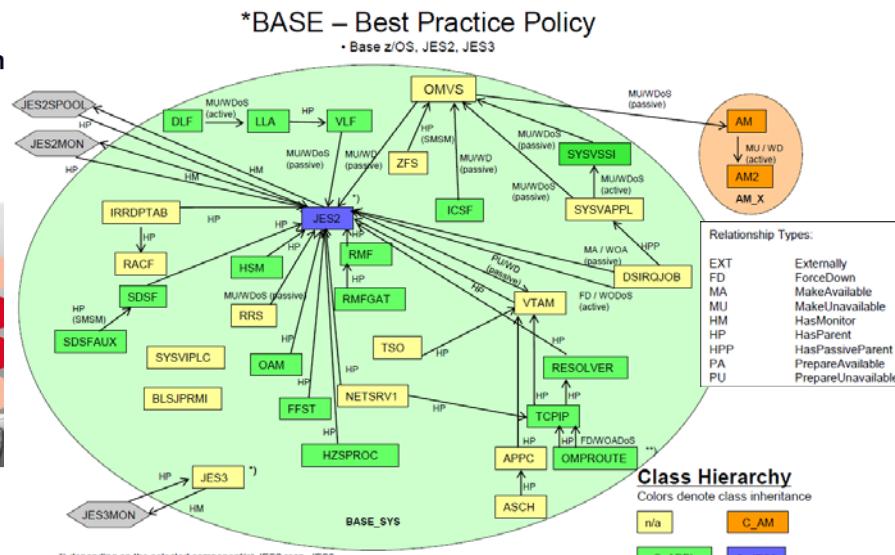


\*BASE – Best Practice Policy

Sample resources and relationships are displayed

**Located in USS:**

[/usr/lpp/ing/doc/policies/](#)



© Copyright IBM Corporation 2019

3-19

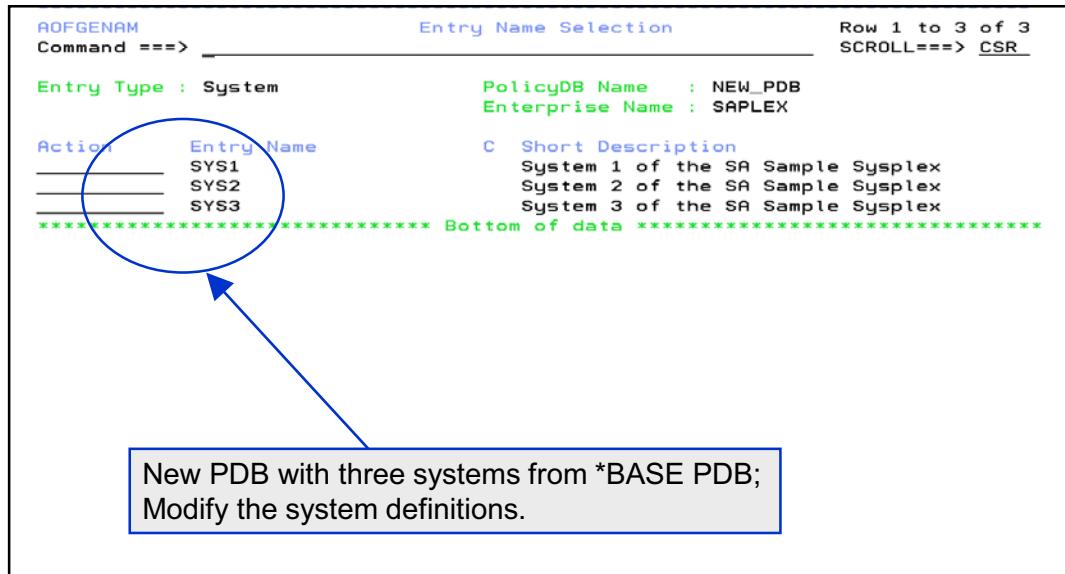
Policy PDF files

Diagrams of the add-on policies are also provided as PDF files, located in SA z/OS Wiki at:

<https://www.ibm.com/developerworks/community/wikis/home?lang=en#!wiki/Tivoli+System+Automation/page/Best+Practices>

You can download the files and view them to see, for example, the structure, Applications, Application Groups, or relationships.

## New PDB contents



Begin customization at the system level (option 4, SYS).

### New PDB contents

This Entry Name Selection panel is for the newly created PDB. You can access this panel from the main panel from option 4 (SYS). Because this PDB was created using the \*BASE sample PDB, three systems are defined: SYS1, SYS2, and SYS3. Begin your customization by defining the systems in your environment. Type **S** (Select) in the Action column next to the system name and press Enter to edit the policy for each system.

# Lesson 3 Introduction to defining automation policy

## Lesson 3: Introduction to defining automation policy

Use the NEW\_PDB

- Modify a system entry (SYS): SYS2
- Modify an Application entry (APL): SYSVSSI
  - Relationships to other Applications
  - Startup and shutdown requirements: When to start, how to stop
  - WHERE USED policy, such as systems or Application Groups

### What this lesson is about

In this lesson, you learn basic automation policy definitions using the new PDB that was created. For example, three systems (SYS1, SYS2, and SYS3) were created when the PDB was created. You must modify the system policies to match your environment. This lesson provides an introduction to defining basic System Automation for z/OS 4.1 policy.

### Objectives

After completing this lesson, you should be able to describe how to define basic System Automation for z/OS 4.1 policy.

### References



## Modify a system entry

- Select option **4 (SYS)** to view system entries
- Rename system policy from SYS2 to MVSB

| MENU HELP   |     | Entry Type Selection |
|---|-----|----------------------|
| A0FGETYP  |     | Option ==> _____     |
| Enter number or entry type or use "BR <entry type>" |     |                      |
| 1   | ENT | Enterprise           |
| 2   | GRP | Groups               |
| 3   | SBG | SubGroups            |
| 4   | SYS | Systems              |
| 5   | APG | ApplicationGroups    |
| 6   | APL | Applications         |
| 7   | EVT | Events               |
| 8   | SVP | Service Periods      |
| 9   | TRG | Triggers             |
|   |     | 30 TMR               |
|   |     | 32 TPA               |
|   |     | 33 MVC               |
|   |     | 34 MDF               |
|   |     | 35 SDF               |
|   |     | 36 ADF               |
|   |     | 37 AOP               |
|   |     | 38 NYF               |
|   |     | 39 NTW               |

- Modify the system policy for MVSB by modifying the SYSTEM INFO policy:
  - System name
  - SDF root name
  - NetView domain name
- Press **PF3** to save

© Copyright IBM Corporation 2019

3-22

### Modify a system entry

To define a new system or modify an existing system policy, perform either of the following actions:

- Enter **SYS** on the command line.
- Select option **4** from the Entry Type panel, A0FGETYP.

## Rename system entry

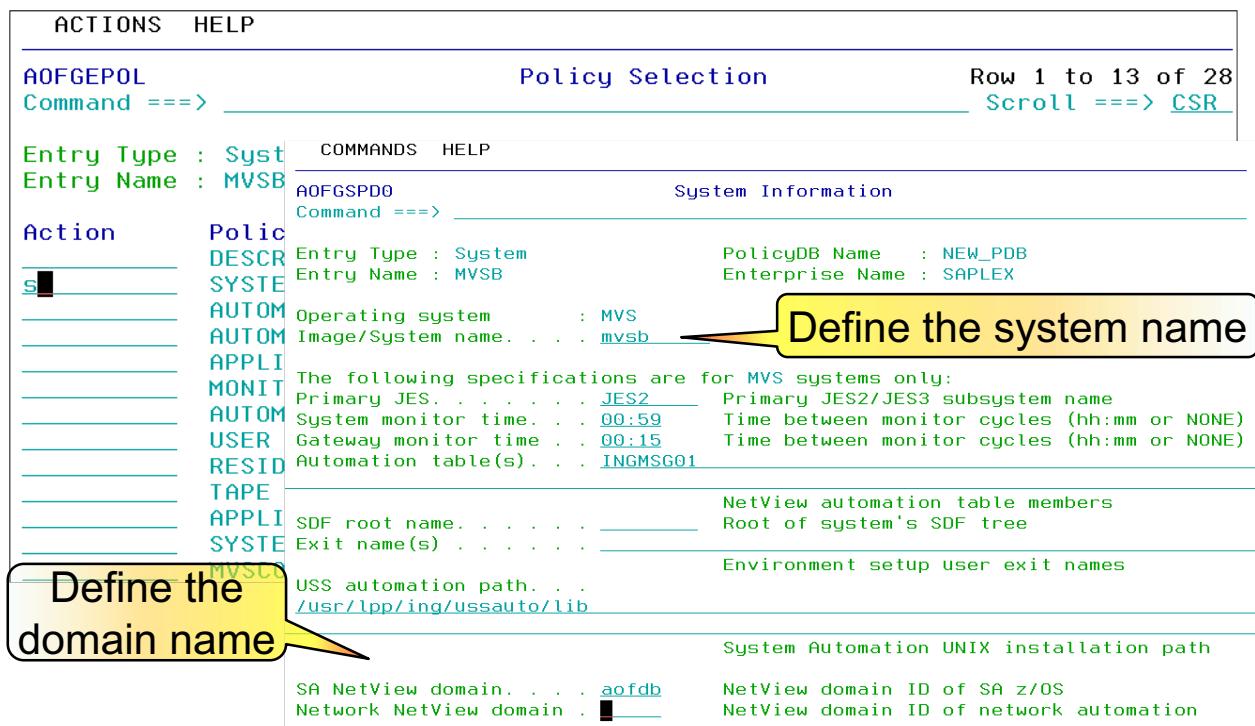


| AOFGENAM  |            |  | Entry Name Selection                            | Row 1 of 3 |
|---|------------|--|---|------------|
| Entry Type : System   |            |  | PolicyDB Name : NEW_PDB                         |            |
|   |            |  | Enterprise Name : SPLEX                         |            |
| Action  | Entry Name |  | C Short Description                             |            |
| SYS1  |            |  | System 1 of the SA Sample Sysplex               |            |
| R   | SYS2       |  | System 2 of the SA Sample Sysplex               |            |
| SYS3  |            |  | System 3 of the SA Sample Sysplex               |            |
| *****   |            |  | *****   | *****      |
| AOFGPREN  |            |  | Entry Rename                                    |            |
|   |            |  | Description : System 2 of the SA Sample Sysplex |            |
|   |            |  | Old Name : SYS2                                 |            |
|   |            |  | New Name . . . MVSB_                            |            |
| Press ENTER to rename member.<br>Press CANCEL to cancel rename. |            |  |   |            |

Use action **R** to rename SYS2 to MVSB.

Use the **R** row command to rename a system entry. Press **F1** for help on possible row commands.

# SYSTEM INFO policy item



Animated

© Copyright IBM Corporation 2019

3-24

## SYSTEM INFO policy item

Select the SYSTEM INFO policy item for system MVSB and press Enter to customize the policy. You see the System Information panel with the following fields:

- **Entry Name:** The name that you want this system known as. This name does not have to be the system name, and can be up to 20 characters long without blanks.
- **Image/System name:** Enter the name that is defined by the system programmer for this system. To determine your system name, issue a DISPLAY SYMBOLS command at the z/OS console. Look for the &SYSNAME. symbol.
- **Automation Table(s):** The agent can use multiple Automation Tables. The tables are loaded by SA z/OS in the order specified in the panel. INGMSG01 is the AT generated by the dynamic Automation Table build feature of SA z/OS.

If you use several tables for SA z/OS, all must be specified, including INGMSG01. Only specify your SA z/OS Automation Tables here. If you load other Automation Tables, in CNMSTYLE, for example, do not specify them here.

- **SDF Root Name:** This field must match the root name you define in your SDF tree structure so that status updates are displayed correctly. It defaults to the system name.
- **Exit name(s):** In this field, you can identify user exits that will run when agent initialization is done. Be careful if you use these exits. Not all agent facilities are available at the time the exit

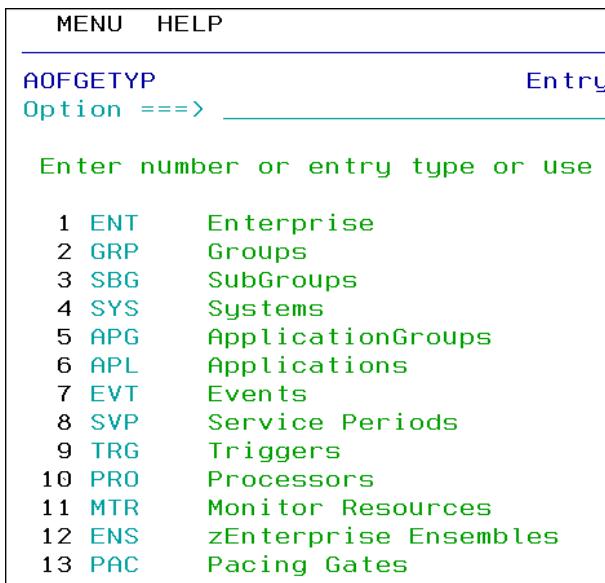
REXX EXEC starts execution. (If you want to ensure that the agent is fully initialized, use the AOF540I message, remembering that it is issued each time an ACF is loaded. You can also use the AOFEXINT exit.)

- **USS automation path:** Defines the path where SA z/OS can find its z/OS UNIX System Services programs.
- **SA NetView Domain:** Enter the NetView domain ID of the automation agent.

Press PF8 to define more parameters, such as the network NetView domain. These entries define a basic system object. There are many more entries that can be defined.

## Modify an application

- Select option **6** (APL)
- Modify the sample SYSVSSI (NetView SSI) Application



- Example definition changes:

- Rename to AUTOSSI
- Use the default start command:  
MVS S AUTOSSI
- Define two passes for shutdown:
  - First pass: Issue MVS P AUTOSSI to stop
  - Second pass: If that does not work after one minute, issue MVS C AUTOSSI

- Define relationships:

- As AUTOSSI has no relationships we define a HasParent from TSO to VTAM
- TSO started when VTAM is UP
- VTAM is defined

© Copyright IBM Corporation 2019

3-25

### Modify an Application

To modify an Application, enter **APL** or **6** on the Entry Type Selection panel. Using the newly created PDB, modify the AUTOSSI Application. Its basic automation attributes are shown. AUTOSSI will have two shutdown passes and a dependency with JES2, for example.

## Rename SYSVSSI

```
COMMANDS ACTIONS VIEW HELP
AOFGENAM          Entry Name Selection      Row 37 to 57 of 57
Command ===>                               Scroll ==> CSR

Entry Type : Application          PolicyDB Name   : NEW_PDB
                                         Enterprise Name : SAPLEX

Action       Entry Name           C Short Description
A0FGPREN     Entry Rename
Description : Automation Subsystem Interface
Old Name    : SYSVSSI
New Name    . . . autossi_____
Press ENTER to rename member.
Press CANCEL to cancel rename.
F1=Help     F2=Split   F3=End      F9=Swap     F12=Cancel

R           SYSVSSI             Automation Subsystem Interface
                         TCP/IP
                         TN_PORT          Port which is used by TELNET server
```

Use Action **R** to rename SYSVSSI to AUTOSSI.

© Copyright IBM Corporation 2019

3-26

### Rename SYSVSSI

The first step in this process is to rename the SSI Application from the default of SYSVSSI to AUTOSSI. Use an action of **R** (abbreviation for RENAME) for SYSVSSI to rename the entry. The Entry Rename window opens, where you can specify AUTOSSI in the New Name field.

The Entry Name for the Application is renamed. It does not change any policy for the Application, such as job name. You might have to edit the policy items. For example, no references in the Description fields are changed. No other fields are changed (such as subsystem name or job name of the Application). You can rename the enterprise name using EDIT, which is available on the PDB selection list (option **4**).

## Policy Selection panel: Application

| ACTIONS HELP             |                      | Policy Selection                              | Row 1 to 21 of 21<br>Scroll ===> CSR |
|--------------------------|----------------------|---|--------------------------------------|
| AOFGEPOL                 | Command ==>          |   |                                      |
| Entry Type : Application | Entry Name : AUTOSSI | PolicyDB Name : NEW_PDB                       |                                      |
| Action                   | Policy Name          | Policy Description                            |                                      |
| _____                    | DESCRIPTION          | Enter description                             |                                      |
| _____                    | UPWARD CLASS         | Select a class to inherit data from           |                                      |
| _____                    | APPLICATION INFO     | Define application information                |                                      |
| _____                    | AUTOMATION FLAGS     | Define application automation flags           |                                      |
| _____                    | APPLICATION SYMBOLS  | Define application symbols                    |                                      |
| _____                    | TRIGGER              | Select trigger                                |                                      |
| _____                    | SERVICE PERIOD       | Select service period                         |                                      |
| _____                    | PACING GATE          | Select pacing gate                            |                                      |
| _____                    | RELATIONSHIPS        | Define relationships                          |                                      |
| _____                    | MESSAGES/USER DATA   | Define messages and user data                 |                                      |
| _____                    | STARTUP              | Define startup procedures                     |                                      |
| _____                    | SHUTDOWN             | Define shutdown procedures                    |                                      |
| _____                    | THRESHOLDS           | Define error thresholds                       |                                      |
| _____                    | MINOR RESOURCES      | Define application minor flags and thresholds |                                      |
| _____                    | SYSTEM ASSOCIATION   | Define primary and secondary associations     |                                      |
| -----                    | -----RESOURCES-----  | -----RESOURCES-----                           |                                      |
| _____                    | GENERATED RESOURCES  | List resources generated for this entry       |                                      |
| _____                    | MEMBER OF            | List resources where this entry is a member   |                                      |
| -----                    | -----                | -----   |                                      |
| _____                    | WHERE USED           | List application groups linked to this entry  |                                      |
| _____                    | COPY                 | Copy data from an existing entry              |                                      |

© Copyright IBM Corporation 2019

3-27

### Policy Selection panel: Application

This Policy Selection panel is for the AUTOSSI Application. The panel displays the PDB name (NEW\_PDB) and enterprise name (SAPLEX) that you are working with. Type an **S** in the Action column on the APPLICATION INFO line to see the Application Information definition panel shown on the next slide. The SHUTDOWN, RELATIONSHIPS, and WHERE USED policy items are modified after the APPLICATION INFO policy.

## APPLICATION INFO panel

|                                    |                                     |   |
|------------------------------------|-------------------------------------|---|
| COMMANDS                           | HELP                                |   |
| AOFGDYNA                           | Application Information             | Line 00000001                                     |
| Command ==> _____                  | Scroll ==> <a href="#">CSR</a>      |   |
| Entry Type : Application           | PolicyDB Name : NEW_PDB             |   |
| Entry Name : AUTOSSI               | Enterprise Name : SAPLEX            |   |
| Category . . . . .                 | _____                               | (IBM-defined, user-defined or blank,<br>see help) |
| Subcategory . . . . .              | _____                               | (IBM-defined, user-defined or blank,<br>see help) |
| Subsystem Name . . . . .           | <a href="#">autoSSI</a>             |   |
| Job Type . . . . .                 | <a href="#">autoSSI</a>             | (MVS NONMVS TRANSIENT)                            |
| Job Name . . . . .                 | <a href="#">autoSSI</a> :           | (YES NO)  |
| Transient Rerun . . . . .          | <a href="#">MSTR</a>                | (MSTR, JES Subsystem)                             |
| Scheduling Subsystem . . .         | <a href="#">CNMSJ010</a>            |   |
| JCL Procedure Name . . . . .       |                                     |   |
| Job Log Monitor Interval . . . . . | _____                               | (mm:ss NONE)                                      |
| Captured Messages Limit . . . . .  | _____                               | (0 to 999)  |
| Desired Available . . . . .        | _____                               | (ALWAYS ONDEMAND ASIS)                            |
| Restart after IPL . . . . .        | _____                               | (START NOSTART NONE)                              |
| Monitor for IPL complete . . . . . | _____                               | (YES NO)  |
| Start Delay . . . . .              | _____                               | (time for "UP" status checks, hh:mm:ss)           |
| Start Cycles . . . . .             | ____                                | (start delay checks, 0 to 99)                     |
| UP Status Delay . . . . .          | _____                               | (time to delay "UP" status, hh:mm:ss)             |
| Restart option . . . . .           | <a href="#">abendonly</a>           | (ALWAYS ABENDONLY NEVER)                          |
| External Startup . . . . .         | <input checked="" type="checkbox"/> | (INITIAL ALWAYS NEVER)                            |

Press PF8 to see other policy definitions.

© Copyright IBM Corporation 2019

3-28

### APPLICATION INFO panel

On this slide, you see the main definition panel for an Application. Many Applications require definitions from this panel. The following fields are on this panel:

- **Category:** Identifies IBM-defined or user-defined applications.
- **Subcategory:** Specifies any IBM-defined or user-defined subcategory of the application for which a category was specified.
- **Subsystem Name:** Enter the name that is used by the agent when monitoring and controlling the subsystem. In other words, when you issue INGREQ, this name is part of the resource name. Typically, the subsystem name is the same as the job name. It can be 11 characters long and must be unique among subsystems selected for a system. In this example, the same name, AUTOSSI, is used. This is also known as the AUTOMATION NAME.
- **Job Type:** Because this is an MVS job, specify MVS. This can be omitted, and a default value will be used.
- **Job Name:** Enter the job name (such as the started task name **AUTOSSI**).
- If the JCL Procedure Name is specified, you use the following default start command:

MVS S *JCL\_procedure\_name*,JOBNAME=*jobname*

- If the JCL Procedure Name is not specified, you use the following default start command:

MVS S jobname

- **Job Log Monitor Interval:** Specifies how frequently the Job Log must be monitored. Frequency must be specified by using format mm:ss. The minimum time is one second (00:01); the maximum is 60 minutes (60:00). If the field is blank, the value can be inherited from a class definition. If monitoring must not be done, specify NONE to prevent inheritance. 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.
- **Desired Available:** Defines the default desired status for the resource. ALWAYS (default) means that the automation manager tries to start the resource. ONDEMAND means that the resource must be started by an operator, Service Period, or a request propagated from an Application Group. ASIS sets the desired status to the observed status which means that the resource remains in its current state and no action is taken by SA z/OS unless requested.
- **Restart after IPL:** If the job is to be startable at IPL, even if its Agent Status is BROKEN, STOPPED, or CTLDOWN, then you specify **START**. START is typically correct for base address spaces such as LLA, JES2, and VTAM. For Applications such as CICS, blank is the best choice.
- Specify a Desired Available value of ONDEMAND instead of using Restart after IPL=NOSTART.
- **Monitor for IPL complete:** Specifies whether the application has to be in an AVAILABLE status before IPL is considered to be completed.
- **Start Delay:** After the expiration of the Start Delay, a check occurs to determine whether the application recorded a status of STARTED and/or ACTIVE. Enter a value that represents a reasonable time for the UP message to be displayed.
- **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.
- **UP Status Delay:** Specifies an interval which begins when the UP message is received and ACTIVMSG UP=MSG is triggered. The application will not be set to an UP status before the specified time has expired.
- **Restart Option:** This field indicates under what circumstances a subsystem is restarted automatically by the automation agent. A value of ABENDONLY indicates that it is restarted only if it has a recoverable (restartable) termination.
- **External Startup:** Specifies whether the application is started externally or via a specified startup procedure.

Press PF3 to save your changes. You see the Policy Selection panel.

# Application SHUTDOWN policy

```
COMMANDS HELP
AOFGASHT          Subsystem Shutdown Processing
Command ==> _____
Entry Type : Application      PolicyDB Name   : NEW_PDB
Entry Name  : AUTOSSI         Enterprise Name : SAPLEX
Shutdown Pass Interval . . . (hh:mm:ss)
Line Commands: S/C (Cmd), R (Rep)

Cmd  Phase    Description
____ INIT      Executed when shutdown is initiated
█ NORM      Executed when normal shutdown is invoked
____ IMMED     Executed when immediate shutdown is invoked
____ FORCE     Executed when force shutdown is invoked
____ FINAL     Executed after final termination message

COMMANDS HELP
AOGFDDYNC      Command Processing : SHUTNORM           Line 00000001 Col 001 075
Command ==> █
Mixed case . . . NO (YES NO)
Cmd Ps AutoFn/* Command Text
____ 1 MVS P &SUBSJOB
____ 4 MVS C &SUBSJOB
***** Bottom of data *****
```

A yellow speech bubble on the right side of the screen contains the text: "AUTOSSI SHUTDOWN policy with commands inherited from the class level". Below the speech bubble, there is a small diagram showing two overlapping circles: one blue circle labeled "Cmd" and one orange circle labeled "Rep". To the right of these circles, three numbers are circled: 2, 2, and 1.

© Copyright IBM Corporation 2019

3-29

## Application SHUTDOWN policy

You can define five different shutdown phases. Enter **cmd** (or **select**) in the Action column against the **NORMal** phase to define commands for a normal shutdown of the Application. This example defines one pass for the normal shutdown of AUTOSSI. The first pass issues an **MVS P AUTOSSI** command to stop the Application. In this example, two commands are inherited from the class level for a normal shutdown. The second command was defined as the fourth pass which results in three *Shutdown Pass Intervals* before the command is issued.

Notice the use of the **&SUBSJOB** variable. There is no full stop (period) at the end of **&SUBSJOB** (unlike other **&-prefix** definitions in the policy). The value of **&SUBSJOB** is then substituted with the name of the job name for this subsystem. Press PF3 to save the SHUTNORM commands and passes.

# Create HasParent relationship

COMMANDS ACTIONS VIEW HELP

---

AOFGXREL Relationship Selection List  
Command ==> new hasparent  
No entries currently exist. Use the NEW command to create an entry.  
Entry Type : Application PolicyDB Name : NEW\_PDB  
Entry Name : AUTOSSI Enterprise Name : SAPLEX

External Startup . . . \_\_\_\_\_ (INITIAL ALWAYS NEVER)  
External Shutdown . . . \_\_\_\_\_ (FINAL ALWAYS NEVER)

Action # Type Supporting Resource Auto Chain  
\*\*\*\*\* Bottom of data \*\*\*\*\*

AOFGXRE0 Define Relationship  
Command ==> \_\_\_\_\_

Entry Type : Application PolicyDB Name : NEW\_PDB  
Entry Name : AUTOSSI Enterprise Name : SAPLEX

Description . . . . JES2 must be up

Relationship Type . . HASPARENT MAKEAVAILABLE MAKEUNAVAILABLE  
PREPAVAILABLE PREPUNAVAILABLE  
HASPARENT HASPASSIVEPARENT  
HASMONITOR PEEROF  
FORCEDOWN EXTERNALLY

Supporting Resource . JES2/APL/= Resource Name

Define HasParent relationship to JES2 to ensure that JES2 is running before the SSI.

**Define HasParent**  
relationship to JES2 to  
ensure that JES2 is  
running before the SSI.

### Create HasParent relationship

To create a simple HasParent dependency for AUTOSSI, select the RELATIONSHIPS policy item. The \*BASE PDB does not contain any relationships for AUTOSSI. To create a new dependency, type **NEW** on the command line and press Enter.

On this panel, you define the type of relationship (HasParent) and the other Application in the relationship (JES2). This example defines that AUTOSSI has a parent of JES2. The resource name in this case is SubsystemName/APL/MVS\_Sysname, where an equal sign (=) can be used instead of the local MVS system name. For example, you can enter **JES2/APL/=**. Press PF3 to create the relationship.

## WHERE USED policy

| COMMANDS   |          | ACTIONS   | VIEW   | HELP |
|--|----------|---|--|------|
| AOFGXWHU   |          | Where Used  | Row 1 to 11 of 11<br>SCROLL==> <a href="#">CSR</a> |      |
| Command ==> <input type="text"/>                       |          |   |  |      |
| Entry Type : Application<br>Entry Name : AUTOSSI       |          | PolicyDB Name : NEW_PDB<br>Enterprise Name : SAPLEX |  |      |
| Action   | Status   | Name  | Type   |      |
|  | SELECTED | AM_X  | APG  |      |
|  |          | BASE_SYS  | APG  |      |
|  |          | BASE_USS  | APG  |      |
|  |          | FTP_DAEMON  | APG  |      |
|  |          | MYSERV1   | APG  |      |
|  |          | MYSERV1_X   | APG  |      |
|  |          | NFS_CLNT  | APG  |      |
|  |          | NFS_SERV_X  | APG  |      |
|  |          | SSH_DAEMON  | APG  |      |
|  |          | TELNET_2270   | APC  |      |
| AOFGXWHU   |          | Where Used  |  |      |
| *****<br>Command ==> <input type="text"/>              |          |   |  |      |
| Entry Type : ApplicationGroup<br>Entry Name : BASE_SYS |          | PolicyDB Name : NEW_PDB<br>Enterprise Name : SAPLEX |  |      |
| Action   | Status   | Name  | Type   |      |
|  | SELECTED | MVSBR   | SYS  |      |
|  | SELECTED | SYS1  | SYS  |      |
|  | SELECTED | SYS3  | SYS  |      |

**BASE\_SYS APG is linked to all systems**

**AUTOSSI is linked to the BASE\_SYS Application Group**

© Copyright IBM Corporation 2019

3-31

### WHERE USED policy

The last step in this example is to verify the WHERE USED policy item. This panel shows that AUTOSSI is linked to the SYSVIEW Application Group. In other words, AUTOSSI are linked to every system that the SYSVIEW APG runs on. To determine which systems are linked, you can view the WHERE USED or GENERATED RESOURCES policy items for the SYSVIEW APG. In this example, the SYSVIEW APG is linked to MVSBR.

# Lesson 4 Other policy customization features

## Lesson 4: Other policy customization features

- Data Management Menu
  - Import policy from an existing PDB
  - Import from predefined add-on policies
  - Read from flat files created from customization dialog
- Query policy statistics
- Copy Application policy to create an Application

### What this lesson is about

This lesson provides a description of additional policy definitions that can be defined in System Automation for z/OS 4.1.

### Objectives

After completing this lesson, you should be able to define and describe additional policy items that in System Automation for z/OS 4.1.

The Data Management Menu provides selections for importing sample policy databases and policy add-ons. You can also select option 3 to use files for updating policy data.

## Query policy statistics

The screenshot shows a terminal window with the following text:

```

COMMANDS ACTIONS VIEW HELP
AOFGENAM          Entry Name Selection      Row 22 to 37 of 37
Command ===>                               Scroll ==> CSR
Entry Type : Application        PolicyDB Name : OPSU1EX1
Action   AOFGPQST      Entry Statistics
        Entry name . . . . . SYSVAPPL
        Last update by . . . INGC102
        Date/Time modified . . 18/08/26 at 20:34:10
        Date/Time build . . . 18/12/16 at 11:57:28
        ACF fragment name . . Z982AAPL
        F1=Help    F2=Split   F3=End     F9=Swap   F12=Cancel
Q           SYSVIPLC      IPL Data Gatherer
           SYSVSSI       Automation Subsystem Interface
           TCPIP         TCP/IP

```

A yellow callout box highlights the entry statistics for 'SYSVAPPL'. The text inside the box is:

**Automation  
Agent  
NetView**

Below the highlighted box, the text continues:

em  
lity  
y  
r  
cility

Use the **Q** action to query statistics for a policy entry:

- Last user to update and when
- Time of last build
- ACF fragment for build

© Copyright IBM Corporation 2019

3-33

### Query policy statistics

You can view statistics about each policy by entering **Q** in the Action field. In this example, Application AUTONETV is part of the ACF fragment Z993AAPL and was last updated by user INGC200. The statistics show you the time stamps for the last update and last build. You can compare the time stamps to determine if the last change was not in the built policy, for example. If there is no time stamp for the build or ACF fragment name, then the selected entry was not built.

## Copy Application policy example

The screenshot shows the AOFGEPOL Policy Selection panel. At the top, it displays 'Entry Type : Application' and 'Entry Name : TEST1'. In the 'Action' column, 'COPY' is selected. A yellow callout bubble points to the 'COPY' option with the text: 'New APL TEST1: Select COPY'. Another callout bubble points to the 'Policy Description' field with the text: 'Select an existing APL to copy policy from and press PF3 to save'. The 'PolicyDB Name' is set to 'NEW\_PDB' and 'Enterprise Name' is 'SAPLEX'. The 'Policy Description' field contains 'Select Entry for Copy'. The bottom section shows a list of applications with their descriptions, starting with 'AM' (Advanced Peer-to-Peer Communication) and 'APPCC' (APPCC Scheduler). The 'WHERE USED' column indicates 'COPY' for the selected entry.

| Policy Selection                            |                                  | Entry created                                |
|---|----------------------------------|--|
| Command ==> <input type="button" value=""/> |                                  | Scroll ==> CSR                               |
| Entry Type : Application                    | PolicyDB Name : NEW_PDB          |  |
| Entry Name : TEST1                          | Enterprise Name : SAPLEX         |  |
| Action                                      | Policy Name                      | Policy Description                           |
|   | DESCRIPTION                      | Enter description                            |
| AOFGXCPY                                    |                                  | Select Entry for Copy                        |
| Command ==> <input type="button" value=""/> |                                  | Row 1 from 58                                |
| Entry Type : Application                    |                                  | SCROLL==> PAGE                               |
| Entry Name : TEST1                          |                                  |  |
| Action                                      | Entry Name                       | Short Description                            |
|   | AM                               | Advanced Peer-to-Peer Communication          |
|   | AM2                              | APPCC Scheduler                              |
|   | APPCC                            | Automation Subsystem Interface               |
|   | ASCH                             | Build SNAP Tables for IPCS                   |
|   | AUTOSSI                          | Class for Automation Manager Definitions     |
|   | BLSJPRMI                         | Class for general APL definitions            |
|   | C_AM                             | Class for common DVIPA definitions           |
|   | C_APPL                           | Data Lookaside Facility                      |
|   | C_DVIPA                          | NetView JES-JobID-Requestor                  |
|   | DLF                              | Dynamic Virtual IP Address                   |
|   | DSIRQJOB                         |  |
|   | DVIPA1                           |  |
| WHERE USED                                  | COPY                             | LIST application groups linked to this entry |
|   | <input type="button" value="S"/> | Copy data from an existing entry             |

Animated

© Copyright IBM Corporation 2019

3-34

### Copy Application policy example

Suppose the definition of a new Application is similar, but not identical, to an existing Application. You can create the new Application by copying the attributes of the existing entry into the new one.

In this example, the Policy Selection panel is for a new Application, TEST1. Select the **COPY** policy item and press Enter to display a list of Applications to copy attributes from. You see a list of Applications of the same type and subtype that are defined in the PDB. Select any appropriate Application to copy its attributes and then press PF3 to copy the policy items to the new Application, TEST1. If you copy an existing Application definition, you copy all its attributes except those that were defined when the new entry was created, such as subsystem name or description.

When you return to the list of Applications, you find that the new Application definition (TEST1) has all the attributes of the original, except that it is not selected as a member in any Application Group. Modify TEST1, as needed, including defining the WHERE USED policy to link the Application to an appropriate APG.

## Data Management: Import from PDB

```
A0FGIMPP          Import entries from a Policy Database
Option ===> _____
Current Policy Database      : NEW_PDB
Enterprise Name             : SAPLEX
1 Import Policy Data
    Source Policy Database. . . . OPSU3EX1           (? or name)
    Entry type . . . . . . . APG                  (? or type)
    Import linked entries . . . YES                (YES or NO)
                                (applies to ENS, GRP, SYS, APG, APL, TRG and DMN only)
2 View import report
```

- Import policy from another PDB
  - Define source PDB name or select from list
  - Define entry type, for example: APL, APG, MTR
  - Import linked entries = YES: All entries linked to the selected one are imported
- Select option 1 to continue
- Entries are displayed in the next panel. APGs in this example

© Copyright IBM Corporation 2019

3-35

### Data Management: Import from PDB

On this panel, you can specify the information that is required to import entry data from another Policy Data Base. You have two options:

- **1 (Import Policy Data):** Use this option to select a source Policy Data Base from where entry data is imported and an entry type for which selected entries are imported.  
Specifying **YES** for the Import linked entries field also imports linked entries such as Service Periods, Triggers and Events, Application classes, and Application Groups.
- **2 (View import report):** Use this option to view the import report available in the target Policy Data Base. This option starts the ISPF VIEW function.

After you press Enter, the entry name selection list is displayed. You can modify the selection list if needed. You must define the source and target PDBs. You might have to add the source PDB to the list. The target PDB can be the same as the source PDB. Both PDBs must be from the same SA z/OS release, at the same PTF level. Create a backup copy of the PDB before importing any changes.

## Import from PDB: Select one or more entries

| Action | Status | Entry Name  | Short Description                    |
|--------|--------|-------------|--------------------------------------|
|        |        | AM_X        | Automation Manager Group for Sysplex |
|        |        | BASE_SYS    | Base z/OS, Network and Automation    |
|        |        | BASE_USS    | Base USS resources                   |
|        |        | FTP_DAEMON  | File Transfer Protocol Daemon        |
|        |        | HZSPROC     | Used to link suspended APL HZSPROC   |
|        |        | LOOKASIDE   | LookAside Components                 |
|        |        | RVBASIC     | RV Apps APG for system A             |
|        |        | TELNET_3270 | TELNET Server for 3270 connections   |

- Select one or more entries to import

- Press Enter to continue

The selected monitor resources are imported to the NEW\_PDB

Use the Entry Name Selection panel to select the entries to import into your current Policy Data Base. In this example, select the monitor resource to import and press Enter. In the Command Progress Display window, you can see status messages related to the import of the monitor resource.

## Data management: Import add-on policy

| ACTIONS   |        | HELP          | Import Add-on Policies |  | Row 1 to 13 of 13 |
|---|--------|---------------|------------------------|--|-------------------|
| AOFGPIMA<br>Option ==>                                      |        |               |                        |  |                   |
| 1 Import selected add-on policies<br>2 View import report   |        |               |                        |  |                   |
| Current Policy Database: NEW_PDB                            |        |               |                        |  |                   |
| Add-on policies to be added to the current policy database: |        |               |                        |  |                   |
| Action  | Status | Add-on Policy | Customizable           |  |                   |
|   |        | *BASE         | YES                    |  |                   |
|   |        | *CICS         | YES                    |  |                   |
|   |        | *DB2          | YES                    |  |                   |
|   |        | *E2E          | YES                    |  |                   |
|   |        | *GDPS         | YES                    |  |                   |
|   |        | *HYPERSWAP    | YES                    |  |                   |
|   |        | *IBMCOMP      | YES                    |  |                   |
|   |        | *IMS          | YES                    |  |                   |
|   |        | *ITM          | YES                    |  |                   |
|   |        | *IWS          | YES                    |  |                   |
|   |        | *PROCOPS      | YES                    |  |                   |
|   |        | *SAPSrv       | YES                    |  |                   |
|   |        | *TBSM         | YES                    |  |                   |
| ***** Bottom of data *****                                  |        |               |                        |  |                   |

Select option 1 with one or more add-on policies

If this option is customizable, you can import a subset of the add-on policies

© Copyright IBM Corporation 2019

3-37

### Data management: Import add-on policy

The policy definitions can be imported using the Data Management panel, option 2. This panel displays the list of add-on policies provided by SA z/OS for each of the products. In this example, the add-on policy for ITM is imported.

You can select from the following possible actions:

- **Select (S):** Import the complete add-on policy.
- **Remove (M):** Clear the add-on policy.
- **Customize (C):** Define a subset of the add-on policy to import.
- **Edit (E):** Use Edit the same way that you use Customize.
- **View (V):** Display documentation about the add-on policy file.

## Example add-on import: \*ITM

```
A0FGIMPC          Select Add-on Policy Components           Row 1 to 10 of 10
Command ==> [ ]  SCROLL==> CSR

Components of Add-on Policy : *ITM

Select one or more components to be added to your Policy Database:

Action Status      Component
_____
_____  
SELECTED   Automation Monitoring Agent (TEMA)
_____  
SELECTED   Monitoring Analytic
_____  
SELECTED   OMEGAMON - NetView z/OS
_____  
SELECTED   OMEGAMON - z/OS
_____  
SELECTED   OMEGAMON - JVM on z/OS
_____  
SELECTED   OMEGAMON - CICS
_____  
SELECTED   OMEGAMON - DB2 Performance Monitor
_____  
SELECTED   OMEGAMON - IMS
_____  
SELECTED   OMEGAMON - Mainframe Networks on z/OS
_____  
SELECTED   OMEGAMON - MQ
***** Bottom of data *****
```

- The \*ITM add-on is customized to import all but the SA z/OS monitoring agent
- Remove to deselect components
- Press END to continue
- Enter 1 to start reading entries (no writing yet)

© Copyright IBM Corporation 2019

3-38

### Example add-on import: \*ITM

This panel displays the policies that will be imported for the \*ITM add-on.



When the \*BASE policy was imported, SYS2 was renamed to MVSB (earlier in this unit). Rename SYS2 to MVSB here (with an Action of R or by over typing its name in the entry field) to automatically generate the appropriate links to the systems when the add-on is imported. If you do not rename the system entries here, new entries for SYS1 and SYS2 are created. The ITM policies are linked to them.

You rename the system entries by typing the new name in the Entry Name field. For example, press the Tab key to SYS2 and overtype MVSB.

## Remove (M) or rename (R) (duplicate) entries

| A0FGIMP3      |              | Entries of selected Add-on Policies |   |              |   | Row 1 to 22 of 47 |  |
|---------------|--------------|-------------------------------------|---|--------------|---|-------------------|--|
| Option ==>    |              |                                     |   |              |   | SCROLL==> CSR     |  |
| 1 Modify data |              | 2 Check data                        |   | 3 Run import |   | 4 View report     |  |
| Action        | Entry Name   | Type                                | C | D            | Short Description                       |                   |  |
| m             | SYSPLEX1     | GRP                                 | Y |              | Placeholder. Original in *BASE          |                   |  |
| m             | SYS1         | SYS                                 | Y |              | Placeholder. Original defined in *BASE  |                   |  |
|               | SYS2         | SYS                                 |   |              | Placeholder. Original defined in *BASE  |                   |  |
| m             | SYS3         | SYS                                 | Y |              | Placeholder. Original defined in *BASE  |                   |  |
|               | BASE_ITM     | APG                                 |   |              | ITM Base Components                     |                   |  |
|               | ING_ANALYTIC | APG                                 |   |              | Automation Monitoring Analytics         |                   |  |
|               | OM_CICS      | APG                                 |   |              | OMEGAMON - CICS                         |                   |  |
|               | OM_DB2       | APG                                 |   |              | OMEGAMON - DB2 Performance Monitor      |                   |  |
|               | OM_IMS       | APG                                 |   |              | OMEGAMON - IMS                          |                   |  |
|               | OM_JVM       | APG                                 |   |              | OMEGAMON - JVM on z/OS                  |                   |  |
|               | OM_MN        | APG                                 |   |              | OMEGAMON - Mainframe Networks z/OS      |                   |  |
|               | OM_MQ        | APG                                 |   |              | OMEGAMON - MQ                           |                   |  |
|               | OM_NV        | APG                                 |   |              | OMEGAMON - Netview z/OS                 |                   |  |
|               | OM_TOM_X     | APG                                 |   |              | OMEGAMON - TOM                          |                   |  |
|               | OM_ZOS       | APG                                 |   |              | OMEGAMON - z/OS                         |                   |  |
|               | C_ITM        | APL                                 | * |              | ITM Application Class                   |                   |  |
|               | C_LOOPSUPP   | APL                                 | * |              | Looping Address Space Suppression Class |                   |  |

- Remove (M) or rename (R) SYStem or other duplicate entries

**Note:** Because SYS2 was renamed to MVSB, it is no longer a duplicate entry  
SYS2 could be also removed

- Press Enter to continue

### Remove (M) or rename (R) (duplicate) entries

This panel displays the policies that will be imported for the \*ITM add-on.



When the \*BASE policy was imported, SYS2 was renamed to MVSB (earlier in this unit). Rename SYS2 to MVSB here (with an Action of R or by over typing its name in the entry field) to automatically generate the appropriate links to the systems when the add-on is imported. If you do not rename the system entries here, new entries for SYS1 and SYS2 are created. The ITM policies are linked to them.

You rename the system entries by typing the new name in the Entry Name field. For example, press the Tab key to SYS2 and overtype MVSB.

# Lesson 5 Policy Data Base reports

## Lesson 5: Policy Data Base reports

- PDB report
  - Detailed description of all objects in PDB
  - Shows all objects in the PDB not linked to a system or sysplex
  - Includes all nonMVS systems
  - Includes all relationships that are not used during build
- Unlinked report
  - Report of all ignored entries as they are not linked to a system or sysplex
  - Relationships that are not used during a build
- Resources report
  - Shows all generated resources for each system
- Unused Members report
  - Lists members not referenced in AOFACFMP
- Status Messages report
  - Lists messages used for status changes
- Symbols report
  - Creates a report of all symbols (AOCCCLONEx)

### What this lesson is about

This lesson describes the policy data base reports available in System Automation for z/OS 4.1.

### Objectives

After completing this lesson, you should be able to describe and produce the policy data base reports available in System Automation for z/OS 4.1.

Use option 3 from the primary menu panel. Run the Unlinked and Resources report before the build to look for any possible errors.

# Report Selection Menu

| Report Selection Menu |   |
|-----------------------|---|
| AOFGREPK              | Option ==>  |
| 1 Report              | Create Policy Database Report                           |
| 2 Unlinked            | Create List of entries not linked in Policy Database    |
| 3 ViewUnlinked        | View List of unlinked entries                           |
| 4 Resources           | Create List of generated Resources                      |
| 5 ViewResources       | View List of generated Resources                        |
| 6 MemberList          | Create List of unused members in configuration data set |
| 7 ViewMemberList      | View List of unused members                             |
| 8 StatusMsgs          | Create Status Messages Report                           |
| 9 ViewStatusMsgs      | View Status Messages Report                             |
| A Symbols             | Create List of Symbols (AOCCLONE) in Policy Database    |
| B ViewSymbols         | View List of Symbols                                    |

- Can only run after a PDB build: Option 6
- View function for unlinked (3), resources (5), unused members (7), status messages (9), and symbols (B) reports

## Report Selection Menu

This slide displays the following options for creating and viewing reports:

- Option 1 (Report) creates a report of all automation data in the Policy Data Base. The report offers two options: to report either the full automation policy or to select a specific entry type and name to be reported. With this function, you can have different output formats.
- Option 2 (Unlinked) creates a report of all entries within the current PDB that are ignored by the build function, which includes the following items:
  - All objects in the Policy Data Base that are not linked to a system or sysplex.
  - All systems that are not MVS.
  - All relationships that are not used during build.
- Option 4 (Resources) creates a report of all automatically generated resources for the automation manager including Monitor resources, Application Groups and Application. It is also shown how resources are linked.
- Option 6 (MemberList) creates a report of all members in the build output data set that are not referenced by the actual configuration or anchored in the ACF map member AOFACFMP. It helps to identify members that can be deleted when performing a data set cleanup. The report is divided into two sections:

- The first section lists all unreferenced members that are identified as valid configuration files and some additional information such as entry name or build time.
- The second section lists all other unreferenced members, for which either no build statistics are found or which are not identified as valid configuration files.

All report files are written to the report output data set. If one does not exist, it is created with a default name and size. Report members created by the customization dialog itself, such as build report member \$BLDRPT, are not included in the report.

- Option **8** (StatusMsgs) creates a report of all messages that are defined to indicate status changes for an Application.
- Option **A** (Symbols) creates a report with information about all the system automation symbols (AOCCCLONEx) in the policy database.

## PDB report files

- Based on report selected, you might find these messages or reports:
  - \$RPTFLAT: Policy Data Base report
  - \$RPTHTML: Policy Data Base report in HTML format
  - RPTNBLD: Unlinked members report
  - RPTRES: Generated resources report
  - RPTBMBR: Members not referenced
  - RPTSMMSG: Status messages; for example, Up or Down
  - RPTSYM: Symbols report
- By default, the report data set name is the same as the PDB data set name with REP concatenated; for example, SAZOS.PDB.ADMIN.REP
- Report data set can also contain other members, such as flat file update reports

© Copyright IBM Corporation 2019

3-42

*PDB report files*

If report format is FLAT, the default name is \$RPTFLAT. If report format is HTML, the default name is \$RPTHTML. You can also specify to split the HTML report into multiple files.

## Reports: FLAT versus HTML

- Hypertext Markup Language (HTML) reports
  - Can be split into multiple files
  - Contain hyperlinks per resource, such as Entry name or Resource name
- FLAT and HTML reports
  - Write to data set or USS file system (HFS or ZFS)  
Directory is created if it does not exist
  - Member name is customizable (default is \$RPTFLAT or \$RPTHTML)
  - INGEBRPT JCL in SINGSAM for batch invocation

### Reports: FLAT versus HTML

HTML reports can be split into multiple files. Using the total report size and the number of files specified, an average size is calculated so that all files are approximately the same size.

## Policy Data Base Report Parameters panel

- Select format: FLAT or HTML
  - Define member name: \$RPTFLAT or \$RPTHTML
  - Select HTML file split, if format HTML. (Note: \* = 10,000 lines per page)
  - Select option 1 to create report for an enterprise

© Copyright IBM Corporation 2019

3-44

## *Policy Data Base Report Parameters panel*

The Policy Data Base report does not provide a viewing option. Instead, it creates a PDS member flat file or Hypertext Markup Language (HTML) file that can be viewed using ISPF or a browser in the specified data set. Both files are stored in the report data set, as shown on the slide.

This selection creates a report of all automation data in the Policy Data Base. The report offers two options:

- Generate the report for the full automation policy (option 1).
  - Select a specific entry type and name to be reported (option 2).

# HTML report example

## System Automation for z/OS Policy database report

|                           |                    |
|---------------------------|--------------------|
| <b>UserId:</b>            | INGC102            |
| <b>PolicyDB Name:</b>     | IBM_ALL            |
| <b>PolicyDB data set:</b> | 'SAZOS.PDB.IBMALL' |
| <b>Creation Date:</b>     | Monday, 4 Mar 2019 |
| <b>Creation Time:</b>     | 15:19:47           |

### Selected Entry Type(s)

[ENT: Enterprise, AT, MRT and MPF Specifications](#)  
[GRP: Groups](#)  
[SBG: SubGroups](#)  
[SYS: Systems](#)  
[APG: ApplicationGroups](#)  
[APL: Applications](#)  
[EVT: Events](#)  
[SVP: Service Periods](#)  
[TRG: Triggers](#)  
[PRO: Processors](#)  
[MTR: Monitor Resources](#)  
[ENS: zEnterprise Ensembles](#)  
[TMR: Timers](#)  
[TPA: Tape Attendance](#)  
[MVC: MVS Components](#)  
[PAC: Pacing Gates](#)  
[MDF: MVSCOMP Defaults](#)  
[SDF: System Defaults](#)  
[ADF: Application Defaults](#)  
[AOP: Automation Operators](#)  
[NFI: Notify Operators](#)  
[NTW: Networks](#)  
[XDF: Sysplex Defaults](#)  
[RES: Resident CLISTS](#)  
[SCR: Status Display](#)  
[UET: User E-T Pairs](#)  
[DMN: Remote Domains](#)  
[REF: Resource References](#)  
[OPEN: OPC System Details](#)  
[OCS: Controller Details](#)  
[QSR: Special Resources](#)  
[ODM: Workstation DomainIDs](#)

© Copyright IBM Corporation 2019

3-45

### HTML report example

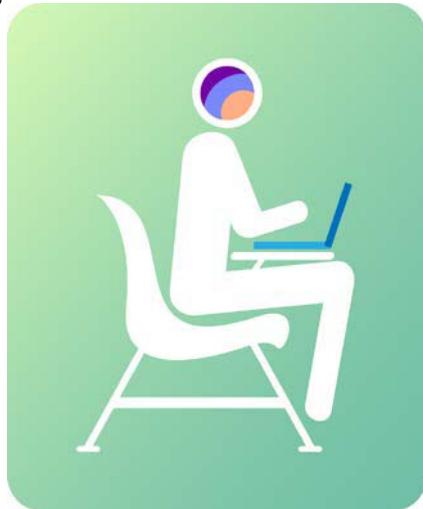
On this slide, you see an example of the HTML report viewed in a browser.

## Student exercises

### Unit 3

---

1. Start automation on MVS and test the installation
2. Navigate the ISPF customization dialogs and populate the policy database



---

© Copyright IBM Corporation 2019

3-46

*Student exercises*

Open your *Student Exercises* book and perform the exercises for this unit.

## Summary

Now that you have completed this unit, you should be able to perform the following tasks:

- Navigate the ISPF customization dialog
- Create and manage new Policy Data Bases
- Create and link policy objects in the Policy Data Base
- Copy, import, and rename Policy Data Base entries
- Generate Policy Data Base reports

### *Summary*

Now that you completed this unit, you can perform the following tasks:

- Navigate the ISPF customization dialog
- Create and manage new Policy Data Bases
- Create and link policy objects in the Policy Data Base
- Copy, import, and rename Policy Data Base entries
- Generate Policy Data Base reports





## 4 Defining applications and building automation configuration files



### Unit 4 Defining applications and building automation configuration files



© Copyright IBM Corporation 2019

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

**What this unit is about**

This unit focuses on how to define an Application by using the SA z/OS Customization dialog and how to build the configuration files that are used by the automation manager and automation agent. You also learn about the format of the configuration files and how to update the Policy Data Base (PDB) from flat files.

**How you check your progress**

You can check your progress in the lab exercises.

**References**

*SC34-2717 System Automation for z/OS Version 4*

*Release 1 Defining Automation Policy*

[https://www.ibm.com/support/knowledgecenter/SSWRCJ\\_4.1.0/com.ibm.safos.doc\\_4.1/manuals.html](https://www.ibm.com/support/knowledgecenter/SSWRCJ_4.1.0/com.ibm.safos.doc_4.1/manuals.html)

# Objectives

---

After completing this unit, you should be able to perform the following tasks:

- Define an application in the Policy Data Base
- Describe the automation configuration file format and how it is used
- Build the automation configuration file
- Perform flat file updates to the Policy Data Base

## *Objectives*

After completing this unit, you should be able to perform the following tasks:

- Define an application in the Policy Data Base
- Describe the automation configuration file format and how it is used
- Build the automation configuration file
- Perform flat file updates to the Policy Data Base

# Lesson 1. Defining Application policy

## Lesson 1: Defining application policy

- When you create an application, you use the Define New Entry panel to define types, such as Object type, application type, subtype, or job type
- After you create the application, you can edit the following policies:
  - APPLICATION INFO
    - Transient jobs, nonMVS jobs, and other definitions
    - Desired Available and Inform List
    - Recycle options, restart options
  - RELATIONSHIPS
  - MESSAGES/USER DATA
  - AUTOMATION FLAGS
  - STARTUP
  - SHUTDOWN
  - THRESHOLDS

### What this lesson is about

This lesson describes how to define application policy using the System Automation for z/OS ISPF customization dialog.

### Objectives

After completing this lesson, you should be able to define application policy using the System Automation for z/OS ISPF customization dialog.

### References

*SC34-2717 System Automation for z/OS Version 4  
Release 1 Defining Automation Policy*

When you define an Application through the dialogs, there are many possible customization options. In this lesson, you learn about each of these items.

# New application creation

| Define New Entry                            |   |
|---|---|
| <b>AOFGDYN3</b>                             | Command ==> _____   |
| <b>Define new entry of type Application</b> |   |
| Entry name . . . . .                        | <u>AUTOSSI</u> _____  |
| Subsystem Name . . . . .                    | <u>AUTOSSI</u> _____  |
| Object Type . . . . .                       | <u>INSTANCE</u> _____   |
| Category . . . . .                          | _____ (CLASS INSTANCE)<br>(IBM-defined, user-defined or blank,<br>see help) |
| Subcategory . . . . .                       | _____ (IBM-defined, user-defined or blank,<br>see help)                     |
| Job Type . . . . .                          | <u>MVS NONMVS TRANSIENT</u> _____   |
| Job Name . . . . .                          | <u>AUTOSSI</u> _____  |
| Transient Rerun . . . . .                   | _____ (YES NO)  |
| Scheduling Subsystem . . . . .              | _____ (MSTR, JES Subsystem)   |
| JCL Procedure Name . . . . .                | _____   |
| Short Description . . . .                   | <u>NetView SSI</u> _____  |
| Long Description 1 . . . . .                | _____   |
| Long Description 2 . . . . .                | _____   |
| Long Description 3 . . . . .                | _____   |
| Long Description 4 . . . . .                | _____   |
| Long Description 5 . . . . .                | _____   |

Define a new MVS application, AUTOSSI

## New Application creation

Application policy is found under option **6 (APL)** of the customization dialog. Type **NEW** on the command line and press Enter to define a new Application.

On the Define New Entry panel, you can enter basic information about a new Application. Initially, all fields except Object Type and Application Type are null.

One of the most important decisions you make about an Application definition is whether or not it will function as a class or an instance. A class is like a template that is used to store attributes that are common to Application instances of that class. The instance is an actual occurrence of the Application. On the slide is a simple example of defining the NetView SSI as an Application.

*Object Type* can be CLASS or INSTANCE:

- **CLASS:** Use this type if the entry represents common policy to be inherited by one or more other Application entries defined as INSTANCEs. Classes are used only to define inherited policy; they do not represent Applications that are managed by SA z/OS.
- **INSTANCE:** Use this type if the entry represents an Application to be managed by SA z/OS. You can link an Application that is defined as an INSTANCE to an Application that is defined as a CLASS. When you link these Applications, you can omit policy items from the

INSTANCE where the policy is inherited from the specified CLASS. Application classes are beneficial. Use them whenever possible.

Category identifies one of the following types:

- CICS
- DB2
- E2EAGT
- IMAGE (if the Application represents a system image)
- IMS
- INFOSPHERE
- ITM
- JES2
- JES3
- LIFELINE
- MQ
- OPC (if the Application represents an IWS Application)
- USS
- TCPIP

When you select an Application Category specification, you have additional policy options, that are specific to the automation of those Applications. These options are displayed on the *Policy Selection* panel. The Application Type value cannot be changed.

Subcategory specifies the region subtype for CICS, DB2, and IMS, or the controller type for TWS. You can specify the following subtype values:

- **CICS:** AOR TOR FOR DOR QOR CMAS WUI
- **IMS:** CTL TP DBRC DLS FP BMP FDR
- **ITM:** KAHAGENT TEMS
- **DB2:** MSTR SPAS IRLM DBM1 DIST WLMS ADMT
- **TWS:** CONTROLLER, TRACKER, SERVER, or DATASERVER
- ADVISOR AGENT for Category **INFOSPHERE**

For type INFOSPHERE or when type is blank, the subtype can be any value containing alphabetic and numeric characters. For other JES2, JES3, USS, IMAGE Application Types, the value is blank.

*Job Type* indicates these job properties of the Application:

- **MVS:** Applications that are usually long-running (for example, TSO) and monitored using Monitor Routine INGPJMON which monitors status via ASCB checking (not the MVS DISPLAY ACTIVE command).
- **NONMVS:** Applications that cannot be monitored using the Monitor Routine INGPJMON.
- **TRANSIENT:** Applications that are monitored with the Monitor Routine INGPJMON and are usually short-running and self-terminating.

If the Application Type is IMAGE, the job type is forced to NONMVS. For other Applications, the default value is MVS. NONMVS Applications must have STARTUP commands specified, or errors occur.

SA z/OS accepts status change messages for MVS Application resources, as in the following example:

- IEF403I job started
- IEF404I job ended
- IEF450I job abended

SA z/OS ignores these messages for non-MVS resources or the Application is not a process.

# APPLICATION INFO policy

| COMMANDS HELP              |                  | Application Information                           | Line 00000001<br>Scroll ==> CSR |
|----------------------------|------------------|---|---------------------------------|
| Command ==>                |                  |   |                                 |
| Entry Type : Application   |                  | PolicyDB Name : NEW_PDB                           |                                 |
| Entry Name : AUTOSSI       |                  | Enterprise Name : SAPLEX                          |                                 |
| Category . . . . .         |                  | (IBM-defined, user-defined or blank,<br>see help) |                                 |
| Subcategory . . . . .      |                  | (IBM-defined, user-defined or blank,<br>see help) |                                 |
| Subsystem Name . . . . .   | <u>AUTOSSI</u>   |   |                                 |
| Job Type . . . . .         | <u>MSTR</u>      | (MVS NONMVS TRANSIENT)                            |                                 |
| Job Name . . . . .         | <u>AUTOSSI</u>   |   |                                 |
| Transient Rerun . . . . .  | <u> </u>         | (YES NO)  |                                 |
| Scheduling Subsystem . . . | <u>MSTR</u>      | (MSTR, JES Subsystem)                             |                                 |
| JCL Procedure Name . . .   | <u>CNMSJ010</u>  |   |                                 |
| Job Log Monitor Interval . | <u> </u>         | (mm:ss NONE)                                      |                                 |
| Captured Messages Limit .  | <u> </u>         | (0 to 999)  |                                 |
| Desired Available . . . .  | <u> </u>         | (ALWAYS ONDEMAND ASIS)                            |                                 |
| Restart after IPL . . . .  | <u> </u>         | (START NOSTART NONE)                              |                                 |
| Monitor for IPL complete . | <u> </u>         | (YES NO)  |                                 |
| Start Delay . . . . .      | <u> </u>         | (time for "UP" status checks, hh:mm:ss)           |                                 |
| Start Cycles . . . . .     | <u> </u>         | (start delay checks, 0 to 99)                     |                                 |
| UP Status Delay . . . . .  | <u> </u>         | (time to delay "UP" status, hh:mm:ss)             |                                 |
| Restart option . . . . .   | <u>ABENDONLY</u> | (ALWAYS ABENDONLY NEVER)                          |                                 |
| External Startup . . . . . | <u> </u>         | (INITIAL ALWAYS NEVER)                            |                                 |

More definitions are available when you press PF8

© Copyright IBM Corporation 2019

4-5

## APPLICATION INFO policy

The Application Information panel has several of the same fields as the Define New Entry panel. You can add new values for the Application or change existing values for the Application. Most of the Applications that you define have a Job Type of MVS. However, you might have to define TRANSIENT or NONMVS Job Types as well. This lesson is about several definitions, such as Desired Available, Restart after IPL, and Inform List.

## APPLICATION INFO policy (continued)

| COMMANDS                         | HELP                                   |
|----------------------------------|--|
| <b>Application Information</b>   |  |
| <b>Command ==&gt;</b>            | Line 000000023<br>Scroll ==> CSR       |
| Entry Type : Application         | PolicyDB Name : NEW_PDB                |
| Entry Name : AUTOSSI             | Enterprise Name : SAPLEX               |
| Skip ACTIVE status . . . . .     | (YES NO)                               |
| Startup Parameters . . . . .     |  |
| External Shutdown . . . . .      | (FINAL ALWAYS NEVER)                   |
| Shutdown Pass Interval . . . . . | (hh:mm:ss)                             |
| Cleanup Delay . . . . .          | (hh:mm:ss)                             |
| Command Prefix . . . . .         |  |
| Message Prefix . . . . .         |  |
| Sysname . . . . .                |  |
| Monitor Routine . . . . .        | (name NONE)                            |
| Monitor Interval . . . . .       | (hh:mm NONE)                           |
| Inform List . . . . .            | (SDF EIF E2E IOM ITM SMF TTT USR NONE) |
| ARM Element Name . . . . .       |  |
| WLM Resource Name 1 . . . . .    |  |
| WLM Resource Name 2 . . . . .    |  |
| WLM Resource Name 3 . . . . .    |  |
| Owner . . . . .                  |  |
| Info Link . . . . .              |  |
| Runtokens . . . . .              |  |

More definitions are available when you press PF7

© Copyright IBM Corporation 2019

4-6

### APPLICATION INFO policy (cont.)

The following fields are on this panel: You might have to press PF8 to see all fields.

- **Skip ACTIVE Status:** If YES, sets the status to UP if triggered by ACTIVMSG command.
- **Startup Parameters:** Specifies the desired subsystem startup parameters. These values are added to the MVS START command for the subsystem.
- **External Shutdown:** Specifies whether the application is stopped externally or via a specified shutdown procedure.
- **Shutdown Pass Interval:** Enter a reasonable value for the time needed to shut down the subsystem. If it does not shut down in this time, further shutdown commands are entered, and it waits for this interval again.
- **Cleanup Delay:** Use this value to specify an interval that begins when the final termination message (FTERMMSG) for the subsystem is processed. With this delay, additional processing

is performed by or on behalf of the subsystem to the message being issued. If no value is specified, the automation agent default value is used.

- **Monitor Routine:** The name of the application monitor routine that is used to monitor the application's status. INGPJMON for Type MVS.
- **Monitor Interval:** This is the amount of time between application subsystem monitoring cycles.
- **Inform List:** Where all status changes are propagated to and whether SMF records are written. Typically you specify SDF at the class level or at the systems default definition.
- **WLM Resource Name 1–3:** the WLM resource status is set to ON when the application is in the UP or ENDED automation status.
- **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.

## APPLICATION INFO: Desired Available

- You can use the Desired Available policy to define a default goal
  - ALWAYS: Desired status of the resource is set to Available
  - ONDEMAND: Desired status of the resource is set to Unavailable. The resource can be started
    - Manually by an operator
    - Based on a vote from a Service Period
    - Based on a propagated vote from a parent resource
  - ASIS: Desired status of the resource is set to the observed status. No action is taken unless requested
- The default value is ALWAYS; to be consistent with prior releases of SA z/OS

© Copyright IBM Corporation 2019

4-7

### APPLICATION INFO: 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 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.

#### **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.

If nothing is specified or inherited the default value is ALWAYS.

## APPLICATION INFO: Restart after IPL

- Restart after IPL specifies how SA z/OS determines the initial status of a resource when SA z/OS is started after a system IPL
  - START: Force soft down status to allow start of resource
  - NOSTART: Never start; injects stop request with FORCE priority
  - Null: Application can be started, depending on its status
  - NONE: Similar to null but does not inherit the value from a class
- INITSTART automation flag applies

### APPLICATION INFO: Restart after IPL

*Restart after IPL* refers to the action that is taken the first time the agent initializes after an IPL.

The following supported values are included:

- **START:** Indicates that the Application is started, regardless of the Application status. Only consider START when you are sure that the Application can be restarted satisfactorily after any failure, or where the effect of not starting it is too great. Typically, only Applications like JES, VTAM, LLA, and TSO are in this category.

START does not override any persistent INGREQ STOP requests on a re-IPL. For example, suppose that an operator used INGREQ STOP for an Application, intending to manually restart at some future time. The operator STOP request overrides the restart after IPL specification.

Starting systems is subject to the setting of automation flags. Specifying START does not override any flag settings (agent or manager).

- **NOSTART:** Injects a STOP request with priority FORCE during SA z/OS initialization. The result is a Desired Status of UNAVAILABLE and an Observed Status of SOFTDOWN, which results in a Compound Status of SATISFACTORY.

If the Observed Status was HARDDOWN before the IPL, the resource remains in HARDDOWN state. This yields a Compound Status of PROBLEM.

- **Null:** Indicates that the Application is startable, depending on its status. If the status is STOPPING, STOPPED, BREAKING, BROKEN, or CTLDOWN, it is not changed and the Application is not started. Any other status is changed to DOWN during initialization and, if the INITSTART flag is set to YES, the Application is started when all its relationships are fulfilled.
- **NONE:** The same as blank except that this Application, if it is an INSTANCE, does not inherit the value from its related CLASS Application.

If **null** or **NONE** is specified, you see a MAYBE setting when using DISPSTAT. In this case, the Automation Status File (ASF) is consulted to see the last status of the Application. If it was BROKEN, CTLDOWN, or STOPPED, then no start is attempted, because these are considered to be statuses that require operator action.

You can also specify a MESSAGES/USER DATA policy for the special message REFRESHSTART or ANYSTART; specifying commands that can be issued when the agent restarts or recycles. For example, you can issue query commands to determine internal Application status that might have been lost while automation was not active.

Related to these options is the **Start Cycles** field. Start Cycles specifies the number of times to cycle the start timeout period before posting the subsystem as a problem. The Application is given a STARTED2 or INACTIVE status. If no value is specified, a value of one (1) is assumed.

## APPLICATION INFO: Restart option

- The Restart option controls the restart of an Application
  - NEVER: Do not restart the Application; always sets status to STOPPED or BROKEN
  - ABENDONLY: Restart Application when abend is detected
    - Interim Agent Status was ABENDING
  - ALWAYS:
    - Application encounters restartable failure
    - Operator stops Application or voluntary termination (previous Agent Status was STOPPING or UP)
    - Not restarted if interim Agent Status is BREAKING or if critical threshold is met
- Resource thresholds apply
- RESTART automation flag applies

### APPLICATION INFO: Restart option

The *Restart Option* specification defines the circumstances under which the Application is restarted.

- **NEVER:** Application is never restarted.

A specification of NEVER is obvious. You can use NEVER if you know that all failures need attention. The subsystem status is BROKEN. The Agent Status of NEVER is specified.

- **ABENDONLY:** Application is restarted only when an abend message is detected for the Application.

A recoverable ABEND is any abnormal termination that is not signaled as a BREAKING or STOPPING condition. By default, this means any ABEND condition. If you specify ABENDONLY, consider also identifying terminations that are BREAKING conditions; that is, cases in which a restart is always impossible.

A special specification in the System Defaults policy object typically sets system 122 and 222 abends to an Agent Status of STOPPING. This specification ensures that the end status becomes STOPPED and that the subsystem is not restarted.

- **ALWAYS:** Application is restarted if it was terminated without using the INGREQ command.

A specification of ALWAYS causes a subsystem to start immediately after any recoverable failure or operator stop (not an irrecoverable failure). The subsystem starts again as soon as it goes down, even if an operator stops it with an MVS command. An unsuspecting operator might be confused by these results. Avoid this option unless you are sure it is what you want.

Consider using the goal-oriented automation, RESTART=ALWAYS. Ensure that you analyze and restart each Application according to its requirements. When you restart systems, you must set the typical automation flags. Specifying ALWAYS or ABENDONLY does not override any flag settings. Applications that restart because of an ABENDONLY specification are also subject to threshold checking. Resource thresholds apply. See the THRESHOLDS policy item for the Application.

# APPLICATION INFO: Inform List

- The Inform List policy identifies where status notifications are propagated to
  - SDF: SA z/OS Status Display Facility
  - IOM: System Automation for Integrated Operations Management
    - For email and pager support, for example
  - SMF: System management facility
  - EIF: Event integration facility: Tivoli Netcool OMNIbus
  - E2E: application will be shown on the e2e domain view
  - ITM: Tivoli Enterprise Portal “pure event situations” for captured messages
  - TTT: Trouble ticket application (XML data format, uses Tivoli Directory Integrator)
    - Tivoli Service Request Manager
    - Peregrine Service Center
  - USR: For user-written notification handlers.
  - NONE: No status notifications are sent (default).
- The Inform List can be inherited from Application class or system defaults

© Copyright IBM Corporation 2019

4-10

## APPLICATION INFO: Inform List

Inform List controls where SA z/OS status updates are propagated to: SDF, EIF, E2E, IOM, ITM, SMF, TTT and USR. If no Inform List policy is specified, you can inherit the Inform List policy from an Application Class, MVS\_COMP, system defaults, and sysplex defaults. The default value is for no status updates. Status updates are sent based on predefined SA z/OS alert points, using the INGALERT command. SA z/OS automatically generates clearing events for notification types of EIF and USR to remove the event from the OMNIbus, for example.

### Inform SDF

The Status Display Facility (SDF) is informed about any status change of automation resources. SDF also receives any exceptional message that has been captured either by means of the policy or directly (implicitly by SA z/OS or explicitly by user-defined Automation Table entries or scripts).

### Inform ITM

When you use the Tivoli Enterprise Portal to monitor the operational environment, you can use so-called “pure event situations” to monitor certain messages of interest. When ITM is specified in

the Inform List field, SA z/OS uses its Monitoring Agent to create a pure event for each message captured either by means of the policy or directly

### **Inform IOM, EIF, USR, TTT**

The targets IOM, EIF, USR and TTT all have in common that they are intended for alert-based notification. Alerts are generated at an alert point, which can be pre-defined by SA z/OS, user-defined by means of using the INGALERT command or when exceptional messages are captured with a severity of CRITICAL

## APPLICATION INFO: Additional options

- Skip ACTIVE Status: When set to YES, the UP status is based on standard ACTIVEMSG message (IEF403I) instead of UP message
- WLM resource names: You can specify up to three names
- Subsystem Name: The default setting is the entry name, if the entry name is valid as a subsystem name (during new processing). You can modify the subsystem name later. Improves identification of resources
- Job Name: The default name is the subsystem name if the subsystem name is valid as a job name
- Captured Messages Limit: The limit is the number of messages to save
- Owner: This specification can be any text up to 50 characters. You can use this option to add information for the operator, such as a contact name and phone number
- Runtokens: Specify one or more runtokens to qualify the applications for one or more runmodes

### APPLICATION INFO: Additional options

*Work Load Manager (WLM)* resource names are the WLM resources associated with the Application. When the Application is in either the UP or the ENDED Agent Status, the resource is set to ON. When the Application is in any other status, the resource is set to OFF.

*Subsystem Name* gives a unique, 11-character name to the Application you define. This name is used by SA z/OS automation. You can have multiple Applications with the same subsystem name, but only one of them can be linked to any one system. If no subsystem name is specified, the default is the value of the entry name if this name meets the following conditions:

- Has a maximum of eight characters for Applications of type IMAGE
- Has a maximum of eleven characters for all other Application types

*Job Name* is the job name that the instances of this Application run with. The job names must be unique within each system and correspond to the started task or batch job name. You can include system symbols and system automation symbols. The *Owner* specification can be seen using the INGINFO, DISPINFO, and DISPMTR commands. The *Captured Messages Limit* specifies the maximum number of messages that are saved for display using the DISPINFO command.

Runtokens are text strings that are used to qualify the applications for one or more runmodes.

## APPLICATION INFO: Transient resources

| COMMANDS HELP                      |                       | Application Information                        |                              | Line 00000001  |
|------------------------------------|-----------------------|--|------------------------------|----------------|
| Command ==> <input type="text"/>   |                       | PolicyDB Name : NEW_PDB                        | Enterprise Name : SAPLEX     | Scroll ==> CSR |
| Entry Type : Application           | Category . . . . .    | (IBM-defined, user defined or blank, see help) |                              |                |
| Entry Name : BLSJPRMI              | Subcategory . . . . . | (IBM-defined, user defined or blank, see help) | <b>BLSJPRMI is transient</b> |                |
| Subsystem Name . . . . .           | <u>BLSJPRMI</u>       | (MVS NONMVS TRANSIENT)                         |                              |                |
| Job Type . . . . .                 | <u>TRANSIENT</u>      |  |                              |                |
| Job Name . . . . .                 | <u>BLSJPRMI</u>       |  |                              |                |
| Transient Rerun . . . . .          | <u>YES</u>            | (YES NO)                                       |                              |                |
| Scheduling Subsystem . . .         | <u>MSTR</u>           | (MSTR, JES Subsystem)                          |                              |                |
| JCL Procedure Name . . .           |                       |  |                              |                |
| Job Log Monitor Interval : _____   |                       | (mm:ss NONE)                                   |                              |                |
| Captured Messages Limit : _____    |                       | (0 to 999)                                     |                              |                |
| Desired Available . . . . .        |                       | (ALWAYS ONDEMAND ASIS)                         |                              |                |
| Restart after IPL . . . . .        |                       | (START NOSTART NONE)                           |                              |                |
| Monitor for IPL complete . . . . . |                       | (YES NO)                                       |                              |                |
| Start Delay . . . . .              |                       | (time for "UP" status checks, hh:mm:ss)        |                              |                |
| Start Cycles . . . . .             |                       | (start delay checks, 0 to 99)                  |                              |                |
| UP Status Delay . . . . .          |                       | (time to delay "UP" status, hh:mm:ss)          |                              |                |
| Restart option . . . . .           |                       | (ALWAYS ABENDONLY NEVER)                       |                              |                |
| External Startup . . . . .         |                       | (INITIAL ALWAYS NEVER)                         |                              |                |

- Transient resources must run and complete before dependents can start
- You can define transients with Rerun = Yes or No
- Transients can be parents
- Examples include CAS9, EZAZSSI, IXFP, ABENDAID, and BLSJPRMI

© Copyright IBM Corporation 2019

4-12

### APPLICATION INFO: Transient resources

*Transient resources* are resources that must run as part of some initialization. Their completion is typically the signal for dependent jobs to be started. Transient resources do not stay active, but complete normally. The dependent resources can only be started when the transient resource completes.

A transient job is handled differently by the automation agent. When its UP message is received, the job is set to an Agent Status of RUNNING instead of UP. When it ends, it is set to an Agent Status of ENDED instead of STOPPED. (In many cases, the UP message is not defined.)

If the status for a transient resource is RUNNING, and there is a WhenAvailable condition, the dependents cannot start. However, it might start if there is a WhenRunning or similar condition. Some transient jobs must only run once within an IPL lifetime. Others can be run as often as required. The *Transient Rerun* field specifies whether or not a TRANSIENT Application can be rerun and can only be specified for a TRANSIENT Application. If the field is blank, a default value of NO is used. A value of START for the *Restart after IPL* field specifies that the transient job runs once at IPL time. This sets the status to DOWN. It is not a guarantee that the job will run.

## APPLICATION INFO: NonMVS job type

- Can be used to define automation for processes, transactions, performance levels, and IBM Workload Scheduler automation
- Define job type field as NONMVS:
  - Provide a job name
  - Use customized start and stop commands
  - Consider your own monitoring routine

### APPLICATION INFO: NonMVS job type

NonMVS Applications (subsystems) can also be defined to SA z/OS. As the name suggests, NONMVS represents an automated resource that is not an MVS started task or job; for example, a database or a performance level.

More work is involved because some of the generic processing does not apply. This processing takes place, for example, when jobs start and stop or during MVS job monitoring. Specifically, you might have to specify the explicit start commands for starting the subsystem. You also might want to consider using a Monitor Option (or NONE). The default monitor option is to run the regular checking for an MVS job.

NonMVS subsystems require that you specify a job name. The job name is used when running commands such as ACTIVMSG and TERMMMSG. It serves to identify the associated subsystem, like the following one:

```
* Message that indicates nonMVS appl is UP  
IF MSGID = 'APPL001I' THEN  
EXEC (CMD('ACTIVMSG UP=YES,JOBNM=NMJOBNM')) ;
```

A dummy job name is used when running commands that identify status changes. You might also consider statements that indicate other statuses. Because there is no default routing provided by SA z/OS, carefully consider the autotask where these commands are run. They might all run under the AUTLOG task.

# RELATIONSHIPS policy

COMMANDS HELP

Define Relationship

Command ==> \_\_\_\_\_

|   |  |
|---|--|
| Entry Type : Application                | PolicyDB   |
| Entry Name : VTAM                       | Enterprise   |
| Subsystem Name : VTAM                   |  |
| Description . . . . . Depending no JES2 |  |
| Relationship Type . . HASPARENT         |  |
| Supporting Resource . . JES2/APL/=      |  |
| Sequence Number . . . . .               | Resource Name<br>Sequence Number (1-99, blank)       |
| Automation . . . . .                    | ACTIVE PASSIVE                                       |
| Chaining . . . . .                      | STRONG WEAK  |
| Condition . . . . .                     | Satisfy condition<br>(? for list of possible values) |

```

graph LR
    VTAM((VTAM)) -- "MakeAvailable / WhenAvailable" --> JES2((JES2))
  
```

**Current system**

Before VTAM can be started, JES2 must be available

## RELATIONSHIPS policy

This slide illustrates an example of defining a relationship between two resources: VTAM and JES2. A MakeAvailable request will be generated against VTAM when JES2 is available.

The *Relationship Type* can be one of the following types:

- **MakeAvailable:** Generates a MakeAvailable request to start a resource.
- **MakeUnavailable:** Generates a MakeUnavailable request to stop a resource.
- **PrepAvailable:** Prepares resource to be started.
- **PrepUnavailable:** Prepares resource to be stopped.
- **Forcedown:** For relationships to force shutdown in case the supporting resource reached the specified satisfy condition.
- **Externally:** For relationships to the externally started and stopped resources.
- **HasMonitor:** For relationships that define connections between Application or Application Group resources and monitor resources.

A start request is issued against the supporting resource (parent) if it is down and a stop request is issued against the supported resource (child).

- **HasParent:** For simple parent/child relationships.
- **HasPassiveParent:** For simple parent/child relationships with no vote propagation.
- **PeerOf:** 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 is specified.

*Automation* defines the processing to satisfy the relationship. It can be active or passive:

- **Active:** Specifies that SA z/OS will attempt to bring the resource to a desired condition to satisfy the relationship. ACTIVE is the default setting for MakeAvailable and MakeUnavailable.
- **Passive:** Specifies that SA z/OS will wait until the resource is in a desired condition before satisfying the relationship.

*Chaining* defines if only the status of the supporting resource, or the status of the attached subtree, is to be considered. It can be strong or weak:

- **Strong:** Specifies that SA z/OS considers the status of the attached subtree.
- **Weak:** Specifies that SA z/OS considers only the status of the supporting resource. This is the default setting for MakeAvailable and MakeUnavailable.

## List of conditions

|  |   |
|--|---|
| WhenObservedAssumedDown                  | WhenRunning   |
| WhenObservedAvailable                    | WhenRunningOrStarting   |
| WhenObservedDown                         | WhenStoppableOrAssumedDown  |
| WhenObservedHardDown                     | WhenStoppableOrDown   |
| WhenObservedRunning                      | WhenStoppableOrSoftdown   |
| WhenObservedSoftDown                     | WhenGroupHasNotFailed   |
| WhenObservedAssumedDownOrStopping        | WhenHardDown  |
| WhenObservedDownOrStopping               | WhenHealthAssumedNormal   |
| WhenObservedWasAvailable                 | WhenHealthFatal   |
| WhenObservedWasAvailableUnknownOrSysgone | WhenHealthNeitherNormalNorFatal   |
| WhenAssumedDown                          | WhenHealthNormal  |
| WhenAssumedDownOrStopping                | WhenHealthNotFatal  |
| WhenAvailable                            | WhenHealthNotNormal   |
| WhenAvailableOrStarting                  | Only for HASPARENT,<br>HASPASSIVEPARENT, EXTERNALLY:<br>• StartsMe<br>• StartsMeAndStopsMe<br>• StopsMe |
| WhenDown                                 |   |
| WhenDownOrStopping                       |   |
| WhenSoftDown                             |   |

Note: There is a default condition for each relationship.

Not all conditions are available for a specific relationship.

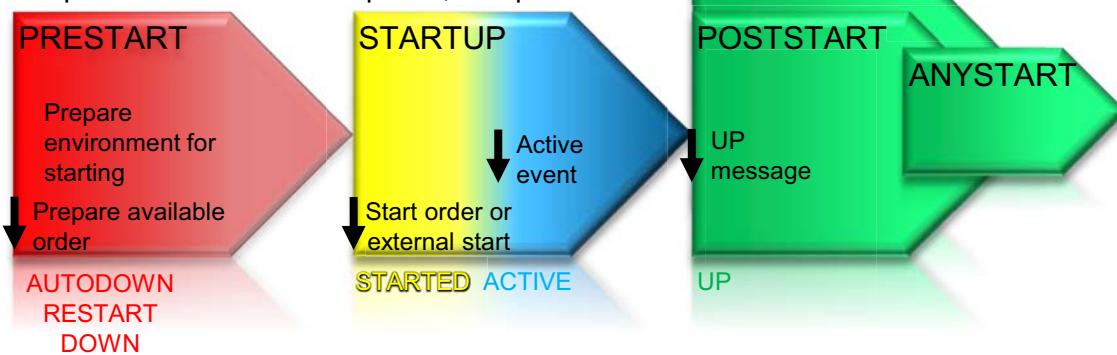
4-15

### *Relationship conditions*

Entering a question mark (?) in the Conditions field of the Define Relationships panel displays the list of possible relationship conditions. You can select the condition that the supporting resource must have to satisfy the relationship. Only one condition can be selected. If you select multiple conditions, the last selected condition is propagated to the preceding panel. If no condition is selected, the condition remains blank, causing the default condition of the relationship type to be taken. For example, you might choose to start another member of an Application Group when the Health Status of the APG is not normal. In that case, you use the WhenHealthNotNormal condition.

# Starting applications

- A start has three major phases:
  - PRESTART: Prepare environment for starting, if required
  - STARTUP: The actual start commands
  - Optionally, after startup is completed:
    - POSTSTART: The actions after the application is UP
    - REFRESHSTART: After a NetView recycle or INGAMS refresh only
    - ANYSTART: Issued after POSTSTART and after REFRESHSTART
- Different start types for each phase are possible, such as COLD, RECOVERY, and so on
- Multiple commands for each phase, if required



© Copyright IBM Corporation 2019

4-16

## STARTUP policy

When you define how a subsystem starts, you must look at all the ways in which it can be started. Starting an Application has three phases. The PRESTART phase begins when an Application acquires a Desired Status of AVAILABLE. If available, the PrepareAvailable condition must be fulfilled also. This can happen before the Application is ready to be started (it might never be started, in fact). The STARTUP phase is the actual start of the Application. The POSTSTART phase occurs when the Application is fully initialized (UP or ENDED). Additionally, the installation can specify different start types with different commands in all phases for such starts.

The *Start Timeout* is specified in the AUTOMATION INFO panel. For the Start Timeout, specify a value to include time for the UP message to be issued. If the Start Timeout is too short, you might see STARTED2 statuses, before eventually being superseded by UP statuses. The *Start Cycles* value is also important and works with the Start Timeout. Start Cycles specifies the number of times to cycle the Start Timeout period before posting the subsystem as a problem, meaning the Application has a STARTED2 or INACTIVE status. If no value is specified for Start Cycles, a value of 1 is assumed. Select the STARTUP policy item to specify how to start the Application.

## Start processing

- Commands can be specified as *must work successfully*
  - Enter an asterisk (\*) in AutoFn field to enable return code checking
- Related APPLICATION INFO policies are as follows:
  - *External Startup*: INITIAL, ALWAYS, NEVER, or null
  - *Start Timeout*: Checking for UP messages
  - *Start Cycles*: Number of times to schedule Start Timeout timer
    - Default is one (1)
    - When exceeded, resource status is set to STARTED2 or INACTIVE

### Start processing

Enter an asterisk (\*) in AutoFn field:

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.

The start processing is also determined by the listed, related APPLICATION INFO policies.

# STARTUP policy item

```

COMMANDS HELP
Subsystem Startup Processing
Command ==> _____
Entry Type : Application          PolicyDB Name   : OPSDEMO
Entry Name  : RV01                 Enterprise Name : OPSDEMO
Scheduling Subsystem . . .         (MSTR, JES Subsystem)
JCL Procedure Name . . . : RUNVAPL
Startup Parameters . . .

Line Commands: S/C (Cmd)
Cmd  Phase      Description
____ PRESTART    Executed before startup is initiated
____ STARTUP     Executed to initiate the startup
____ POSTSTART   Executed after startup has completed
____ REFRESHSTART Executed after recycle or INGAMS REFRESH
____ ANYSTART    Executed after POSTSTART and REFRESHSTART

```

Cmd  
1  
3  
1

STARTUP policy item for RV01 with PRESTART, STARTUP, and POSTSTART phases

## STARTUP policy item

Selecting the STARTUP policy item for an Application displays the *Subsystem Startup Processing* panel. You can select one of the three start phases and then define commands for that phase. Often only the STARTUP phase has definitions. In this panel, you enter the following information:

- **Scheduling Subsystem** (optional): The name of the subsystem that is to schedule this Application.
- **MSTR**: Indicates that the Application is scheduled directly by the z/OS master scheduler. The string, **SUB=MSTR** is appended to the default start command.
- **Blank (null)**: Indicates that the Application is scheduled by the primary JES.
- **Other**: Indicates that the Application is scheduled by another scheduling subsystem. The value entered is the name of that subsystem. The specified subsystem must be defined as being type JES2 or JES3.
- **JCL Procedure Name**: Specifies the name of the JCL procedure used to create this instance of the Application. If specified, the default start command for the Application is the following one:

**MVS S procname,JOBNAM=jobname**

- **Startup Parameters:** Specifies the subsystem start parameters. These values are added to the MVS START command for the subsystem. For example, the start command for VTAM might be **S VTAM,,, (LIST=00)**. If the start command is submitted from SA z/OS, the startup parameters field contains this information:

, , , (LIST=00)

Whatever is entered in the *Startup Parameters* field is appended directly to the MVS START command for the subsystem. Therefore, the three commas preceding (LIST=00) are necessary. These three fields are also part of the APPLICATION INFO policy item.

## PRESTART phase commands

```

COMMANDS HELP
Command Processing : PRESTART
Command ==> Line 00000001 Col 001 075
Mixed case . . . NO (YES NO)
Cmd Type AutoFn/* Command Text
s COLD RVCOLD RV01
***** Bottom of data *****

A0FGDYN7 Command Processing : line 1
Command ==> -
This panel displays the complete input fields of a single data line.
Press END to save changes or CANCEL to discard changes.
Type . . . . . COLD
Auto Function/*'*' . . .
Command Text . . . .
RVCOLD RV01

```

In this example, RV01 STARTUP policy has one PRESTART command for a COLD start.

Issue the RVCOLD command (REXX EXEC) for the RV01 application

### PRESTART phase commands

You use the PRESTART phase to run a command or commands before starting the Application. In this case, the PRESTART phase issues the RVCOLD (user-written REXX EXEC) command when a COLD start is requested.

# STARTUP phase commands

```

COMMANDS HELP
AOFGDYNC Command Processing : STARTUP Line 00000001 Col 001 075
Command ==> Scroll ==> CSR
Mixed case . . . NO (YES NO)
Cmd Type AutoFn/* Command Text
COLD MVS S RUNVAPL,JOBNAM=8SUBSJOB,NAME=VAPL21
NORM MVS S RUNVAPL,JOBNAM=8SUBSJOB,NAME=VAPL21
WARM MVS S RUNVAPL,JOBNAM=8SUBSJOB,NAME=VAPL21
***** Bottom of data *****

```

Runtime variable job name

RV01 STARTUP policy with three commands: NORM, WARM, and COLD

## STARTUP phase commands

On this slide, you see an example of the STARTUP phase commands for Application RV01. The STARTUP phase is where you define the commands that are issued to start the Application, based on start type.

If only a simple **MVS START jobname** command is required, no action is needed. This command is the default start command, and it is probably sufficient for many MVS Applications.

If a more complex **MVS START jobname** command is required, you can specify it by adding the rest of the command in the **Startup Parameters** field on the Subsystem Startup Processing panel. You can also include commands with the parameters as follows:

```

MVS S VTAM,,, (LIST=&LPAR.)
MVS S GTF.TEST,PARM='USR'
MVS S INGEAM,TYPE=COLD,SUB=MSTR

```

where &LPAR is a system symbolic set by the system programmer.

If an **MVS START jobname,SUB=MSTR** command is required, you can specify it as shown in the previous example or you can specify MSTR in the Scheduling Subsystem field on the Subsystem Startup Processing panel. If an **MVS START proc,JOBNAME=jobname** command is required, you

can specify it as shown in the previous example or you can specify the procedure name in the **JCL Procedure Name** field on the Subsystem Startup Processing panel. Combinations of these specifications are also valid.

If several commands are needed to start the Application, or if the command is not an MVS START command, specify the commands in the lower part of the panel. The full form of the command must be entered.

Each command can have a *Type* specification. Use this specification to issue a command for a specific type of start. This example has a different command defined for NORM, WARM, and COLD start types.

You can also specify that a command must run successfully; that is, it must give a return code of zero (0). Specify an asterisk (\*) in the **Automation Function** field to enable return code checking. If it is not successful, no further commands are entered by the agent for that phase, and the Application goes to an Automation Status of PROBLEM. In the example, the first command (to vary a device) is selected. If the device is not online, the STARTUP phase does not continue.

Some subsystems are started by their parents. Examples include some TCP/IP address spaces, TMON, and OMEGAMON. In this case, you must specify that the start is *external* (in the APPLICATION INFO policy) and provide a StartsMe condition.

You can specify the *automation function*, for example, SHUTOPER, to indicate which autotask is used to issue the start command. By default, all commands run on the AUTWRKxx operator that is assigned to that Application. In such a case, all commands run in the order in which you have specified them. You can also request that different commands run on different operators. In these cases, you cannot predict the order in which these commands will run.

STARTUP commands include cases where a REXX EXEC is run to do everything that is required for a correct start of the Application. Enter the EXEC name. Messages can be specified as UP messages.

When using MVS or VTAM commands from NetView, you must use the **Pipe** command to retrieve the return code from the command. Otherwise, the return code is always zero (0), indicating that the command was run.

# POSTSTART phase commands

```

COMMANDS HELP
AOFGDYNC Command Processing : POSTSTART
Command ===> Line 00000001 Col 001 075
Mixed case . . . NO (YES NO)
Cmd Type AutoFn/* Command Text
__ COLD RVPOST COLD RV01
***** Bottom of data *****

```

- In this example, RV01 STARTUP policy has one POSTSTART command for a COLD start
- Issue the RVPOST command (REXX EXEC) with parameters COLD and RV01

© Copyright IBM Corporation 2019

4-21

## POSTSTART phase commands

You can use the POSTSTART phase to run a command or commands after the Application is started. In this case, the POSTSTART phase will run the user-written RVPOST command (REXX EXEC). The POSTSTART commands will run when the Application reaches the UP status, and if it was started by automation.

## External start considerations

- Define parent or start dependency
- Select APPLICATION INFO policy external startup options.
  - ALWAYS: Parent always starts application
  - INITIAL: Parent initially starts application, but automation can start it afterward
  - NEVER: Application is always started by SA z/OS
- In the RELATIONSHIPS policy, create a HasParent/StartsMeAndStopsMe relationship to supporting resource (parent)

### *External start considerations*

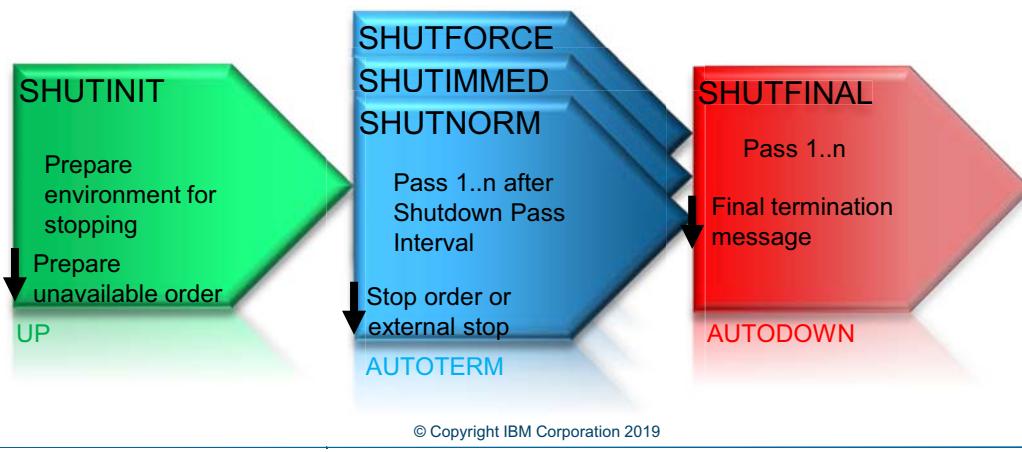
When the Application start is performed by the parent, then a special technique is required. You create the HasParent (or MakeAvailable) relationship. Then, you select the APPLICATION INFO policy item for the (child) Application. In the child Application, you specify one of ALWAYS or INITIAL for the *External Startup* value. (A value of NEVER indicates that start commands is typically issued directly against the child Application.)

A value of ALWAYS is specified when the parent always starts the Application. In this case, automation does not directly start the Application. Even if a request is made to start the Application through automation, no start commands are issued for that Application. PRESTART and POSTSTART commands are issued, however.

A value of INITIAL is specified when automation can issue commands to start the Application. However, in an initial start, when the parent is also being started, the automation expects the start of the parent to also start the Application. In the Relationships policy item, you create a HasParent/StartsMeAndStopsMe relationship to the parent.

# Stopping applications

- A stop has three major phases:
  - SHUTINIT: Prepare environment for stopping , if required
  - SHUTxxxx: The actual stop commands
    - NORM: Normal shutdown
    - IMMED: Immediate shutdown
    - FORCE: Force a subsystem off the system without any delay
  - SHUTFINAL: Activities after stop is completed, if required
- Multiple passes for SHUTxxxx and SHUTFINAL, if required



© Copyright IBM Corporation 2019

4-23

## SHUTDOWN policy

When defining how a subsystem is stopped, consider all the ways in which it can be stopped. Remember that the automation agent issues the command or commands that are indicated in the policy. An Application is stopped in three phases. The SHUTINIT phase occurs as soon as an Application acquires a Desired Status of UNAVAILABLE. The prepare unavailable condition must be fulfilled also. This phase can occur before the Application is ready to be correctly stopped (it might never be stopped, in fact). The actual shutting down happens in one of three ways: NORM, IMMED, and FORCE. The SHUTFINAL phase occurs when the Application becomes inactive. Select the SHUTDOWN policy item to specify how to stop the Application.

# Stop processing

- Commands
  - NetView command
  - Reply to outstanding WTOR
  - REXX routine
- Related APPLICATION INFO policies
  - External Shutdown: FINAL, ALWAYS, NEVER, or null
  - Shutdown Pass Interval: Time between passes
  - Cleanup Delay: Time to wait before processing final termination message

## Stop processing

Within any shutdown phase, several passes, each with a set of commands, are defined. The *shutdown delay* is the time between shutdown passes. Shutdown delay is specified on the APPLICATION INFO policy panel. The commands defined for a pass can be NetView commands, replies to an Application outstanding WTOR, or REXX EXECs.

# SHUTDOWN phases

| COMMANDS                                       |       | HELP  |     |     |
|--|-------|---|-----|-----|
| AOFGASHT                                       |       | Subsystem Shutdown Processing               |     |     |
| Command ==>                                    |       |   |     |     |
| Entry Type : Application                       |       | PolicyDB Name : NEW_PDB                     |     |     |
| Entry Name : JES2                              |       | Enterprise Name : SAPLEX                    |     |     |
| Shutdown Pass Interval . . 00:02:00 (hh:mm:ss) |       |   |     |     |
| Line Commands: S/C (Cmd), R (Rep)              |       |   |     |     |
| Cmd  | Phase | Description                                 | Cmd | Rep |
|  | INIT  | Executed when shutdown is initiated         | 4   |     |
|  | NORM  | Executed when normal shutdown is invoked    | 1   |     |
|  | IMMED | Executed when immediate shutdown is invoked | 2   |     |
|  | FORCE | Executed when force shutdown is invoked     | 1   |     |
|  | FINAL | Executed after final termination message    | 1   |     |

Shutdown phases for JES2:

SHUTINIT, SHUTxxxx (NORM, IMMED, FORCE), and SHUTFINAL

## SHUTDOWN phases

There are a maximum of five shutdown phases, as shown on this slide. The SHUTINIT specifies commands that are typically issued when the Desired Status of the Application becomes UNAVAILABLE. It must not include any commands that cause the address space to terminate. For example, SHUTINIT can be used to send warning messages, block new requests, or close internal processes. SHUTFINAL indicates commands that can be run when the Application becomes inactive as a result of an automated shutdown.

The SHUTDOWN phase has three possible types: NORM, IMMED and FORCE. The type is specified by an operator during an INGREQ STOP command. The type is typically used for increasingly drastic shutdown methods. For example, a SHUTNORM might reply to an outstanding WTOR to end the Application. A SHUTIMMED might issue an MVS CANCEL command to end the Application. For each shutdown type, enter **CMD** to define commands, or **REPLY** to define replies to WTOR messages. Then, specify the pass that the command or reply refers to and the actual command or reply. The command is any NetView command or EXEC. Remember to specify the MVS prefix if it is an MVS command (common mistake). REPLY is not supported for FINAL.

## JES2 SHUTINIT and SHUTNORM commands

```
AOFGDYNC      Command Processing : SHUTINIT          Line 00000001 Col 001 075
Command ===> _____ Scroll ===> CSR

Mixed case . . . NO (YES NO)                         Runtime variable application name

Cmd Type      AutoFn/* Command Text
_____| JESOPER DRAINJES &SUBSAPPL
_____| JESOPER MVS F BPXOINIT,SHUTDOWN=FORKS
_____| JESOPER INGRCLUP AXR0*,*
_____| JESOPER PIPE NETV INGJLM STATUS ! LOC /INGTJLM: ACTIVE/ ! EDIT /S
***** Bottom of data *****
```

SHUTINIT: 1 Issue **DRAINJES JES2** when a stop of JES2 is requested 2 Disable forks in OMVS 3 Cancel all REXX system address spaces 4 Stop joblog monitoring

```
AOFGDYNC      Command Processing : SHUTIMMED         Line 00000001 Col 001 075
Command ===> _____ Scroll ===> CSR

Mixed case . . . NO (YES NO)                         Runtime variable application command prefix
SUBSCMDPFX followed by P

Cmd Ps AutoFn/* Command Text
_____| 1 MVS &SUBSCMDPFXP&SUBS.JOB
_____| 2 MVS &SUBSCMDPFXP&SUBS.JOB,ABEND
***** Bottom of data *****
```

SHUTIMMED:

Pass 1, issue **MVS \$PJES2** command

Pass 2, issue **MVS \$PJES2,ABEND** command

Job name

### TSO SHUTINIT and SHUTNORM commands

In this example, the SHUTINIT phase defines:

- Issue DRAINJES JES2 when a stop of JES2 is requested (Explanation see later).  
Runtime variable &SUBSAPPL application name is used
- Disable forks in OMVS
- Cancel all REXX system address spaces using ingrclup axr0\*,\*
- Stop joblog monitoring

The SHUTNORM phase contains two passes for shutting down JES2. If the first pass fails to shut down JES2 the next pass is executed. The next pass could be also skipped then no action is taken for the skipped pass. In this way, you can define different intervals for actions between the passes. Then, the MVS \$PJES2,ABEND command is issued for the second pass.

A typical way to shut down a subsystem is to use the **MVS STOP** and **CANCEL** commands, especially for a normal shutdown.

You might want commands to be issued by specific operators. To do this, enter the automation operator name (for instance, **SHUTOPER**) in the Automated Function column. This name is not the autotask name. The default is the work operator used for that Application. You can also specify an

asterisk (\*) to indicate that the command must succeed; that is, it must give a return code of zero (0). If it does not succeed, no more commands are issued for that phase, and the Automation Status of the Application is set to PROBLEM.

As a last pass for started tasks, you might add an **MVS FORCE** command. However, because of the effect of **FORCE** commands, an alternative approach is to code the **FORCE** as part of an **IMMED** or **FORCE** shutdown definition. Then, FORCE is only started specifically, by an operator, after the failure of a NORM shutdown and not as the normal shutdown.

Note again the use of &SUBSJOB to indicate the *subsystem job name*. There are other variables that you can use in this way. Use of symbolic variables is documented in the *IBM System Automation for z/OS Programmer's Reference* "AOCQRY Subsystem Task Global Variables".

## SHUTNORM: Reply and command example for NetView

AOFGDYN9      Reply Processing : \$SHUTNORM  
Command ==>

Cmd Ps C    Reply Text  
1            CLOSE STOP

SHUTNORM policy for “another” NetView

Line 00000001 Col 001 075  
1 Respond CLOSE STOP to the  
outstanding WTOR

Scroll ==> PAGE

AOFGDYNC      Command Processing : \$SHUTNORM  
Command ==>

Mixed case . . . NO (YES NO)

Cmd Ps AutoFn/\* Command Text  
1            MVS P &SUBSJOB  
2            CLOSE STOP  
3            CLOSE IMMED  
4            MVS C &SUBSJOB

SHUTNORM policy for SA NetView

- 1 MVS Stop command against jobname
- 2 Issue the CLOSE STOP command
- 3 Issue the CLOSE IMMED command
- 4 MVS Cancel command against jobname

© Copyright IBM Corporation 2019

4-27

### SHUTNORM: Reply and command example for NetView

You can have multiple passes using a mix of CMD (command) and REP (reply) actions. REP replies to the current outstanding WTOR issued by the Application. You can have many commands or replies in a pass. If you enter several commands in a pass, they are run in the order specified.



**Important:** Commands that are specified are issued before replies within the same pass.

The first example shows the SHUTNORM pass for “another” NetView using a reply.

To have the reply processed first, define it as PASS1. It will be a reply to the outstanding DSI802A (or DSI803A) WTOR message from NetView. The reply text is CLOSE STOP which logs off all tasks to bring NetView down.

The second example shows the SHUTNORM passes for the SA NetView in which the command is executed using:

1. MVS Stop command against jobname
2. Issue the CLOSE STOP command

3. Issue the CLOSE IMMED command
4. MVS Cancel command against jobname

The second pass is issued if NetView is active after the first *shutdelay* interval. If NetView is active, then the CLOSE STOP command is issued for the second pass and so on.

#### Shut down the NetView program in this sequence:

1. Issue a CLOSE NORMAL command to notify operators of the impending shutdown and prevent new operators from logging on. Because autotasks are not directly affected, the NetView program does not shut down without further action.
2. After determining that operators are not adversely affected, issue the CLOSE STOP command. In all usual circumstances, the NetView program will close promptly, after allowing for global keep ENDCMD statements to process. For additional information, see the endcmd.close.leeway CNMSTYLE statement in IBM Tivoli NetView for z/OS Administration Reference.

If you issue a CLOSE IMMED command before issuing the CLOSE STOP command, the NetView program automatically converts the option to STOP. Issuing a second CLOSE IMMED command renews requests for all tasks to terminate and sets one-minute timers to issue user abends if necessary. This same logic applies to CLOSE commands issued by replying to the NetView DSI802A WTOR message. Processing for the MVS STOP (P) command follows similar processing: The first instance acts like a CLOSE STOP command and the second instance acts like a CLOSE IMMED command.

3. If you determine that the NetView program cannot shut down properly or if you require a memory dump, issue the CLOSE ABEND or CLOSE DUMP command.

## SHUTFINAL commands

```
Command Processing : line 1
Command ==> _____ Scroll ==> CSR

This panel displays the complete input fields of a single data line.

Type . . . . . . . . . . (NORM IMMED FORCE)
AutoFn/* . . . . . . . . JESOPER
Command Text . . . . . .
MVS F BPX0INIT,SHUTDOWN=FILESYS
```

After the stop of JES2 completes, the OMVS file system is shut down

### SHUTFINAL commands

The commands specified for the SHUTFINAL phase are run after the final termination message is received. For this shutdown phase, only commands can be specified.

Here, after the stop of JES2 completes, the OMVS file system is shut down.

## JES2 shutdown example

| Phase            | Pass | Commands to issue   |
|------------------|------|---|
| <b>SHUTINIT</b>  |      | DRAINJES &SUBSAPPL<br>MVS F BPXOINIT,SHUTDOWN=FORKS<br>INGRCLUP AXR0*,*<br>PIPE NETV INGJLM STATUS   LOC /INGTJLM:<br>ACTIVE/   EDIT /STOP TASK=INGTJLM/ 1   NETV |
| <b>SHUTNORM</b>  | 4    | MVS &SUBSCMDPFXP&SUBSJOB<br>(MVS \$PJES2)   |
| <b>SHUTIMMED</b> | 1    | MVS &SUBSCMDPFXP&SUBSJOB<br>(MVS \$PJES2)   |
|                  | 2    | MVS &SUBSCMDPFXP&SUBSJOB,ABEND<br>(MVS \$PJES2,ABEND)   |
| <b>SHUTFORCE</b> | 1    | MVS &SUBSCMDPFXP&SUBSJOB,ABEND  |
| <b>SHUTFINAL</b> |      | MVS F BPXOINIT,SHUTDOWN=FILESYS   |

### JES2 shutdown example

On this slide, you see an example of the JES2 shutdown phases. Commands specified in the SHUTINIT are typically run as soon as the Desired Status is changed to UNAVAILABLE (when SA z/OS receives information that the subsystem will be shut down). If many subsystems are involved in the request (INGREQ ALL REQ=DOWN), this shutdown is performed when the request is entered. It might be a while before the commands in the PASS1 definition for JES2 are run. By running a **DRAINJES** command (a standard SA z/OS command), the initiators and other JES tasks can be drained as soon as possible. Note the repeated use of the **\$PJES2** command to shut down JES2. If the JES2 internal tasks are slow to end, you might have to reenter the **\$PJES2** command to finish the shutdown. If this fails after a number of passes, a **\$PJES2,ABEND** forces it down. The interval between SHUTNORM passes is defined in *Shutdown Pass Interval* of APPLICATION INFO policy item, for example, two minutes. The JES2 DRAIN policy item indicates what JES2 tasks are to be drained when a DRAINJES command is issued. Usually, all tasks are selected. The DRAINJES command then determines active selected resources in JES and issues a **\$P** request for each. The DRAINJES command can be issued at any time by the operator, or automatically if it is part of the SHUTINIT process. The DRAINJES process does not include handling of initiators.

## Stop considerations

- SHUTINIT can be independent from shutdown
- Specify NORMAL shutdown for all subsystems:
  - A NORMAL shutdown is the default, therefore it should be defined
  - Use shutdown passes with more aggressive commands, like force. Time between passes is defined in Shutdown Pass Interval
  - Existence of start or stop commands is checked (PRECHECK=YES) by the agent before sending the request to the AM
    - Warning message AOF757I issued
    - When SA z/OS runs out of shutdown commands of the selected type, the agent status is set to STUCK
- IMMED or FORCE shutdowns skip some of the "polite" stop commands
- SHUTFORCE is often identical to SHUTIMMED

© Copyright IBM Corporation 2019

4-30

### Stop considerations

All subsystems should have a normal shutdown defined so that an **INGREQ ALL REQ=DOWN** (with the default stop type of NORM) command is effective. The only exceptions are those subsystems that are always shut down by their parent, and those that do not have to be shut down. If no NORM shutdown is specified, and **INGREQ STOP TYPE=NORM** is issued, the subsystem moves to the STUCK status immediately, indicating that it has run out of shutdown passes. If you specify SHUTINIT commands for a subsystem for which a INGREQ STOP command is issued and then canceled, remember that the SHUTINIT commands can be issued even though the subsystem might not have entered the AUTOTERM status.

If you use a command like **INGREQ xxx REQ=STOP**, the existence of start or stop commands is checked (default is PRECHECK=YES) by the agent before sending the request to the AM. Message AOF757I is displayed if no NORMAL shutdown is specified as the default is PRECHECK=YES. The request is rejected.

Use shutdown passes with more aggressive commands, like force. The time between passes is defined in the Shutdown Pass Interval. When SA z/OS runs out of shutdown commands of the selected type, the agent status is set to STUCK.

IMMED or FORCE shutdowns skip some of the "polite" stop commands.  
SHUTFORCE is often identical to SHUTIMMED.

## Stop by parent

If the “parent” (supporting resource) stops a child (dependent resource), you define at the child level:

- APPLICATION INFO policy *external shutdown* field
  - ALWAYS: Parent always stops application
  - FINAL: Automation can stop application alone, but parent does the final shutdown
  - NEVER: Application always stopped by SA z/OS
- RELATIONSHIPS policy
  - Create a HasParent or MakeUnavailable relationship with condition StopsMe to the supporting resource (parent)

### Stop by parent

When the shutdown of the Application is performed by a parent, then a special technique is required. You select in the APPLICATION INFO policy for the (child) application. Specify either ALWAYS or FINAL for the **External Shutdown** value. A value of NEVER indicates that shutdowns happen when commands are run that affect the child Application directly.

A value of ALWAYS is specified when the parent always stops the Application. In this case, automation never tries to directly stop the Application. Even if a request is made to stop the Application through automation, no Shutdown commands are issued. SHUTINIT and SHUTFINAL commands are issued, however.

A value of FINAL is specified when automation can issue commands to shut down the Application directly. However, in a final shutdown, when the parent is also shut down, the parent can shut down the child Application. If the child is running after the parent has ended, automation issues the commands to stop the child.

In the RELATIONSHIPS policy item, you create a HasParent or MakeUnavailable relationship with condition StopsMe to the parent.

# MESSAGES/USER DATA policy

- Messages can be issued as part of start, stop, or normal operation:
  - Identify some action required, often to continue the start or stop process
  - You can use WTOs and WTORs; they can be specific to a start or stop process
- Default Automation Table and message revision table entries are provided
  - To view them, select the MESSAGE AUTOMATION policy item under entry type MSG (option 21)
- Message automation is defined by the Application MESSAGES/USER DATA policy actions
  - The actions can be commands or replies
  - The actions can be used, for example, to set the resource status, define Automation Table overrides, or define message revision table entries
  - The actions are controlled by using agent automation flags

## MESSAGES/USER DATA policy

When an Application starts up or shuts down, it might issue messages indicating, for example, that part of the process is complete or additional operator action is required. A specific example arises with WTORs. They usually require some responses before the start or stop can resume.

An example is the TSO message IKT010D, which is issued during shutdown. This message indicates that users are logged on. You must indicate whether or not they must be canceled. A common reply to this WTOR is to enter SIC (for System Initiated Cancel). Such a reply can be entered into the policy and used when the WTOR is issued. Alternatively, when a specific message is issued, you must enter additional commands.

You can specify these commands and responses using the MESSAGES/USER DATA policy item for the Application definition. Select the MESSAGES/USER DATA policy item and create an entry using the message ID as an identifier. Specify the action to take, as you do for SHUTDOWN requests.

## Default message entries MSG (option 21)

| AOFGMSLT  |  | Message Definitions               | Row 457 to 476 of 694 |          |                           |                 |                   |               |          |                      |   |  |            |                                   |      |                     |  |           |                            |  |         |                           |    |  |         |                           |   |  |         |            |   |     |         |                 |   |  |         |                    |   |  |         |             |   |  |         |                            |   |  |         |           |   |  |         |           |   |  |         |           |   |  |         |                 |   |
|---|--|-----------------------------------|-----------------------|----------|---------------------------|-----------------|-------------------|---------------|----------|----------------------|---|--|------------|-----------------------------------|------|---------------------|--|-----------|----------------------------|--|---------|---------------------------|----|--|---------|---------------------------|---|--|---------|------------|---|-----|---------|-----------------|---|--|---------|--------------------|---|--|---------|-------------|---|--|---------|----------------------------|---|--|---------|-----------|---|--|---------|-----------|---|--|---------|-----------|---|--|---------|-----------------|---|
| Command ==> _____   |  | Scroll ==> CSR                    |                       |          |                           |                 |                   |               |          |                      |   |  |            |                                   |      |                     |  |           |                            |  |         |                           |    |  |         |                           |   |  |         |            |   |     |         |                 |   |  |         |                    |   |  |         |             |   |  |         |                            |   |  |         |           |   |  |         |           |   |  |         |           |   |  |         |                 |   |
| PolicyDB Name : NEW_PDB   |  | Enterprise Name : SAPLEX          |                       |          |                           |                 |                   |               |          |                      |   |  |            |                                   |      |                     |  |           |                            |  |         |                           |    |  |         |                           |   |  |         |            |   |     |         |                 |   |  |         |                    |   |  |         |             |   |  |         |                            |   |  |         |           |   |  |         |           |   |  |         |           |   |  |         |                 |   |
| Make a selection ( S ) or use shortcuts ( WHU AC AS AO MS MO MF IG ).   |  |                                   |                       |          |                           |                 |                   |               |          |                      |   |  |            |                                   |      |                     |  |           |                            |  |         |                           |    |  |         |                           |   |  |         |            |   |     |         |                 |   |  |         |                    |   |  |         |             |   |  |         |                            |   |  |         |           |   |  |         |           |   |  |         |           |   |  |         |                 |   |
| <table> <thead> <tr> <th>Cmd</th> <th>Message ID</th> <th>Description</th> <th>AT M F</th> </tr> </thead> <tbody> <tr><td></td><td>IEFC452I</td><td>JCL error</td><td>0</td></tr> <tr><td></td><td>IEF238D</td><td>Cannot complete device allocation</td><td>0</td></tr> <tr><td></td><td>IEF251I</td><td>JCL error</td><td>0</td></tr> <tr><td></td><td>IEF352I</td><td>Address space unavailable</td><td>CA</td></tr> <tr><td></td><td>IEF378I</td><td>CATALOG disposition error</td><td>0</td></tr> <tr><td></td><td>IEF402I</td><td>Job failed</td><td>0</td></tr> <tr><td>WHU</td><td>IEF403I</td><td>Job is starting</td><td>0</td></tr> <tr><td></td><td>IEF404I</td><td>Job has terminated</td><td>0</td></tr> <tr><td></td><td>IEF450I</td><td>Job abended</td><td>0</td></tr> <tr><td></td><td>IEF451I</td><td>Job cancelled by cond code</td><td>0</td></tr> <tr><td></td><td>IEF452I</td><td>JCL error</td><td>0</td></tr> <tr><td></td><td>IEF453I</td><td>JCL error</td><td>0</td></tr> <tr><td></td><td>IEF743I</td><td>Job abend</td><td>0</td></tr> <tr><td></td><td>IFB040I</td><td>LOGREC recovery</td><td>0</td></tr> </tbody> </table> |  |                                   |                       | Cmd      | Message ID                | Description     | AT M F            |               | IEFC452I | JCL error            | 0 |  | IEF238D    | Cannot complete device allocation | 0    |                     | IEF251I                                | JCL error | 0                          |  | IEF352I | Address space unavailable | CA |  | IEF378I | CATALOG disposition error | 0 |  | IEF402I | Job failed | 0 | WHU | IEF403I | Job is starting | 0 |  | IEF404I | Job has terminated | 0 |  | IEF450I | Job abended | 0 |  | IEF451I | Job cancelled by cond code | 0 |  | IEF452I | JCL error | 0 |  | IEF453I | JCL error | 0 |  | IEF743I | Job abend | 0 |  | IFB040I | LOGREC recovery | 0 |
| Cmd   | Message ID                             | Description                       | AT M F                |          |                           |                 |                   |               |          |                      |   |  |            |                                   |      |                     |  |           |                            |  |         |                           |    |  |         |                           |   |  |         |            |   |     |         |                 |   |  |         |                    |   |  |         |             |   |  |         |                            |   |  |         |           |   |  |         |           |   |  |         |           |   |  |         |                 |   |
|   | IEFC452I                               | JCL error                         | 0                     |          |                           |                 |                   |               |          |                      |   |  |            |                                   |      |                     |  |           |                            |  |         |                           |    |  |         |                           |   |  |         |            |   |     |         |                 |   |  |         |                    |   |  |         |             |   |  |         |                            |   |  |         |           |   |  |         |           |   |  |         |           |   |  |         |                 |   |
|   | IEF238D                                | Cannot complete device allocation | 0                     |          |                           |                 |                   |               |          |                      |   |  |            |                                   |      |                     |  |           |                            |  |         |                           |    |  |         |                           |   |  |         |            |   |     |         |                 |   |  |         |                    |   |  |         |             |   |  |         |                            |   |  |         |           |   |  |         |           |   |  |         |           |   |  |         |                 |   |
|   | IEF251I                                | JCL error                         | 0                     |          |                           |                 |                   |               |          |                      |   |  |            |                                   |      |                     |  |           |                            |  |         |                           |    |  |         |                           |   |  |         |            |   |     |         |                 |   |  |         |                    |   |  |         |             |   |  |         |                            |   |  |         |           |   |  |         |           |   |  |         |           |   |  |         |                 |   |
|   | IEF352I                                | Address space unavailable         | CA                    |          |                           |                 |                   |               |          |                      |   |  |            |                                   |      |                     |  |           |                            |  |         |                           |    |  |         |                           |   |  |         |            |   |     |         |                 |   |  |         |                    |   |  |         |             |   |  |         |                            |   |  |         |           |   |  |         |           |   |  |         |           |   |  |         |                 |   |
|   | IEF378I                                | CATALOG disposition error         | 0                     |          |                           |                 |                   |               |          |                      |   |  |            |                                   |      |                     |  |           |                            |  |         |                           |    |  |         |                           |   |  |         |            |   |     |         |                 |   |  |         |                    |   |  |         |             |   |  |         |                            |   |  |         |           |   |  |         |           |   |  |         |           |   |  |         |                 |   |
|   | IEF402I                                | Job failed                        | 0                     |          |                           |                 |                   |               |          |                      |   |  |            |                                   |      |                     |  |           |                            |  |         |                           |    |  |         |                           |   |  |         |            |   |     |         |                 |   |  |         |                    |   |  |         |             |   |  |         |                            |   |  |         |           |   |  |         |           |   |  |         |           |   |  |         |                 |   |
| WHU   | IEF403I                                | Job is starting                   | 0                     |          |                           |                 |                   |               |          |                      |   |  |            |                                   |      |                     |  |           |                            |  |         |                           |    |  |         |                           |   |  |         |            |   |     |         |                 |   |  |         |                    |   |  |         |             |   |  |         |                            |   |  |         |           |   |  |         |           |   |  |         |           |   |  |         |                 |   |
|   | IEF404I                                | Job has terminated                | 0                     |          |                           |                 |                   |               |          |                      |   |  |            |                                   |      |                     |  |           |                            |  |         |                           |    |  |         |                           |   |  |         |            |   |     |         |                 |   |  |         |                    |   |  |         |             |   |  |         |                            |   |  |         |           |   |  |         |           |   |  |         |           |   |  |         |                 |   |
|   | IEF450I                                | Job abended                       | 0                     |          |                           |                 |                   |               |          |                      |   |  |            |                                   |      |                     |  |           |                            |  |         |                           |    |  |         |                           |   |  |         |            |   |     |         |                 |   |  |         |                    |   |  |         |             |   |  |         |                            |   |  |         |           |   |  |         |           |   |  |         |           |   |  |         |                 |   |
|   | IEF451I                                | Job cancelled by cond code        | 0                     |          |                           |                 |                   |               |          |                      |   |  |            |                                   |      |                     |  |           |                            |  |         |                           |    |  |         |                           |   |  |         |            |   |     |         |                 |   |  |         |                    |   |  |         |             |   |  |         |                            |   |  |         |           |   |  |         |           |   |  |         |           |   |  |         |                 |   |
|   | IEF452I                                | JCL error                         | 0                     |          |                           |                 |                   |               |          |                      |   |  |            |                                   |      |                     |  |           |                            |  |         |                           |    |  |         |                           |   |  |         |            |   |     |         |                 |   |  |         |                    |   |  |         |             |   |  |         |                            |   |  |         |           |   |  |         |           |   |  |         |           |   |  |         |                 |   |
|   | IEF453I                                | JCL error                         | 0                     |          |                           |                 |                   |               |          |                      |   |  |            |                                   |      |                     |  |           |                            |  |         |                           |    |  |         |                           |   |  |         |            |   |     |         |                 |   |  |         |                    |   |  |         |             |   |  |         |                            |   |  |         |           |   |  |         |           |   |  |         |           |   |  |         |                 |   |
|   | IEF743I                                | Job abend                         | 0                     |          |                           |                 |                   |               |          |                      |   |  |            |                                   |      |                     |  |           |                            |  |         |                           |    |  |         |                           |   |  |         |            |   |     |         |                 |   |  |         |                    |   |  |         |             |   |  |         |                            |   |  |         |           |   |  |         |           |   |  |         |           |   |  |         |                 |   |
|   | IFB040I                                | LOGREC recovery                   | 0                     |          |                           |                 |                   |               |          |                      |   |  |            |                                   |      |                     |  |           |                            |  |         |                           |    |  |         |                           |   |  |         |            |   |     |         |                 |   |  |         |                    |   |  |         |             |   |  |         |                            |   |  |         |           |   |  |         |           |   |  |         |           |   |  |         |                 |   |
| <table> <thead> <tr> <th>AOFGMSL1</th> <th>Entries linked to Message</th> <th>Row 1 to 1 of 1</th> </tr> <tr> <th>Command ==&gt; _____</th> <th>SCROLL==&gt; CSR</th> <th></th> </tr> </thead> <tbody> <tr> <td colspan="2">Message ID : IEF403I</td><td></td></tr> <tr> <td>Entry name</td><td>Short Description</td><td>Type</td></tr> <tr> <td>+SA_PREDEFINED_MSGS</td><td>Messages predefined by SA for AT build</td><td>MVC</td></tr> <tr> <td colspan="3">***** Bottom of data *****</td></tr> </tbody> </table>  |  |                                   |                       | AOFGMSL1 | Entries linked to Message | Row 1 to 1 of 1 | Command ==> _____ | SCROLL==> CSR |          | Message ID : IEF403I |   |  | Entry name | Short Description                 | Type | +SA_PREDEFINED_MSGS | Messages predefined by SA for AT build | MVC       | ***** Bottom of data ***** |  |         |                           |    |  |         |                           |   |  |         |            |   |     |         |                 |   |  |         |                    |   |  |         |             |   |  |         |                            |   |  |         |           |   |  |         |           |   |  |         |           |   |  |         |                 |   |
| AOFGMSL1  | Entries linked to Message              | Row 1 to 1 of 1                   |                       |          |                           |                 |                   |               |          |                      |   |  |            |                                   |      |                     |  |           |                            |  |         |                           |    |  |         |                           |   |  |         |            |   |     |         |                 |   |  |         |                    |   |  |         |             |   |  |         |                            |   |  |         |           |   |  |         |           |   |  |         |           |   |  |         |                 |   |
| Command ==> _____   | SCROLL==> CSR                          |                                   |                       |          |                           |                 |                   |               |          |                      |   |  |            |                                   |      |                     |  |           |                            |  |         |                           |    |  |         |                           |   |  |         |            |   |     |         |                 |   |  |         |                    |   |  |         |             |   |  |         |                            |   |  |         |           |   |  |         |           |   |  |         |           |   |  |         |                 |   |
| Message ID : IEF403I  |  |                                   |                       |          |                           |                 |                   |               |          |                      |   |  |            |                                   |      |                     |  |           |                            |  |         |                           |    |  |         |                           |   |  |         |            |   |     |         |                 |   |  |         |                    |   |  |         |             |   |  |         |                            |   |  |         |           |   |  |         |           |   |  |         |           |   |  |         |                 |   |
| Entry name  | Short Description                      | Type                              |                       |          |                           |                 |                   |               |          |                      |   |  |            |                                   |      |                     |  |           |                            |  |         |                           |    |  |         |                           |   |  |         |            |   |     |         |                 |   |  |         |                    |   |  |         |             |   |  |         |                            |   |  |         |           |   |  |         |           |   |  |         |           |   |  |         |                 |   |
| +SA_PREDEFINED_MSGS   | Messages predefined by SA for AT build | MVC                               |                       |          |                           |                 |                   |               |          |                      |   |  |            |                                   |      |                     |  |           |                            |  |         |                           |    |  |         |                           |   |  |         |            |   |     |         |                 |   |  |         |                    |   |  |         |             |   |  |         |                            |   |  |         |           |   |  |         |           |   |  |         |           |   |  |         |                 |   |
| ***** Bottom of data *****  |  |                                   |                       |          |                           |                 |                   |               |          |                      |   |  |            |                                   |      |                     |  |           |                            |  |         |                           |    |  |         |                           |   |  |         |            |   |     |         |                 |   |  |         |                    |   |  |         |             |   |  |         |                            |   |  |         |           |   |  |         |           |   |  |         |           |   |  |         |                 |   |

Default message entries

Messages are critical for SA z/OS automation. To display the list of the messages that are automated by SA z/OS, perform these steps:

1. Select the MSG policy (option 21 in the Entry Type Selection panel).
2. Select the MESSAGE AUTOMATION policy item.
3. Locate the IEF messages with command:

L IEF

The status columns AT, M, and F provide the following information:

1. Column AT shows whether there are specifications for the NetView Automation Table build.
2. Column M shows whether there are specifications for the NetView Message Revision build.
3. Column F shows whether there are specifications for the MPF build.

The indications in these columns have the following meaning:

1. I - "Ignore" is specified for this message.
2. C - An AT condition is specified for this message.

3. A - A definition for this message causes a specific AT or MRT action or MPF entry. For the AT this is a status selection, for the MRT this is a specification resulting in a REVISE statement, and for the MPF this is a specification for the MPFLSTxx entry.
4. O - An AT or MRT override is specified for this message.

The following line commands are available on this panel:

1. Select AO (Automation overrides) to display the Automation Table and MRT entries, including any user-defined overrides.
2. Select WHU (Where used) to display the resources that are associated with a message.  
AC - shortcut to the "Automation Table entry Conditions" panel.
3. AS - shortcut to the "AT Status Specification" panel.
4. AO - shortcut to the "AT Override" panel.
5. MS - shortcut to the "MRT Conditions" panel.
6. MO - shortcut to the "MRT Override" panel.
7. MF - shortcut to the "MPF Specification" panel.
8. IG - shortcut to the "AT/MRT/MPF Ignore Specification" panel.

In the bottom screenshot you see that message IEF403I is used in Entry name +SA\_PREDEFINED\_MSGS Messages predefined by SA for AT build.



**Important:** You should not use any predefined IEF messages by specifying them in the MESSAGES/USER DATA policy as they are used by multiple resources and only one AT entry can be built per message

# Specifying a WTOR response in the policy

```

AOFGDYNYM          Message Processing      Line 00000001 Col 001 075
Command ==> _____ Scroll ==> CSR

Entry Type : Application      PolicyDB Name   : NEW_PDB
Entry Name  : TCPIP           Enterprise Name : SAPLEX

Line Commands: S/C (Cmd), R (Rep), K (Cod), U (Usr), A,M (AT, MRT, MPF)
               I, D (insert or delete lines), CN (Class Name)
Message ID field length . 16  (1 - 32)

Cmd Message ID      Description          C  R  K  U A M F
r_ EZAIN22I         Stop TCP/IP Server    1
***** Bottom of data *****
```

When EZAIN22I  
WTOR is issued for  
TCPIP, reply **C**

```

AOFGDYNY9      Reply Processing : EZAIN22I
Command ==> _____
```

| Cmd   | Ps>Select | Ct    | Reply Text |
|-------|-----------|-------|------------|
| _____ | _____     | _____ | <u>C</u>   |

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

Automatically respond to TCP/IP stopping WTOR (EZAIN22I) with a **C**

## Specifying a WTOR response in the policy

The examples on the slide have messages that are related to a start or stop of an Application. However, you can create code responses and commands that are issued at other times. You do not have to make them depend on an automation flag. They can run in all cases. To specify a response to a WTOR, select the MESSAGES/USER DATA policy item for the Application. In this case, the Application is TCPIP. Create an entry using the message ID and specify **REP** as the Action to provide responses to the WTOR.

Type **C** in the **Reply Text** field to reply to the EZAIN22I WTOR with a **C**. If the **Retry Count** field is zero (**0**) or not specified, the reply is issued only once. Otherwise, the reply is issued for the number of times specified, with a two-second delay between each retry.

Responses can be specific to a start or stop type. Enter the type in the **Pass/Selection** column. A null pass value (blank) means that it is issued for all types of start and stop. Therefore, **ALWAYS** is a possible selection, similar to **NORM**.

# MESSAGES/USER DATA for RV01

| AOFGDYNM  | Message Processing                   | Line 00000001 Col 001 075 |
|---|--------------------------------------|---------------------------|
| Command ==>   |                                      | Scroll ==> CSR            |
| Entry Type : Application  | PolicyDB Name : OPSDEMO              |                           |
| Entry Name : RV01   | Enterprise Name : OPSDEMO            |                           |
| Line Commands: S/C (Cmd), R (Rep), K (Cod), U (Usr), A,M (AT, MRT, MPF) |                                      |                           |
| I, D (insert or delete lines), CN (Class Name)                          |                                      |                           |
| Message ID field length . <u>16</u> (1 - 32)                            |                                      |                           |
| Cmd Message ID  | Description                          | C R K U A M F             |
| <u>IEF404I</u>  | <u>Test msg for example only</u>     | P                         |
| <u>VAPL2TEST</u>  | <u>Specifications for VAPL2TEST</u>  | 1 * * *                   |
| <u>VAPL21000A</u>   | <u>Specifications for VAPL21000A</u> | 3 *                       |
| <u>VAPL21010I</u>   | <u>Specifications for VAPL21010I</u> | *                         |
| <u>WTORS</u>  | <u>Specifications for WTORS</u>      | 2 P P P                   |
| ***** Bottom of data *****  |                                      |                           |

- Define VAPL2101 as the UP message
  - Enter **A** (AT), then **AS** (Automation Status) to display or set the status
  - Enter **C** (Cmd) to display or set the status **and** define commands to run
- Define Automation Table override for VAPL21000A message
  - Enter **A** (AT) to display the Automation Table and MRT statements, and **AO** (AT Override) to edit the automation table statement

## Set automation status based on message ID

| AOFGMAT1  |                          | AT Status Specification               |
|---|--------------------------|---------------------------------------|
| Command ==> _____                                     |                          |                                       |
| View or change specifications for message: VAPL21010I |                          |                                       |
| Application / MVC Component status                    |                          | Monitor health status                 |
| _ ACTIVE  | Starting                 | _ NORMAL Resource shows good results  |
| <u>S</u> UP   | Available                | _ WARNING Resource shows degradation  |
| _ HALTED  | Degraded                 | _ MINOR More severe than WARNING      |
| _ Terminating   | Received STOP command    | _ CRITICAL More severe than MINOR     |
| _ Terminated  | Terminated normally      | _ FATAL More severe than CRITICAL     |
| _ ABENDING  | Going to end abnormally  | _ UNKNOWN Health status not available |
| _ Abended   | Ended abnormally         | _ Check Evaluate health status        |
| _ BREAKING  | Non-recoverable abending |                                       |
| _ BROKEN  | Non-recoverable abend    | _ FAILED Monitor failed               |
| Capture specification                                 |                          | _ BROKEN Monitor finally failed       |
| _ Capture   | Capture Message          |                                       |
| DOM . . .   |                          |                                       |

By default, when VAPL2110I is received for RV01, set the status to AVAILABLE

### Set automation status based on message ID

Select the status by typing **S** in the Action column and press Enter. Press PF3 to save your change. In this example, the status of RV01 is set to UP when message VAPL21010I is received. You learn about setting the Health Status for a resource for monitor resources in Unit 5, Additional customization in [Figure , “Health Status aggregation,” on page 379.](#)

## Issue commands when status changes

```
AOFGDXNC      Command Processing : VAPL2101I          Line 00000001 Col 001 075
Command ===> _____           Scroll ===> CSR

Mixed case . . . NO (YES NO)

Cmd Ps>Select AutoFn/* Command Text
MYREXX P1 P2
_____
_____
_____
_____
_____
_____
_____
_____
***** Bottom of data *****
```

When VAPL2101I is received for RV01, issue the MYREXX EXEC command

### *Issue commands when status changes*

When message VAPL2101I is received for RV01, set the status to UP (based on the previous slide) and issue the MYREXX command. You do not have to create code for the Automation Table statements, because SA z/OS builds the message automation dynamically.

## Automation Table actions

| AOFGMADV   | Message Automation Overview | Scroll ==> CSR             |
|--|-----------------------------|----------------------------|
| Command ==>  |                             |                            |
| AC AT condition  | MS MRT action selection     | MF MPFLSTxx specifications |
| AS AT status   | MO MRT override             | IG Ignore for AT, MRT, MPF |
| AO AT override   |                             |                            |
| <b>Message ID : VAPL21000A</b>   |                             |                            |
| <b>Generated AT entry :</b>  |                             |                            |
| IF MSGID = 'VAPL21000A' THEN<br>EXEC(CMD('ISSUEACT')) ROUTE(ONE %AOFOPGSSOPER%); |                             |                            |
| <b>Generated MRT entry :</b>   |                             |                            |
| UPON ( MSGID = 'VAPL21000A' )<br>REVISE('Y' AUTOMATE)                            |                             |                            |
| <b>Generated MPF entry :</b>   |                             |                            |
| VAPL21000A   |                             |                            |

- In this example, the Automation Table and MRT statements are for the VAPL21000A message
- You can use an **AO** action to define an Automation Table override
  - Opens ISPF edit dialog for the selected Automation Table entry
  - Has automatic syntax checking

© Copyright IBM Corporation 2019

4-38

### Automation Table actions

You can override the default NetView Automation Table processing for a specific message ID for a specific Application using the **AO** action. An ISPF edit dialog panel is displayed. You can use the ISPF edit dialog panel to override predefined NetView Automation Table (AT) entries for an Entry Name (Application) and message ID. If you override the entry, an asterisk is displayed in the A (Automation) column of the Message Processing panel. No syntax checks are made for the AT entry within the customization dialog. Lowercase characters are converted to uppercase. Labels defined for certain SA z/OS default AT entries are not used if the AT entry is overridden. If message automation is neither predefined by SA z/OS nor defined by the user, then no AT entry is generated for the resource.



**Note:** Use caution when coding Automation Table or message revision table statements. Errors prevent refreshing the automation configuration.

## Message override automatic syntax check

```
AOFGMADV          Message Automation Overview
AOFGM054          AT Syntax Check Result
CNM521E INVALID ACTION NAME "EXEK"
DSI417I #0000002 : IF MSGID = 'VAPL21010I' THEN EXEK(CMD('ACTIVMSG UP=YES')
ROUTE(ONE * AUTGSS AUTSYS AUTBASE AUTO1));
DW0525I TEST OF NETVIEW AUTOMATION FILE WAS UNSUCCESSFUL
Press ENTER to keep override as is.
Press END or CANCEL to continue editing.
```

- You can enable automatic syntax checking for the AO action
- The checking is done in the NetView agent address space when you press F3 to save the change; it uses the command receiver facility

© Copyright IBM Corporation 2019

4-39

### Message override automatic syntax check

Message override syntax checking is enabled by default. You must explicitly disable it to turn it off. The syntax checking can be disabled using the **Message** selection of the Settings Menu. For example, to enable checking of AT and MRT overrides requiring NetView, you have the following options:

AT / MRT syntax check . . . NO (YES NO)

## Automation flags

| AOFGAFLG   |                | Automation Flag Processing |       |              |       |
|--|----------------|----------------------------|-------|--------------|-------|
| Command ==>  |                |                            |       |              |       |
| Entry Type : Application   |                | PolicyDB Name : OPSU3EX1   |       |              |       |
| Entry Name : RV01  |                | Enterprise Name : OPSU3EX1 |       |              |       |
| Resource : RV01  |                |                            |       |              |       |
| Line Commands: Exi (Exits), Dis (Disable Times), CN (Class Name) |                |                            |       |              |       |
| Automation Level: YES, NO, LOG, EXITS                            |                |                            |       |              |       |
| Cmd  | Flag           | Auto                       | Exits | DisableTimes |       |
|  | Automation (A) | _____                      | _____ | _____        | _____ |
|  | Initstart (I)  | _____                      | _____ | _____        | _____ |
|  | Start (S)      | _____                      | _____ | _____        | _____ |
|  | Recovery (R)   | NO                         | _____ | _____        | _____ |
|  | Terminate (T)  | _____                      | _____ | _____        | _____ |
|  | Restart (RS)   | _____                      | _____ | _____        | _____ |

- Automation flag (A): Global automation flag for the resource. If NO, all flags are NO
- InitStart flag (I): Checked after IPL only, when application has a true DOWN status
- Start flag (S): Checked for automation after STARTUP command issued and for POSTSTART commands
- Recovery flag (R): Controls automation when application is UP or DOWN
- Terminate flag (T): Controls all shutdown commands and automation during shutdown
- Restart flag (RS): Tested in all other DOWN states

© Copyright IBM Corporation 2019

4-40

### Automation flags

Use the AUTOMATION FLAGS policy item to set any specific agent flags for an Application. For performance reasons, do not specify flags at the resource level unless you want to override higher-level (for example, system-level) settings. Set the flags to **YES** or **NO** as needed. If you set the *Automation flag* to NO, all other agent flags for that Application are effectively set to NO.

If you want to enable automation but specify times when it is disabled, you can enter **YES** in the last field. Another panel is displayed where you can specify one or more time ranges and the times when automation is disabled. For example, you can set times of each week when some automation flags are automatically set to NO. The *Recovery flag* indicates whether automated recovery of the resource can occur. Automated recovery of the resource is an action on a message that is not for the initialization or for the ending of the resource. This recovery involves message automation (no automation manager is involved). You cannot set the manager flags in the policy definitions. Automation flags can also be used for minor resources of an Application. Data that is inherited from a class definition is now displayed for the automation flags application policy items. The Automation flags policy item shows inherited flag exits and disable times.

## Threshold processing

- Before a major resource is restarted or a minor resource is recovered or automated, specific or generic failure thresholds are checked
- There are three thresholds: Infrequent, frequent, and critical
- Actions are taken when a threshold is met

| Threshold         | Actions taken  |
|-------------------|--|
| <b>INFRequent</b> | Issue message<br>Attempt restart. Recover or automate  |
| <b>FREQuent</b>   | Issue more serious message<br>Attempt restart. Recover or automate                               |
| <b>CRITical</b>   | Issue critical message<br>Do NOT attempt restart; status is BROKEN<br>Do NOT recover or automate |

© Copyright IBM Corporation 2019

4-41

### Threshold processing

You can use thresholds to alert operators when too many recoverable restarts have occurred and to disable restart processing altogether. You can use specific values or use default values. Typically, you only have to define threshold values that are different from the defaults that are specified at the MVS Component, System Defaults, or Application Defaults levels. For performance reasons, do not specify thresholds at the resource level unless you want to override higher-level settings; for example, system-level default settings.

A *threshold* is a counter that tracks the number of post failure restarts or automation actions that a resource experienced during a specified period.

The automation product uses three thresholds when automating the restart of resources or when automating or recovering minor resources. These thresholds can be used to identify the frequency of failures or automation actions for an automated resource.

The primary use of error thresholds is to track subsystem abends and ensure that the abend and restart cycle does not become an infinite loop, but they may also be customized for other uses.

Automation provides three threshold levels for which you can specify criteria:

- Infrequent (INFR)
- Frequent (FREQ)
- Critical (CRIT)

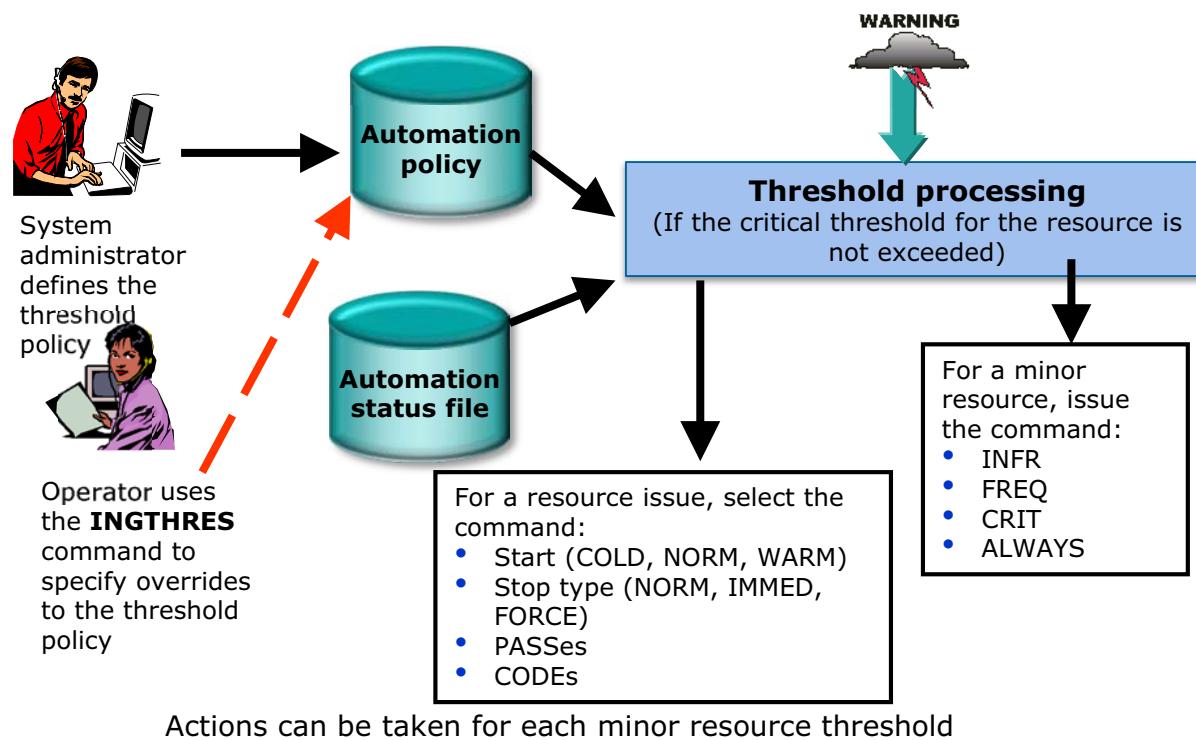
In all cases, when a threshold is met or exceeded, a message is issued. This message is written to the NetView log and can alert operators.

If the application has not exceeded its critical threshold, the automation product posts the application to RESTART and attempts to restart it. Restart automation is stopped when a defined CRIT (critical) threshold is met.

When the critical threshold is met:

- Automatic restart of the resource is halted
- Its agent status is set to BROKEN
- Its manager compound status is set to PROBLEM
- Notify operators can be alerted.
- Operator intervention is required for the automation product to restart the resource.

## Threshold processing and minor resources



© Copyright IBM Corporation 2019

3-42

### Threshold processing

Minor resource thresholds allow for the automation product administrators to define automated actions as reactions to each threshold, or a default action for all thresholds.

Automation tests minor resource thresholds for commands that are issued for a resource; for example, from PASSes or CODEs.

LOG, SYSLOG, LOGREC, MVSDUMP and SMFDUMP are now defined as minor resources of MVSESA with thresholds and actions. You can define actions for INFrequent, FREquent, and CRITical minor resource thresholds. You can also define actions for automation to take if no minor resource threshold is exceeded (ALWAYS). No actions are taken if the resource exceeds its critical threshold.

If you want to display all defined thresholds or add, change, or delete threshold settings for a particular resource, use the INGTHRES command. This displays the related Command Dialogs panel.

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

If you want to check the error counts for resources on a specific system, issue a DISPERRS command to this target system. SA z/OS displays the DISPERRS Command Dialogs panel with a list of all the errors that have been recorded for resources on the target system. Although DISPERRS displays the errors that have occurred for all of a system's resources, DISPASF lets you see detailed information about errors for a resource on a specific system.

## Major and minor resources

- Used for agent automation flags and threshold processing
- Major resources are:
  - Applications
  - Generic settings
    - SUBSYSTEM: all applications
    - MVSESA: all MVS components
    - System DEFAULTS
- Minor resources belong to a major resource and can be pre-defined or dynamically created, representing for example:
  - Messages
  - Transactions
  - Single MVS component under MVSESA:
    - LOG, SYSLOG, LOGREC, MVSDUMP and SMFDUMP
- Minor resources notation:
  - Major\_resource\_name.minor\_resource\_name
  - Example: MVSESA.SMFDUMP, VTAM.IST020I

© Copyright IBM Corporation 2019

4-43

### Major and minor resources

Major and minor resources are used for agent automation flags and threshold processing.

**Major resources** are:

- Applications
- Generic settings
- SUBSYSTEM: all applications
- MVSESA: all MVS components
- System DEFAULTS

You can also define your own resources, called **minor resources**. Minor resources belong to a major resource and can be pre-defined in the policy or dynamically created by the operator, representing for example:

- Messages
- Transactions
- Single MVS component under MVSESA:  
LOG, SYSLOG, LOGREC, MVSDUMP, and SMFDUMP

Minor resources notation:

- Major\_resource\_name.minor\_resource\_name
- Example: MVSESA.SMFDUMP, VTAM.IST020I, CICS.TRANS

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

Minor resources can be pre-defined in the policy using the Minor Resource Definitions panel which is displayed if you select the MINOR RESOURCES policy item in the Policy Selection panel for the MVS Component or applications entry types.

## Example: Minor resource threshold for RV05

- RV05 is an application
- RV05.VAPL25020I is the minor resource  
RV05 issues the VAPL25020I message
- MINOR RESOURCES thresholds:  
Define RV05.VAPL25020I minor resource with INFRequent, FREQuent, and CRITical thresholds
- MESSAGES/USER DATA: Define message VAPL25020I with actions for ALWAYS, CRIT, FREQ, and INFR

### Example: Minor resource threshold for RV05

On the next several slides, you see an example of how to define thresholds. These thresholds are for a minor resource, that is a message (VAPL25020I) issued by an Application (RV05). The MINOR RESOURCE THRES policy item defines the INFR, FREQ, and CRIT thresholds. The MESSAGES/USER DATA policy item defines the actions to take for each threshold for message VAPL25020I.

## Example: RV05 MINOR RESOURCES thresholds

| AOFGDYNA                               | Thresholds Definition                      | Line 00000001  |
|--|--|----------------|
| Command ==> _____                      |  | Scroll ==> CSR |
| Entry Type : Application               | PolicyDB Name : OPSDEMO                    |                |
| Entry Name : RV05                      | Enterprise Name : OPSDEMO                  |                |
| Resource : RV05.VAPL25020I             |  |                |
| Critical Number . . . . . 4            | (1 to 50)                                  |                |
| Critical Interval . . . . . 00:10:00   | (hh:mm:ss or hhmmss, 00:00:01 to 24:00:00) |                |
| Frequent Number . . . . . 3            | (1 to 50)                                  |                |
| Frequent Interval . . . . . 00:10:00   | (hh:mm:ss or hhmmss, 00:00:01 to 24:00:00) |                |
| Infrequent Number . . . . . 2          | (1 to 50)                                  |                |
| Infrequent Interval . . . . . 00:10:00 | (hh:mm:ss or hhmmss, 00:00:01 to 24:00:00) |                |
| ***** Bottom of data *****             |  |                |

### MINOR RESOURCES thresholds policy item for RV05.VAPL25020I

- INFR: **2** failures in 10 minutes
- FREQ: **3** failures in 10 minutes
- CRIT: **4** failures in 10 minutes

© Copyright IBM Corporation 2019

4-45

#### Example: RV05 MINOR RESOURCES thresholds

Thresholds can be defined for Applications or Application minor resources. Use the THRESHOLDS policy item for the Application. If you specify one of the threshold levels (critical, frequent, or infrequent), you must specify all three levels. Use the MINOR RESOURCES policy item for Application minor resources. On this slide, you see an example of minor resource thresholds for RV05.VAPL25020I. RV05 is the Application, and the message, VAPL25020I, is the minor resource.

Other minor resource thresholds can be MVSESA.MVSDUMP or MVSESA.SMFDUMP. Both thresholds are defined under the MVS Component (MVC, option 33) policy. You can define system-wide threshold defaults under the System Defaults (SDF, option 35) policy. You can define application threshold defaults under the Application Defaults (ADF, option 36) policy.

## Example: RV05 MESSAGES/USER DATA policy

## MESSAGES/USER DATA policy item for RV05:

- Four commands are defined for the message (VAPL25020I) represented by the minor resource (RV05.VAPL25020I)
    - Enter **C** (Cmd) to see command definitions

© Copyright IBM Corporation 2019

4-46

*Example: RV05 MESSAGES/USER DATA policy*

On this slide, you see the MESSAGES/USER DATA policy item for Application RV05. Message VAPL25020I is defined with four commands. This is done by using line command C to select the message and then define the four related commands as shown on the next slide.

## Example: VAPL25020I commands

```
A0FGDYN9      Command Processing : VAPL25020I          Line 00000001 Col 001 075
Command ==> _____ Scroll ==> PAGE

Cmd Ps>Select AutoFn/* Command Text
____ ALWAYS _____ MSG ALL ++++++ ALWAYS ACTION TAKEN ++++++
____ CRIT _____ MSG NETOP1,*** CRITICAL THRESHOLD EXCEEDED FOR RV05.VAP
S INFR _____ MSG NETOP1,*** INFREQUENT THRESHOLD EXCEEDED FOR RV05.V
____ FREQ _____ MSG NETOP1,*** FREQUENT THRESHOLD EXCEEDED FOR RV05.VAP
***** Bottom of data *****
```

Four commands for VAPL25020I message are:

INFR, FREQ, CRIT, and ALWAYS

```
A0FGDYN7      Command Processing : line 2
Command ==> _____
```

This panel displays the complete input fields of a single data line.  
Press END to save changes or CANCEL to discard changes.

|                             |   |
|-----------------------------|---|
| Pass/Selection . . . . .    | <u>CRIT</u>   |
| Auto Function/'*' . . . . . | <u> </u>  |
| Command Text . . . . .      | <u>MSG NETOP1,*** CRITICAL THRESHOLD EXCEEDED FOR RV05.VAPL25020I ***</u> |

© Copyright IBM Corporation 2019

4-47

*Example: VAPL25020I commands*

Four commands are defined for message VAPL25020I. The pass count field contains these actions:

- ALWAYS: Take this action when no threshold for the minor resource is met.
- CRIT: Take this action when the critical threshold for the minor resource is met.
- INFR: Take this action when the infrequent threshold for the minor resource is met.
- FREQ: Take this action when the frequent threshold for the minor resource is met.

## Policy inheritance for applications

- For many attributes, the policy value is derived from the following definitions and values:
  - Specific Application definition (option 6)
  - Application Class definition (option 6)
  - Application defaults definition (option 36)
  - System defaults definition (option 35)
  - Hardcoded default values.
- Inherited data is now displayed for the following application policy items:
  - Minor resources
    - Threshold and Flag columns for minor resources
  - Automation flags
    - Inherited flag exits and disable times

© Copyright IBM Corporation 2019

4-48

### Policy inheritance for Applications

For many Application specifications, you can use default values. When you use default values, you can omit many of the specifications when you create Application definitions. Different systems can use different values, for example, *Restart Option* or *Shutdown Delay*. Also, if you want to change a default value, you make only one change instead of many changes. The policy search begins with resource-specific definitions and ends when the first value is found.

- If specified in the policy for the resource, that value is used.
- If specified in the CLASS definition, that value is used.
- If specified in the Application DEFAULTS policy object, that value is used.
- If specified in the SYSTEM DEFAULTS policy object, that value is used.
- In some cases, a default value hard-coded in SA z/OS is used.

You can specify system-level defaults with option **35** (SDF) on the policy Entry Selection panel. You can specify application-level defaults with option **36** (ADF). Policy inheritance works similarly for APGs, MTRs, and MVC. MTRs inherit NY from SDF; MVC inherits from MDF and SDF.

## Minor resource inheritance example

- APL policy MINOR RESOURCES policy item with inherited data

| Minor Resource Definitions                                |                             | Line 00000001 col 001 075 |   |   |   |   |   |    |  |  |  |
|---|-----------------------------|---------------------------|---|---|---|---|---|----|--|--|--|
| Command ==>   |                             | Scroll ==> CSR_           |   |   |   |   |   |    |  |  |  |
| Entry Type : Application                                  |                             | PolicyDB Name : DEMO34    |   |   |   |   |   |    |  |  |  |
| Entry Name : DEMOAPPL                                     |                             | Enterprise Name : DEMO34  |   |   |   |   |   |    |  |  |  |
| Line Commands: F (Flg), s (Thr), T (Thr), CN (Class Name) |                             |                           |   |   |   |   |   |    |  |  |  |
| I, D (insert or delete lines)                             |                             |                           |   |   |   |   |   |    |  |  |  |
| - Automation Flags -                                      |                             |                           |   |   |   |   |   |    |  |  |  |
| Cmd   | Minor Resource Name         | Thres                     | A | I | S | R | T | RS |  |  |  |
|   | MINRES-CLASS-LEVEL2         | *                         | * | * | * |   |   | *  |  |  |  |
|   | MINRES-PARTIALLY-INHERITED1 | *                         |   | * |   |   | * |    |  |  |  |
|   | MINRES-PARTIALLY-INHERITED2 | *                         |   | * | * |   |   | *  |  |  |  |
|   | MINRES-LOCAL                | *                         | * |   | * |   |   | *  |  |  |  |
|   | MINRES-WITHOUT-DATA         |                           |   |   |   |   |   |    |  |  |  |
|   |                             |                           |   |   |   |   |   |    |  |  |  |
|   |                             |                           |   |   |   |   |   |    |  |  |  |
|   |                             |                           |   |   |   |   |   |    |  |  |  |

### Minor resource inheritance example

The example screen capture on this slide shows the minor resource definitions panel in the application policy of an application called DEMOAPPL. The minor resources are listed in a tabular form on the bottom portion of the panel. The minor resource name column contains the name of the minor resource and the columns to the right of the name contain indications for the presence of thresholds and automation flags. The columns are either blank or they have an asterisk to indicate the presence of that flag for the specific minor resource listed. The color of a minor resource name or of an asterisk indicates whether or not it was inherited from a class. A red minor resource name or red asterisk is for inherited data. A black asterisk is for data that was defined specifically for this application. Minor resources with black names can have a mix of inherited and non-inherited data, and are referred to as partially inherited minor resources. The colors options called Inherited Data Color Selection are under the settings menu and can be customized by the user. Red was used in this example to show inherited data.

In the example shown on the slides, the first minor resource listed is called **minres-class-level2**. This minor resource is 100% inherited because both the name and all of the asterisks in data columns are red. The second resource listed, **minres-partially-inherited1**, is only partially inherited because the name and some of the flag columns are black. In this second minor resource listed, only the threshold data is red; therefore, only the threshold data is inherited.

# Application default specifications

```

Policy Selection                               Row 1 to 7 of 7
Command ==> _____                           Scroll ==> CSR

Entry Type : Application Defaults   PolicyDB Name : NEW_PDB
Entry Name  : APPL_DEFAULTS           Enterprise Name : SAPLEX

Action      Policy Name          Policy Description
DESCRIPTION  Enter description
APPLICATION INFO Define application information defaults
AUTOMATION FLAGS Define application automation flag defaults

Application Information Defaults
Command ==> _____
Entry Type : Application Defaults   PolicyDB Name : NEW_PDB
**Entry Name : APPL_DEFAULTS           Enterprise Name : SAPLEX

Captured Messages Limit. . . . . (0 to 999)
Monitor Routine . . . . . (name NONE)
Startup information:
  Restart after IPL . . . . . (START NOSTART)
  Start Delay . . . . . 00:02:00 (time for "UP" status checks, hh:mm:ss)
  Start Cycles . . . . . (start delay checks, 0 to 99)
  Restart Option . . . . . ALWAYS (ALWAYS ABENDONLY NEVER)
Shutdown information:
  Shutdown Pass Interval . . 00:01:00 (hh:mm:ss)
  Cleanup Delay . . . . . 00:00:12 (hh:mm:ss)

```

© Copyright IBM Corporation 2019

4-50

## Application default specifications

On this slide, you see an example of the Application Defaults policy. This is option 37 on the Entry Type Selection panel. You can define default values for Applications for recovering, for example, the resource or thresholds. To see the defaults for all Applications, select the APPLICATION INFO policy item and press Enter.

On the Application Information Defaults panel, you can set system-wide defaults for Application definitions. For example, you might want to use these settings:

- Change the **Restart Option** field to ABENDONLY to only recover Applications when they fail (as opposed to when an operator stops them).
- Modify the default **Start Timeout**.

These fields are also available at the Application level. By providing them here as defaults, you have fewer definitions and more consistent Application automation.

# Lesson 2. Automation configuration file

## Lesson 2: Automation configuration file

- The SA z/OS automation configuration file is a single preallocated partitioned data set: PDS or PDSE
- The build process creates policy objects that represent all resources for all systems in the enterprise
- The file contains the following items:
  - Automation Manager configuration files
  - Automation Agent configuration files
  - Automation Table members
  - Message revision table (MRT)
  - MPF list member

### What this lesson is about

This lesson describes the System Automation for z/OS automation configuration file and how it is used.

### Objectives

After completing this lesson, you should be able to describe the System Automation for z/OS automation configuration file and how it is used.

### References

SC34-2717 *System Automation for z/OS Version 4 Release 1 Defining Automation Policy*

The SA z/OS automation configuration file is a partitioned data set (PDS or PDSE). Many files are created when the policy is built. The PDS contains a main mapping member, AOFACFMP. This mapping member points or maps to the main sysplex AM members (HSAZxxx) and AA members (ACFZxxx) for all defined systems. These automation manager and automation agent members

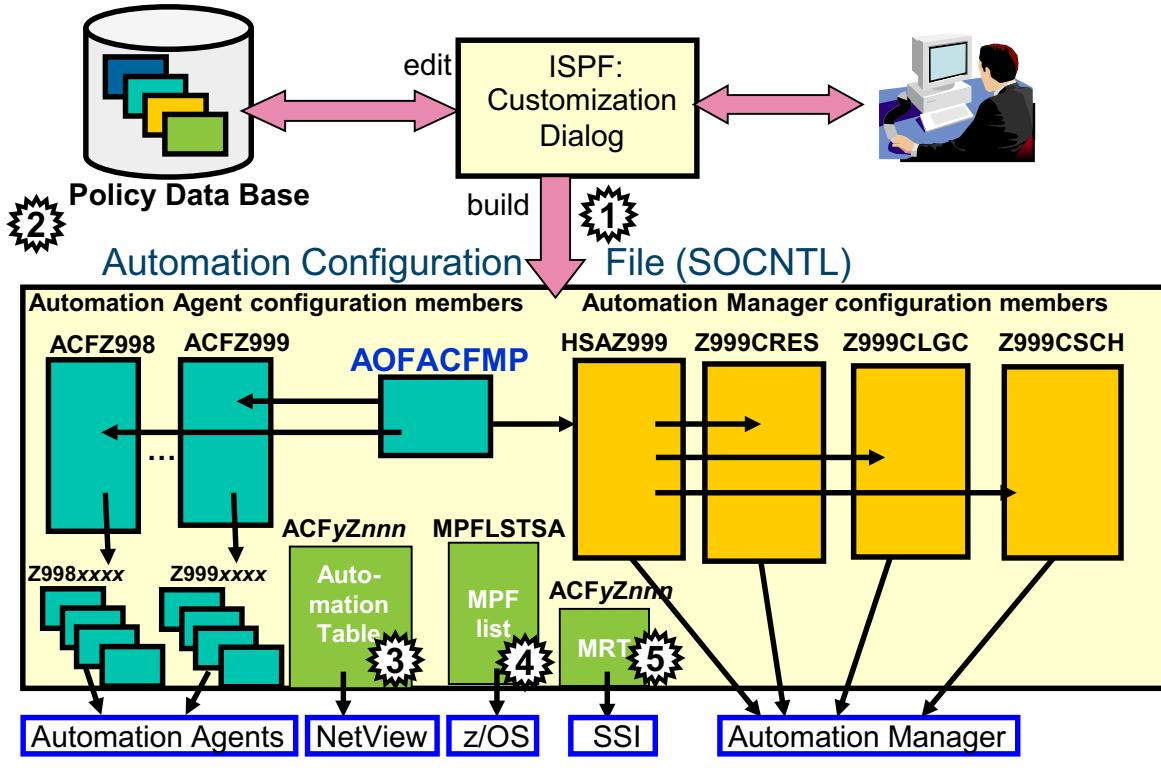
point to or include the remaining configuration members. Their names are eight characters long and start with a Z.

NetView Automation Table members are also produced with the following naming convention:

- ACFM~~Z~~xxx: if AT scope = SYSTEM
- ACFE~~Z~~xxx: if AT scope = ENTERPRISE
- ACFX~~Z~~xxx: if AT scope = SYSPLEX

A z/OS MPF list member (MPFLSTSA) is also produced.

## Customization Dialog: Process



animated

© Copyright IBM Corporation 2019

4-52

### Customization dialog: Process

Using the customization dialog, the system administrator creates and updates entries in a Policy Data Base (PDB). The build process takes the entries in the PDB and creates the following items:

- Automation control files (ACFs), one for each system that is defined in the PDB; ACFZ998 and ACFZ999 are the system include members in this example
- Automation manager configuration files, one set for each SAplex in the enterprise; HSAZ999 in this example
- Automation Table, ACFyZnnn; for example, ACFMZ999
- MPF list, MPFLSTSA
- NetView message revision table (MRT), ACFyZnnn, one for each system

The mapping member, AOFACFMP, identifies which ACF relates to which system, and which set of manager configuration files are used on which sysplex. All of these files are created as members in a single data set. This output data set, or a copy of it, can then be used by the managers and agents on each of the SAplexes defined in the PDB.

## Automation configuration file format

- AOFACFMP, the main member, contains pointers to the following members:
  - AM main members, HSAZyyy
  - AA members, ACFZxxx
- Each member points to or includes all other required AM and AA configuration members
- The main member also contains pointers to the following items:
  - Automation Table (AT) members
    - ACFMZxxx: if AT scope = SYSTEM
    - ACFEZ999: if AT scope = ENTERPRISE
    - ACFXZxxx: if AT scope = SYSPLEX
  - NetView message revision table (MRT): ACFTZxxx
  - MPF list member: MPFLSTSA

### Automation configuration file format

You define the Automation Table scope when you use an *edit* action against the PDB. The scope can be one of the following items:

- **NONE:** No Automation Table is built.
- **ENTERPRISE:** One Automation Table is built for the entire PDB.
- **SYSPLEX:** One Automation Table is built per sysplex group. No Automation Table is built for systems that are not linked to a sysplex group. A standalone system must be exclusively linked to one sysplex group.
- **SYSTEM:** One Automation Table is built per system (default).

## Automation configuration file members example

| SAZOS.ACFS.OPSDEMO |          |        |      |         |         | Row 0000001 of 0000087 | Scroll ==> CSR |
|--------------------|----------|--------|------|---------|---------|------------------------|----------------|
| Command            | Name     | Prompt | Size | Created | Changed | ID                     |                |
|                    | \$BLDRPT |        |      |         |         |                        |                |
|                    | ACFMZ998 |        |      |         |         |                        |                |
|                    | ACFMZ999 |        |      |         |         |                        |                |
|                    | ACFTZ998 |        |      |         |         |                        |                |
|                    | ACFTZ999 |        |      |         |         |                        |                |
|                    | ACFZ998  |        |      |         |         |                        |                |
|                    | ACFZ999  |        |      |         |         |                        |                |
|                    | AOFACFMP |        |      |         |         |                        |                |
|                    | HSAZ999  |        |      |         |         |                        |                |
|                    | MPFLSTSA |        |      |         |         |                        |                |
|                    | Z98LAAPL |        |      |         |         |                        |                |
|                    | Z98MAAPL |        |      |         |         |                        |                |
|                    | Z98NAAPL |        |      |         |         |                        |                |

**AOFACFMP:** Main ACF map member

**ACFMZxxx:** AT members

**HSAZyyy:** AM members

**ACFZxxx:** AA members

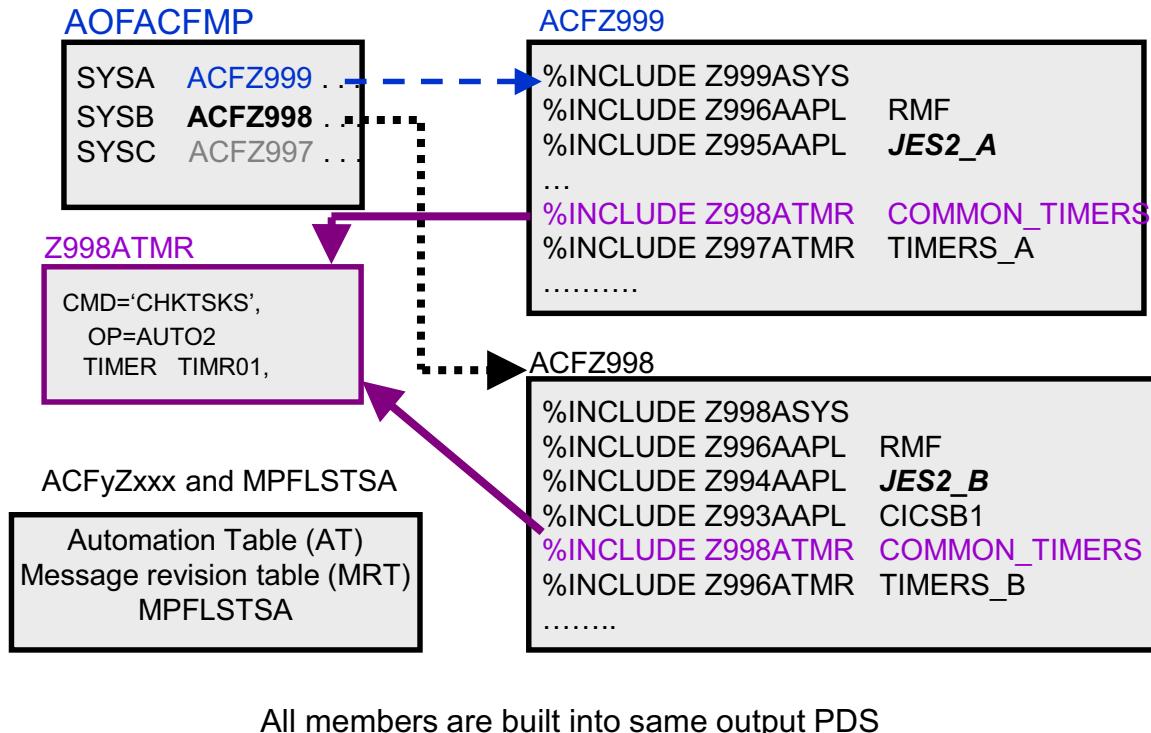
**ACFTZxxx:** MRT member

**MPFLSTSA:** MPF list member

### Automation configuration file members example

On this slide, you see a sample member list from an automation configuration file PDS, SAZOS.ACFS.OPSDEMO. The contents of these members is used internally by System Automation for z/OS and must never be changed or modified manually.

## Automation agent files



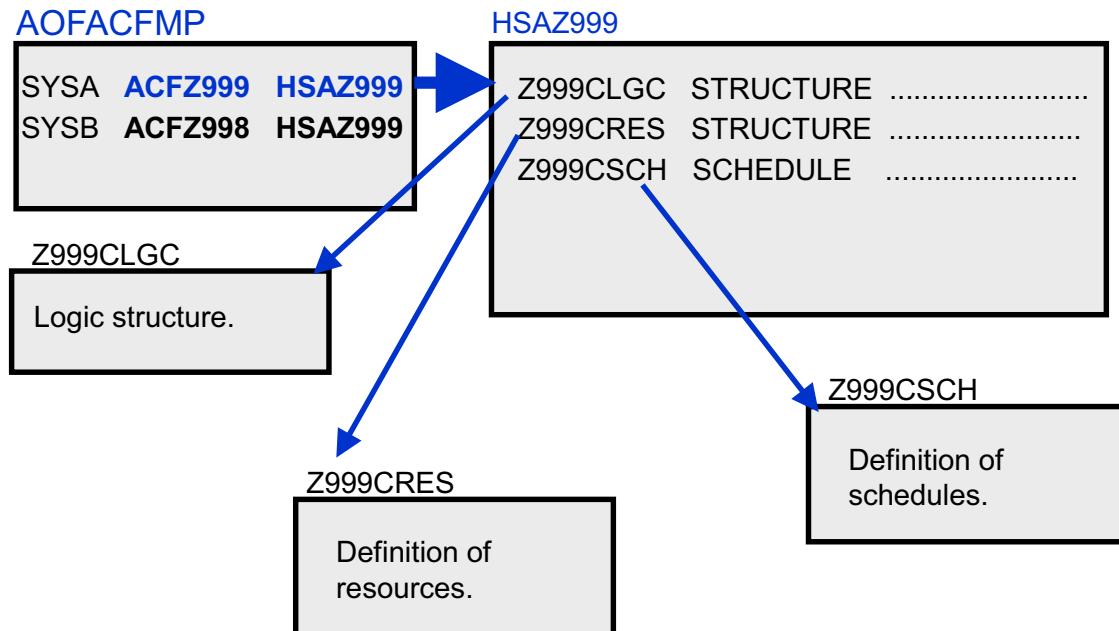
© Copyright IBM Corporation 2019

4-55

### Automation agent files

A full build creates an ACF for each system that is defined in the Policy Data Base. The *ACF map* member (AOFACFMP) shows which ACF to use with each system. In this figure, you see a simplified format of ACFs that are built. You also see how they are mapped by the ACF map member, AOFACFMP. The build created a separate member for the policy object called COMMON\_TIMERS. Because this is selected in both SYSA and SYSB, a %INCLUDE statement for its ACF fragment is in both ACFs. Similar common fragments are shown for RMF. Because there are separate definitions for JES2, there are two fragments built; each one is included (%INCLUDE) only in the appropriate ACF.

## Automation manager files



All members are built into the same output data set

### Automation manager files

Within the same full build, the following four members are created for the sysplex:

- An AM configuration file member for the sysplex. This member, HSAZ999 in the example, points to the other three members.
- A logic member, Z999CLGC.
- A resources member, Z999CRES, which contains information about all resources defined for the sysplex group.
- A schedule member, Z999CSCH, which contains information about all schedules defined for the sysplex group.

For each system, the ACF map also has a value that indicates which sysplex main member is used by any AM running on that sysplex. Thus, all systems in the sysplex group have a pointer to the same main member.

## Map member: AOFACFMP

```
BROWSE      SAZOS.ACFS.OPSDEMO(AOFACFMP)          Line 00000000000 Col 001 080
Command ===> [ ]                                     Scroll ===> PAGE
***** Top of Data *****
* AOFACFMP member SA41
*
* Generated at 15:57:17 on 2018/12/17
*
* Policy Database: OPSDEMO
* Enterprise   : OPSDEMO
* Data Set Name : 'SAZOS.PDB.OPSDEMO'
* Configuration : NORMAL
* LGCLEVEL     : 20181217155707
* ATSCOPE       : SYSTEM
*
* SYSNAME    ACFNAME  CONFNAME SYSTEM NAME
*-----*
MVSA        ACFZ999  HSAZ999  MVSA
MVSB        ACFZ998  HSAZ999  MVS
*-----*
*** End of Data, 2 system(s) processed ***
----- Begin of Signature -----
*4FC7CA819849BEA169804F8C0D5DD6219F2F
*5B340CB7E1EFE92AF10E495321F69457D824
----- End of Signature -----
***** Bottom of Data *****
```

### Map member: AOFACFMP

On this slide, you see an example of an ACF map member. In this case, the map member in this example shows two systems: MVSA and MVS. Each system has a unique ACF name. For example, the ACF for MVSA is member ACFZ999. Systems in the same sysplex group show the same AM configuration file name, HSAZ999. Because the automation manager manages a sysplex, there is only one AM configuration file. This member is consulted whenever an agent or manager initializes, and it indicates which members to use.

## Automation agent main member example

```
BROWSE   SAZOS.ACFS.OPSDemo(ACFZ999)          Line 0000000000 Col 001 080
Command ==>_                                Scroll ==> PAGE
***** Top of Data *****
* Title : Automation Control File
* Desc  : System Identifier : MVSA
* Built : INGC102 18/12/17 15:57:17 MUSA SA41
* ACLEVEL: 20181217155707
* CFGLEVEL: 20181217155707FF01F9672827
* RUNMODE: NO_RVAPPL YES_RVAPPL
* MATLIST: INGMSG01
* MATLIST:
* MATLIST:
*-----
*Action Member    Entry name      Status Last modified Subsystem C
%INCLUDE Z999ASYS MVSA          BLT    20180824165601 MVSA
%INCLUDE Z999ASDF SYSTEM_DEFAULTS BLT    20180824134437
%INCLUDE Z999AXDF SYSPLEX_DEFAULTS BLT    20180824134437
%INCLUDE Z999AADF APPL_DEFAULTS  BLT    20180824134429
%INCLUDE Z98RAAPL RV_MSTR_CLASS  BLT    20180824135854 RV_MSTR_CLS *
%INCLUDE Z98QAAPL RVCLASS        BLT    20180824135858 RVCLASS   *
%INCLUDE Z999AAPL C_AM          BLT    20180824134428 C_AM     *
%INCLUDE Z99KAAPL C_RUNMODE     BLT    20180826203321 C_RUNMODE *
%INCLUDE Z99AAPL C_USS_PORT    BLT    20180826204122 C_USS_PORT *
%INCLUDE Z998AAPL C_APPL        BLT    20180824134428 C_APPL    *
%INCLUDE Z996AAPL C_JES2        BLT    20180824134428 C_JES2    *
%INCLUDE Z990AAPL C_APPL_OUTPUTS BLT    20180824134430

```

**Automation Table members**

**Timestamp**

**Included fragments:**  
Policy object type is the last three characters of the member name

**Build status**

© Copyright IBM Corporation 2019

4-58

### Automation agent main member example

On this slide, you see an example of an agent ACF main member, ACFZ999 for system MVSA. The member is made up of %INCLUDE and comment statements. The agent processes some of the comment statements in the header. The CFGLEVEL value is the token that is used to ensure that consistent files are used by the manager and its agents. The **Built** field indicates when the member was last modified. This information is used to determine whether the agent can do a WARM start from the cached ACF information. The operator sees this value in the output from an **ACF STATUS** command.

Each %INCLUDE statement specifies a fragment of the ACF. Each fragment is built from a single policy object that was selected for that system. You can see what the objects are from the Entry Names, which reflect the names used in the PDB. The names of the ACF fragments have a three-character suffix that identifies the type of policy object; for example, APL for Application.

For each fragment, you also have the following information:

- A date and time stamp, used when the ACF is refreshed so that only updated fragments are refreshed; indicates the time that the PDB was last updated for the policy object
- An indicator that shows whether this is a class definition
- The subsystem name (for Applications)

# Lesson 3. Build automation configuration files

## Lesson 3: Build automation configuration files

- Complete enterprise
  - Build and update configuration files for all systems and all sysplexes.
- Single sysplex or stand-alone system
  - Build and update configuration files for all systems of one sysplex
  - Build and update configuration files for one stand-alone system (a sysplex of one system).
- Single entry type or name
  - Build selected policy items
  - Update ACF fragment onl.
  - Often not possible because of required AA or AM changes
  - Use for AA changes only

### What this lesson is about

This lesson describes how to build the automation configuration file using the System Automation for z/OS ISPF customization panels.

### Objectives

After completing this lesson, you should be able to build the automation configuration file using the System Automation for z/OS ISPF customization panels.

### References

*SC34-2717 System Automation for z/OS Version 4  
Release 1 Defining Automation Policy*

During the build process, the policy definitions that you created are used to create the following files for all systems and sysplexes that you defined. All files are written to the same output data set.

- Configuration files for the automation agent
- Configuration files for the automation manager
- Automation Tables (ATs)
- Message revision table (MRT)
- MPFLSTSA member

You can start the build process by using the panel options (2.1) or by entering **F** as the action for a policy entry to build only that policy entry. You have the following options for a build:

- Build a complete set of configuration files for the installation (option **1**)
- Build configuration files for a single sysplex group (option **2**)
- Build configuration files for a single stand-alone system, that is, a system that is not defined to be part of a sysplex group with other systems (option **2**)

A stand-alone system that is defined in the PDB might be part of a sysplex. You can define systems to be within their own sysplex subgroup. In this case, you have a sysplex (group) entry with one system in it. If you use option 2 to build files for a single system or sysplex, you can enter a question mark (?) in the **Sysplex/System Name** field to see a list of possible choices. To ensure the integrity of your automation configuration data set, back up your PDB and use generation data groups (GDGs) for the build output. The batch build sample job, INGEBBLD, contains a step for copying the PDB to a GDG.

# Invoking a build for the first time

```

Configuration Build          Full build required
Option ==> [ ]  

1 Build a complete enterprise  

2 Build sysplex group or stand alone system  

   Sysplex / System name. . . . . (*)  

3 Build entry type or entry name  

   Entry Type. . . . . (*)  

   Entry Name. . . . . (*)  

4 View build report  

5 Delete unused build output members  

Build options:  

  Output Data Set . . . . . 'SAZOS.ACFS.NEWPDB'  

  Mode. . . . . . . . . ONLINE      (ONLINE BATCH)  

  Type. . . . . . . . . MODIFIED    (MODIFIED ALL) (ALL)  

  Configuration . . . . . NORMAL     (NORMAL ALTERNATE)  

Job statement information: (used for BATCH build)  

//AOFBUILD JOB  

//  

//*
//*

```

The first build generates a **Full build required** message  
 You must specify **TYPE=ALL**

## Invoking a build for the first time

Access the Build Parameters panel by entering **=2.1** from the main customization dialog panel. To start a build, enter the desired build option in the **Option** field. Values entered on the Build Parameters panel are saved across sessions.

When you start a build, you build a consistent set of configuration files by using the information from a single PDB. The PDB is *locked* during build.

If you try to use option **2** with an empty output data set, the **Full build required** message is displayed in the top right of the screen (see slide). You must use option **1** to build a complete enterprise when you start a build for the first time.

The files that are created for a build request are stored as members of the same partitioned data set, which is defined in the **Output Data Set** field. This data set is remembered for future (incremental) builds. You can switch the data set; however, if you do, you must start a full build initially. Also, if the output data set contains build output from a different PDB, the request is rejected.

You can specify whether to do the build ONLINE (during your TSO session) or as a submitted batch job. Requesting ALL starts a complete build, in which every object is rebuilt. Specifying MODIFIED

causes an incremental build, where only changed objects are rebuilt. When you start the first build to a data set, you must use ALL. You can also view the build report from the last build with option **4**.

When a build is started, you see a *Build Progress Display*, indicating the progress of the build process. Do not worry if you cannot follow all the messages being issued. They are saved in the \$BLDRPT member in your output data set. You can view them by invoking option **4**.

When the build is finished, you see the original panel with a message in the upper-right corner indicating that the build is successful. Other values are possible, but they are unlikely for a complete build.

You can also schedule builds as a batch job so that your TSO session is available for other work. Some setup is required at the installation level. The batch job must have the PDB data set. You must leave the current PDB so that the build can take place. Enter **BATCH** in the **Mode** field and the JCL JOB card information. The job is submitted from your TSO ID and must be a valid job for you.

To use the batch options of the build facility, you must first specify in the installation the names of the data sets that are required for the batch processing. Use option 0.11 (Settings) from the Primary Menu to define the data set names. These names form the *ddname* concatenations to be included in the JCL of the batch job. The job card information is saved across sessions.

The **Configuration** field can have a value of NORMAL (the default) or ALTERNATE:

- **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 plus sign (+) in their MVS system name.

NORMAL and ALTERNATE configurations must be built into different data sets.

# Tailoring the message stream

|   |  |  |  |                         |  |             |  |                  |  |                  |  |             |  |             |  |
|---|--|--|--|-------------------------|--|-------------|--|------------------|--|------------------|--|-------------|--|-------------|--|
| <p>AOFGCUST</p> <p>Option ===&gt; _____</p> <p>The following options apply only to your userid:</p> <ul style="list-style-type: none"> <li>1 Alias Define alias for standard commands</li> <li>2 Display Define display characteristics</li> <li>3 Message Define message and logging options</li> <li>4 Convert Define policy database conversion options</li> <li>5 Color Define color for inherited class values</li> </ul>  | <p><b>Settings Menu, option 3, to tailor messages issued during a build</b></p>  |  |  |                         |  |             |  |                  |  |                  |  |             |  |             |  |
| <table border="0" style="width: 100%;"> <tr> <td style="width: 30%; vertical-align: top;"> <p>AOFGCMMSG</p> <p>Command ===&gt; _____</p> <p>The following options apply only to your userid:</p> <ul style="list-style-type: none"> <li>10 Pol</li> <li>11 Dsn</li> <li>12 Wor</li> </ul> </td> <td style="width: 70%; vertical-align: top; background-color: #ffffcc; padding: 10px;"> <p><b>Message and Logging Functions</b></p> <p>The message level is used to control the number of messages that are produced on the "Command Progress Display" panel (AOFGZIPS):</p> <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%; text-align: right; padding-right: 10px;">Message level . . . . .</td> <td style="width: 10%; text-align: center; padding-right: 10px;">1</td> <td style="width: 10%; text-align: left; padding-right: 10px;">0 - Minimum</td> <td style="width: 10%; text-align: left; padding-right: 10px;">One message per major processing section</td> </tr> <tr> <td style="text-align: right; padding-right: 10px;">. . . . .</td> <td style="text-align: center; padding-right: 10px;">2</td> <td style="text-align: left; padding-right: 10px;">1 - Intermediate</td> <td style="text-align: left; padding-right: 10px;">One message per primary entry type being processed</td> </tr> <tr> <td style="text-align: right; padding-right: 10px;">. . . . .</td> <td style="text-align: center; padding-right: 10px;">3</td> <td style="text-align: left; padding-right: 10px;">2 - Maximum</td> <td style="text-align: left; padding-right: 10px;">Additional messages for other than primary entry</td> </tr> </table> <p>Enable checking for Processor Operations and write messages to build report:</p> <p>ProcOps syntax check . . . <u>NO</u> (YES NO)</p> <p>Enable checking of AT and MRT overrides requiring NetView:</p> <p>AT / MRT syntax check . . . <u>YES</u> (YES NO)</p> </td> </tr> </table> |  | <p>AOFGCMMSG</p> <p>Command ===&gt; _____</p> <p>The following options apply only to your userid:</p> <ul style="list-style-type: none"> <li>10 Pol</li> <li>11 Dsn</li> <li>12 Wor</li> </ul> | <p><b>Message and Logging Functions</b></p> <p>The message level is used to control the number of messages that are produced on the "Command Progress Display" panel (AOFGZIPS):</p> <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%; text-align: right; padding-right: 10px;">Message level . . . . .</td> <td style="width: 10%; text-align: center; padding-right: 10px;">1</td> <td style="width: 10%; text-align: left; padding-right: 10px;">0 - Minimum</td> <td style="width: 10%; text-align: left; padding-right: 10px;">One message per major processing section</td> </tr> <tr> <td style="text-align: right; padding-right: 10px;">. . . . .</td> <td style="text-align: center; padding-right: 10px;">2</td> <td style="text-align: left; padding-right: 10px;">1 - Intermediate</td> <td style="text-align: left; padding-right: 10px;">One message per primary entry type being processed</td> </tr> <tr> <td style="text-align: right; padding-right: 10px;">. . . . .</td> <td style="text-align: center; padding-right: 10px;">3</td> <td style="text-align: left; padding-right: 10px;">2 - Maximum</td> <td style="text-align: left; padding-right: 10px;">Additional messages for other than primary entry</td> </tr> </table> <p>Enable checking for Processor Operations and write messages to build report:</p> <p>ProcOps syntax check . . . <u>NO</u> (YES NO)</p> <p>Enable checking of AT and MRT overrides requiring NetView:</p> <p>AT / MRT syntax check . . . <u>YES</u> (YES NO)</p> | Message level . . . . . | 1  | 0 - Minimum | One message per major processing section | . . . . .        | 2  | 1 - Intermediate | One message per primary entry type being processed | . . . . .   | 3  | 2 - Maximum | Additional messages for other than primary entry |
| <p>AOFGCMMSG</p> <p>Command ===&gt; _____</p> <p>The following options apply only to your userid:</p> <ul style="list-style-type: none"> <li>10 Pol</li> <li>11 Dsn</li> <li>12 Wor</li> </ul>  | <p><b>Message and Logging Functions</b></p> <p>The message level is used to control the number of messages that are produced on the "Command Progress Display" panel (AOFGZIPS):</p> <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%; text-align: right; padding-right: 10px;">Message level . . . . .</td> <td style="width: 10%; text-align: center; padding-right: 10px;">1</td> <td style="width: 10%; text-align: left; padding-right: 10px;">0 - Minimum</td> <td style="width: 10%; text-align: left; padding-right: 10px;">One message per major processing section</td> </tr> <tr> <td style="text-align: right; padding-right: 10px;">. . . . .</td> <td style="text-align: center; padding-right: 10px;">2</td> <td style="text-align: left; padding-right: 10px;">1 - Intermediate</td> <td style="text-align: left; padding-right: 10px;">One message per primary entry type being processed</td> </tr> <tr> <td style="text-align: right; padding-right: 10px;">. . . . .</td> <td style="text-align: center; padding-right: 10px;">3</td> <td style="text-align: left; padding-right: 10px;">2 - Maximum</td> <td style="text-align: left; padding-right: 10px;">Additional messages for other than primary entry</td> </tr> </table> <p>Enable checking for Processor Operations and write messages to build report:</p> <p>ProcOps syntax check . . . <u>NO</u> (YES NO)</p> <p>Enable checking of AT and MRT overrides requiring NetView:</p> <p>AT / MRT syntax check . . . <u>YES</u> (YES NO)</p> | Message level . . . . .  | 1  | 0 - Minimum             | One message per major processing section | . . . . .   | 2  | 1 - Intermediate | One message per primary entry type being processed | . . . . .        | 3  | 2 - Maximum | Additional messages for other than primary entry |             |  |
| Message level . . . . .   | 1  | 0 - Minimum  | One message per major processing section   |                         |  |             |  |                  |  |                  |  |             |  |             |  |
| . . . . .   | 2  | 1 - Intermediate   | One message per primary entry type being processed   |                         |  |             |  |                  |  |                  |  |             |  |             |  |
| . . . . .   | 3  | 2 - Maximum  | Additional messages for other than primary entry   |                         |  |             |  |                  |  |                  |  |             |  |             |  |

Use customization dialog  
Settings Menu, option 3, to  
tailor messages issued  
during a build

— 7 —

4-61

The messages that you see in the Command Progress Display are also written to the Build Report. You can customize the amount of messages that are created. Selecting option **3** (MESSAGE) from the Settings Menu displays Message and Logging Functions panel. You can select the message level and specify whether logging is done to ISPF logs.

A message level of **0** means that when you start a build, messages are written to the Command Progress Display panel only when major processing sections are selected. In other words, messages are written when APLs are selected. Message level **1** means there is a message whenever an entry is processed. For example, if you select 1, you have one message for each APL processed, in addition to the one for the APL section. Message level **2** means that all details down to entry name and resource names are displayed. You might want to use message level **2** until you are more familiar with the messages that are shown during the build process.

# Lesson 4. Bulk policy updates

## Lesson 4: Bulk policy updates

- Perform flat file updates:
  - Update multiple policy entries, such as updating contact information across multiple applications
  - Create new policy entries
- Delete multiple policy entries

### What this lesson is about

This lesson describes how to perform bulk policy updates using the System Automation for z/OS ISPF customization panels.

### Objectives

After completing this lesson, you should be able to perform bulk policy updates using the System Automation for z/OS ISPF customization panel.

### References

*SC34-2717 System Automation for z/OS Version 4 Release 1 Defining Automation Policy*

## Flat file update

- You can perform bulk updates on the Policy Data Base with a three-step process:
  - Write selected policy data to a flat file
  - Edit the flat file
  - Update PDB with policies from flat file
- You can create new policy entries
- The flat file update supports APG, APL, MTR, and UET policies
- You can access the option from the Data Management panel by selecting option 3

```
AOFGIMPO          Data Management Menu
Option ===> =
1 Import from PDB      Import from a Policy Database
2 Import from Add-on   Import from predefined add-on policies
3 Update via File     Write selected data to file or read data from file
```

Note: Flat file update is not a replacement for customization dialog

### Flat file update

Suppose you must change the contact for a group of resources in the PDB. In the past, you were required to edit each policy item and make the change manually. You can update the automation policy more easily with a flat file option that is available from the Data Management panel. You can export (write) PDB data to a flat (sequential) file, edit it in TSO, and then import (read) it back into the PDB.

# Supported policy definitions

- APL: all
- SYS
  - DESCRIPTION
  - SYSTEM INFO (Operating System and Image/System name Read-only)
  - AUTOMATION SYMBOLS
  - WHERE USED
- MVS Components
  - DESCRIPTION
  - MESSAGES/USER DATA
  - MINOR RESOURCES
  - WHERE USED
- User E-T Pairs
  - DESCRIPTION
  - E-T DATA
  - WHERE USED
- Groups
  - DESCRIPTION
  - SYSPLEX
- APG:
  - DESCRIPTION
  - APPLGROUP INFO
  - RELATIONSHIPS
  - MESSAGES/USER DATA
  - WHERE USED
- MTR:
  - DESCRIPTION
  - MONITOR INFO
  - RELATIONSHIPS
  - MESSAGES/USER DATA
  - HEALTHSTATE
  - WHERE USED
- Processors
  - DESCRIPTION
  - PROCESSOR INFO
  - PSM INFORMATION
  - LPARS AND SYSTEMS

© Copyright IBM Corporation 2019

4-64

## Supported policy definitions

On this slide, you see some of the supported policy entries that you can update with a flat file.

## Write selected data from PDB to file

```

AOFGFSEL ■ Policy Database Update Selection Data written to file
Option ==> _____
1 Write selected data from Policy Database to file
  Entry Type . . . . . : APL          (? or type)
  Output File Name . . . . . : 'SAZOS.PDB.OPSDCOPY.UPD'
2 Perform syntax check for data in file
3 Update Policy Database with data from file
  Input File Name. . . . . : 'SAZOS.PDB.OPSDCOPY.UPD'
  Mode . . . . . . . . . : ONLINE      (ONLINE BATCH)
4 View write / update report
5 Edit output file
6 Append NEW/DEL templates to output file

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

```

Select option 1 to write policy to a default output PDB file name, concatenated with .UPD (period in front of UPD)

Select option 3 to update a Policy Database with data from a file

### *Write selected data from PDB to file*

First, you must write (export) data from the PDB to a flat file. Use an entry type of **APL**. You can specify the flat file name, or you can use a provided SA z/OS name by appending **.UPD** to the name of the PDB data set. In this example, data is written to the flat file, named **SAZOS.PDB.OPSDCOPY.UPD**, where SAZOS.PDB.OPSDCOPY is the name of the PDB file.

Select option 1 and press Enter to begin writing data to the file. After writing the data to the flat file, you can edit the file using option 5 (Edit output file).

Option 6 is based on new requirements to better document the format of NEW structures. Using option 6 on the Policy Database Update Selection panel, you now can add templates for NEW and DEL structures to the output file defined under option 1. You receive a template for each entry type. There is also a list of templates for all entry types that can be deleted, even though that template shows just one statement per entry type.



**Hint:** Be careful with your choice of flat file names. If you use the same file, it is overwritten with the new data. You do not see a confirmation panel where you can specify a different file name.

You can preallocate the output file with the following data set attributes:

- Organization: PS
- Record format: VB
- Record length: 259
- Space: 1 cyl

## Write selected data from PDB to file (continued)

| AOFGFNAM |        |               | Entry Name Selection                  |  | Row 25 to 48 of 50 |
|----------|--------|---------------|---------------------------------------|--|--------------------|
|          |        |               |                                       |  | SCROLL==> CSR      |
| Action   | Status | Entry Name    | Short Description                     |  |                    |
|          |        | RACF          | Resource Access Control Facility      |  |                    |
|          |        | RESOLVER      | TCP/IP Name Resolver                  |  |                    |
|          |        | RMF           | Resource Measurement Facility         |  |                    |
|          |        | RMFGAT        | IBM CONDITIONAL Data Gatherer         |  |                    |
|          |        | RV_MSTR_CLASS | Master class for RV Applications      |  |                    |
|          |        | RVCLASS       | Class definitions for RV applications |  |                    |
| SELECTED | RV01   | RV01          | RV01 only in system MVSA              |  |                    |
|          | RV02   | RV02          | RV02 only in system MVSB              |  |                    |
|          | RV03   | RV03          | RV_BASIC Sysplex APG                  |  |                    |
|          | RV04   | RV04          | RV04 only in system MVSA              |  |                    |
|          | RV05   | RV05          | RV05 only in system MVSB              |  |                    |
|          | RV06   | RV06          | RV06 RV_BASIC Sysplex APG             |  |                    |
|          | RV07   | RV07          | RV07 only in system MVSA              |  |                    |
|          | RV08   | RV08          | RV08 only in system MVSB              |  |                    |
|          | RV09   | RV09          | RV09 RV_MOVE MOVE Group               |  |                    |
|          | SDFS   | SDFS          | System Display and Search Facility    |  |                    |

| AOFGFPOL |          |                     | Policy Selection |  | Row 1 to 22 of 24 |
|----------|----------|---------------------|------------------|--|-------------------|
|          |          |                     |                  |  | SCROLL==> CSR     |
| Action   | Status   | Policy Name         |                  |  |                   |
|          | SELECTED | DESCRIPTION         |                  |  |                   |
|          | SELECTED | UPWARD CLASS        |                  |  |                   |
|          | SELECTED | APPLICATION INFO    |                  |  |                   |
|          | SELECTED | AUTOMATION FLAGS    |                  |  |                   |
|          | SELECTED | APPLICATION SYMBOLS |                  |  |                   |
|          | SELECTED | RELATIONSHIPS       |                  |  |                   |
|          | SELECTED | MESSAGES/USER DATA  |                  |  |                   |
|          | SELECTED | STARTUP             |                  |  |                   |
|          | SELECTED | SHUTDOWN            |                  |  |                   |
|          | SELECTED | THRESHOLDS          |                  |  |                   |
|          | SELECTED | MINOR RESOURCES     |                  |  |                   |
|          | SELECTED | CICS CONTROL        |                  |  |                   |

© Copyright IBM Corporation 2019

4-66

Write selected data from PDB to file (continued)

The next screen displays a list of the defined Applications. Select the Application policies that you want to export to the flat file and press Enter. The Application status is changed to SELECTED. You can use **SELECT \*** to select all Applications or **M \*** to deselect all Applications. Press Enter a second time to continue to the next panel. The second panel has a list of the supported Application policy items. Select the policy items that you want to write to the flat file and press Enter. In this case, all policy items are selected with a **SELECT \*** command. The status for the selected entries changes to SELECTED. Press Enter again to continue to the next panel.

In this example, all policy items are written for the RV01 Application to the flat file, SAZOS.PDB.OPSDCOPY.UPD. When the write is complete, you see the Policy Database Update Selection panel.

## Flat file format: Header information

```

BROWSE      SAZOS.PDB.OPSDCOPY.UPD          Line 0000000000 Col 001 080
Command ==>                                         Scroll ==> PAGE
***** Top of Data *****
* UserId           INGC102
* PolicyDB Name   OPSDEMOCOPY
* PolicyDB Data Set SAZOS.PDB.OPSDCOPY
* Version          4.1
* Creation Date   Wednesday, 20 Mar 2019
* Creation Time   15:21:01

FILE CREATION TIMESTAMP 20190320152101

UPD APL
  DESCRIPTION
    Short Description : RV01
    Long Description 1 : ( field replacement )
    Long Description 2 : RV01 only in system MVSA
    Long Description 3 :
    Long Description 4 :
    Long Description 5 :
  UPWARD CLASS       : RVCLASS
  APPLICATION INFO   : ( field replacement )
    Restart after IPL :
    Start Delay       : 00:01:00

```

- The beginning of the flat file contains a header section, such as a timestamp when the flat file was created
- The timestamp is used to verify that no updates occurred since the data was written to the flat file

### Flat file format: Header information

On this slide, you see a portion of the flat file containing the RV01 policy, SAZOS.PDB.OPSDCOPY.UPD. The header provides information about the request, such as which user wrote to the file, when the update occurred, and a time stamp. The time stamp is important for synchronizing updates. For most cases, this information is display-only.

On this slide, you see also some of the data that was written to the flat file. The first line must always be **UPD APL**. The remaining lines are excerpts from the file to show examples of three policy items for RV01:

- DESCRIPTION
- UPWARD CLASS
- APPLICATION INFO

## Flat file format example: RV01 policies

```
BROWSE   SAZOS.PDB.OPSDCOPY.UPD          Line 0000000097 Col 001 080
Command ==> ■
    AT Capture message DOM option      :
    AT override                         :
    Message ID : WTORS
    Description : Specifications for WTORS
    Action     - CODE
                : VAPL21000A
                : RV01
                :
                : IMPORTANT
                : VAPL21999A
                : RV01
                :
                : NORMAL
                ( block replacement )
                : STARTUP
    Phase     :
    Mixed case :
    Action     - CMD
                : COLD
                :
                : MVS S RUNVAPL,JOBNAM=&SUBSJOB,NAME=VA
    Type      : NORM
                :
                : MVS S RUNVAPL,JOBNAM=&SUBSJOB,NAME=VA
    Type      : WARM
                :
                : MVS S RUNVAPL,JOBNAM=&SUBSJOB,NAME=VA
    Action     :
    Type      :
    Automated Function/*'*
    Command Text :
    Type      :
    Automated Function/*'*
    Command Text :
    Type      :
    Automated Function/*'*
    Command Text :
```

### Flat file format example: RV01 policies

On this slide, you see more of the data that was written to the flat file for RV01:

- MESSAGES/USER DATA
- STARTUP

# Flat file syntax

```

UPD APL : <Entry_Name>
<Policy_Item> ( field|block|policy replacement )
  <Field_1> : <Value_1>
  <Field_2> : <Value_2>

```

|  |  |
|--|--|
| <b>UPD</b>                                 | Identifies this file as an update request          |
| <b>APL</b>                                 | Entry type is Application: APG, APL, MTR, or UET   |
| <b>Entry_Name</b>                          | Application name, such as <i>TCP_IP</i>            |
| <b>Policy_Item</b>                         | Automation policy, such as <i>APPLICATION INFO</i> |
| <b>Field, block, or policy replacement</b> | Identifies the type of update                      |
| <b>Field_n</b>                             | Policy keyword, such as <i>Job Name</i>            |
| <b>Value_n</b>                             | Policy keyword, a value such as <i>TCPIP</i>       |

## Flat file syntax

On this slide, you see the syntax of the statements that are in the flat file:

- **UPD:** Identifies this file as an update request. ADD identifies this file as a new policy.
- **APL:** The entry type can be APG, APL, MTR, or UET for update.
- **Entry\_Name:** Name of the Application (TCPIP).
- **Policy\_Item:** The individual policy items, such as DESCRIPTION, APPLICATION INFO, WHERE USED, RELATIONSHIPS.
- **Field replacement:** Identifies the type of update. Use the fields from the flat file to replace only the same fields of the policy item in the PDB; for example, the **Job Name** field of the APPLICATION INFO policy.
- **Policy replacement:** Replace the entire policy item in the PDB using the policy from the flat file, for example, the RELATIONSHIPS policy.
- **Block replacement:** Replace a subset of the policy in the PDB with the block of policy data from the flat file, for example, a complete message specification of the MESSAGES/USER DATA policy.

- **Field\_n:** Is replaced by the policy keyword, such as Job Name.
- **Value\_n:** Is the value for the field, such as TCPIP.

## PDB update from flat file

- Your user ID must have exclusive access to the PDB
- Synchronization errors are possible. Other users can modify the policy from the customization dialog while you are editing the flat file
- All modified and new entries are updated or added to the PDB without the display of any panel
- Messages are shown in the Command Progress Display window and held on screen until the user presses the Enter key  
Messages are logged to a report file. Check the report file for all messages
- Syntax is checked before update; If errors exist, the update is not performed

© Copyright IBM Corporation 2019

4-70

### PDB update from flat file

To read policy data back into the PDB, your user ID must have exclusive access to the PDB. If your user ID does not have exclusive access to the PDB, you should see an error message similar to this one:

ALLOCATE not possible. Exclusive access to data set 'SAZOS.PROD.PDB' is needed.  
However, the data set is in use by user ID or jobname USER1.

As the flat file is processed, messages are displayed in the Command Progress Display. If you have several updates, the message display might scroll quickly. You can view the report file to see all the messages.

## Flat file update serialization

```

■ AOFGZIPS                               Command Progress Display
Reading data set 'SAZOS.PDB.OPSDCOPY.UPD'.
Start syntax checking.
Checking syntax of function UPD for APL RV02.
ALTERED (Line 10): Entry RV02 was changed in the Policy Database after
.....cont. data was copied to file.
WARNING: 1 entry alteration(s) detected. Data in the Policy Database
.....cont. was changed after it was copied to the file.

■ AOFGFTS                               Policy Database Update Confirmation
Command ==> _____
Press _____
F1=HE _____
F7=UP _____
The following entries have been updated in the Policy Database
was copied to file.
Press ENTER to start the policy database update.
Press CANCEL or END to cancel update.

Entry Name          Entry Type    Last Change      UserId
RV02                APL          19/03/20 15:39   INGC102

```

- This panel displays the entry (RV02), the name of the person who last modified it (INGC102), and when it was changed. Press Enter if you want to update the PDB, or press PF3 to cancel
- Timestamp is written to output file when all PDB data is opened, and the timestamp is checked when data is read back in
- If an entry (in this example, RV02) is updated after it is written to the flat file, the Policy Database Update Confirmation panel is displayed

© Copyright IBM Corporation 2019

4-71

### Flat file update serialization

Other system administrators might have accessed and changed the PDB contents while you were editing the flat file contents. When you read a flat file in, the time stamp in the flat file is compared to the time stamp stored internally in the policy for each entry. This time stamp helps to avoid destroying changes that you made while you were editing the flat file.

If the time stamp in the flat file is older than the time stamp stored for an entry (such as RV02), you see error messages followed by a confirmation panel. On the confirmation panel, you can cancel the import or continue. If you continue the update, you override the existing data.

## Creating new entries from the flat file

- Edit the flat file
- Use the keywords
- Write to PDB from flat file when done

| Keyword              | Purpose  |
|----------------------|--|
| <b>NEW</b>           | Specify a new entry in the PDB.                      |
| <b>COPY</b>          | Initialize new entry with a copy of another entry.   |
| <b>WHERE_USED</b>    | Link the new entry to an existing one.               |
| <b>LINK_TO_CLASS</b> | Link the new entry to an existing Application class. |

### *Creating new entries from the flat file*

You can also create new policy definitions by using the flat file, and import them into the PDB by using the keywords shown in the table on this slide. You can use the WHERE\_USED keyword multiple times for each Application.

## Example: Creating a new application

- Write RV01 APL policy to flat file
- Edit flat file
  - Add statements to create the **new** application, **RV05**, using **RV01** as a model (**copy**)
  - Change appropriate lines for the new application, RV05
- Use option 3 (Update Policy Database with data from file)

```
NEW API          : RV05
Subsystem Name   : RV05
Job Name         : RV05
Short description: RV05 copied from RV01
LINK_TO_CLASS    : RVCLASS
WHERE_USED       : RV_GROUP
COPY             : RV01
```

### Example: Creating a new application

On this slide, you see how to create a new Application called RV05 from an existing Application called RV01. You use the following steps:

1. Write the policy definitions for Application RV01 to the file.
  2. Add NEW, COPY, WHERE\_USED, and LINK\_TO\_CLASS statements like the ones shown here. Locate them in the flat file between the header and policy for RV01 that was written to the file.
    - **NEW API:** Create a new Application named RV05.
    - **Subsystem Name:** RV05
    - **Job Name:** RV05
    - **Short description:** RV05 copied from RV01
    - **LINK\_TO\_CLASS:** Link RV05 to the RVCLASS Application class.
    - **WHERE\_USED:** Link RV05 to the RV\_GROUP Application Group.
    - **COPY:** RV01
- Use RV01 as a template. 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

category, subcategory, or class-instance compatibilities. 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.

3. Change the RV01 policy for RV05, as appropriate. For example, change the **Short description**, **Long description 1**, **Long description 2**, and **Job Name** fields.

```

UPD APL                               - RV05
DESCRIPTION                           ( field replacement )
    Short description : RV05
    . . .

APPLICATION INFO                     ( field replacement )
    Application Type : STANDARD
    Job Name          : RV05
    . . .

MESSAGES/USER DATA                  ( block replacement )
    Message id       : IEF404I
    Description      :
    Action           - OVR
    NetView AT condition : MSGID = 'IEF404I' & TOKEN(2) = 'RV05BKUP'
    NetView AT action 1   : EXEC(CMD('INGEVENT RV05EVT REQ=SET'))
    . . .

```

These are only a subset of the changes that are required.

4. Read the flat file into the PDB.

When the update is complete, you see the Policy Database Update Selection panel with this message: **Data written to PDB**. View the report file to make sure there are no errors.

## Deleting multiple entries

Use the confirmation panel to perform these tasks:

- Delete for WHERE USED
- Delete for link to an application class
- Delete for multiple entries:

Confirmation level: ALL, LINKED, or NONE

- ALL: User is asked to confirm every deletion
- LINKED: User is asked to confirm only entries that are linked to other entries
- NONE: No confirmation performed

### *Deleting multiple entries*

On the confirmation panel, you have the following options to delete the entries:

- Without any further confirmation (NONE)
- With confirmation for linked entries only (LINKED)
- With a confirmation for every entry (ALL)

If an error occurs with a delete, all deletions up to that point are kept, a message is issued, and the process stops.

## **Deleting multiple entries example**

Delete policy for nine applications and one application class

### *Deleting multiple entries example*

On this slide, you see an example of deleting nine Applications and the Application class that they are associated with. To start the delete process, press Enter. The Confirm Delete panel is displayed. In this example, the first entry to be deleted is the RVCLASS Application class.

You see this confirmation panel under these circumstances:

- Multiple deletes are performed in a series of deletes in one step.
  - The entry is either linked by another entry (WHERE USED), or it is a class with instance links.

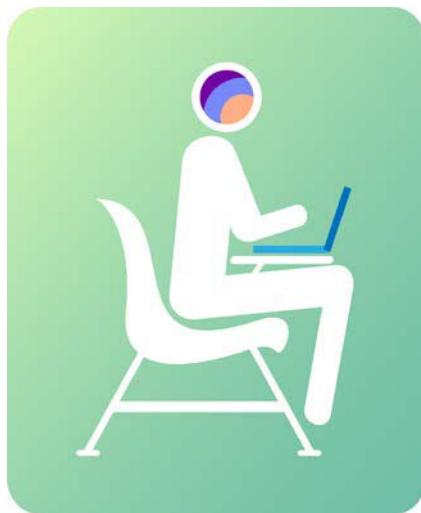
You can control how often this panel is shown during the deletion of the selected entries by using one of the values in the **Show confirmation for** field:

- **ALL**: Show a confirmation panel for each entry to be deleted. ALL is the default value.
  - **LINKED**: Show the confirmation panel only for entries with WHERE USED links, or for classes with linked instances. All other entries without links are deleted without confirmation.
  - **NONE**: All subsequent confirmation panels are suppressed and all requested entries, including their links, are deleted.

## Student exercises

### Unit 4

1. Use the customization dialog to automate a single system



© Copyright IBM Corporation 2019

4-76

*Student exercises*

Perform the exercises for this unit.

## Summary

---

Now that you have completed this unit, you should be able to perform the following tasks:

- Define an application in the Policy Data Base
- Describe the automation configuration file (ACF) format and how it is used
- Build the automation configuration file
- Perform flat file updates to the Policy Data Base

### *Summary*

Now that you have completed this unit, you should be able to perform the following tasks:

- Define an application in the Policy Data Base
- Describe the automation configuration file (ACF) format and how it is used
- Build the automation configuration file
- Perform flat file updates to the Policy Data Base



## 5 Additional customization

---



### Unit 5 Additional customization



Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

---

**What this unit is about**

In this unit, you explore additional customization features and functions that are available with SA z/OS. Topics include use of symbolics in the policy, Application classes, suspend file, Notify Operators, Service Periods, Events, Triggers, and monitor resources.

**How you check your progress**

You can check your progress in the lab exercises.

**References**

*SC34-2717 System Automation for z/OS Version 4  
Release 1 Defining Automation Policy*  
[https://www.ibm.com/support/knowledgecenter/SSWRC\\_J\\_4.1.0/com.ibm.safos.doc\\_4.1/manuals.html](https://www.ibm.com/support/knowledgecenter/SSWRC_J_4.1.0/com.ibm.safos.doc_4.1/manuals.html)

## Objectives

---

When you complete this unit, you can perform the following tasks:

- Define Applications by using SA z/OS symbolics
- Define Application classes and linked instances
- Implement planned suspend capability
- Define Automation setup for Notify Operators, Automation Timers, MVS Automation, JES spool management, and Automation Operators
- Define Service Periods, Triggers, Events, and Pacing Gates
- Define monitor resources
- Describe the SA z/OS dynamic Automation Table build feature

When you complete this unit, you can perform the following tasks:

- Define Applications by using SA z/OS symbolics
- Define Application classes and linked instances
- Implement planned suspend capability
- Define Automation setup for Notify Operators, Automation Timers, MVS Automation, JES spool management, and Automation Operators
- Define Service Periods, Triggers, Events, and Pacing Gates
- Define monitor resources
- Describe the SA z/OS dynamic Automation Table build feature

# Lesson 1. Common Application definitions

## Lesson 1: Common Application definitions

- Using symbols
- Defining application classes and instances
- Suspend file

### What this lesson is about

This lesson describes how to define common application policy definitions and how to implement planned suspend capability.

### Objectives

After completing this lesson, you should be able to define common application policy definitions and implement planned suspend capability.

### References

*SC34-2717 System Automation for z/OS Version 4  
Release 1 Defining Automation Policy*

Frequently, in an enterprise many Applications are similar from an operational point of view. The dialogs provide ways to efficiently specify and administer them. SA z/OS provides support for defining symbolics that are related to system automation (*SA z/OS symbolics*) and symbolics

related to the system itself (*system symbolics*). You can specify both in the automation policies. System automation symbols are described in the *IBM System Automation for z/OS: Defining Automation Policy* manual.

## System automation symbols

- You can use system automation symbols and system symbols in automation policy
- You can use the same Application on many systems with system-dependent variations
  - Use single definition that works differently on different systems
- Values of system automation symbols are defined at SYS level only
- Descriptions of system automation symbols are defined in these policies:
  - ENTerprise policy: SYMBOL DESCRIPTION
  - SYStem policy: AUTOMATION SYMBOLS

### SA z/OS symbols

An Application can run on several systems. Its policy data might vary on each system in a predictable way. For example, a VTAM Application running on SYSA might have a job name of NETA, and its start command might be MVS S NETA,,,LIST=A1). The VTAM Application on SYSB might be named NETB and be started by a command MVS S NETB,,,LIST=B1).

Such predictable variations can be managed by assigning a set of system automation symbols or system symbols for each system and using these to specify the policy for the Applications. By selecting the Application on several systems, different policy values are used because of the substitution of the symbols.

You can also use the z/OS system symbols (symbolics) in MVS commands. For example, if an &LPAR system symbol is defined, you can define the command

**MVS S NET&LPAR,,,LIST=&LPAR.1**) to start VTAM. The z/OS system symbols are defined in IEASYMxx of PARMLIB. Do not define system symbolics of the form &AOCCLOSE. The MVS system programmer usually sets the system symbolic values, but the SA z/OS administrator sets the system automation symbols values.

## Defining system automation symbols

AOFGSSYM System Automation Symbols (AOCCLONEx)  
Command ==> █

Entry Type : System PolicyDB Name : OPSDEMOCOPY  
Entry Name : MVSA Enterprise Name : OPSDEMOCOPY

More: +

AOCCLOSE . . . 1  
General Use:  
Description . . for use by AddOn PDB Entries

AOCCLOSE1 . . . A  
General Use:  
Description 1 . for use by AddOn PDB Entries

AOCCLOSE2. . . .  
General Use:  
Description 2 .

AOCCLOSE3. . . .  
General Use:  
Description 3 .

AOCCLOSE4. . . .  
General Use:  
Description 4 .

AOCCLOSE5. . . .  
General Use:

&AOCCLOSE = 1

&AOCCLOSE1 = A

Each symbol can have a general description (defined in ENT policy) and a specific description (defined in SYS policy). The symbol value is defined in the SYS policy for each system.

© Copyright IBM Corporation 2019

5-5

## *Defining SA z/OS symbols*

System automation symbols are defined by selecting the AUTOMATION SYMBOLS policy item for an existing system (SYS policy). You can specify up to 36 system automation symbols for a system: AOCCLOSE, AOCCLOSE1 - AOCCLOSE9, and AOCCLOSEA - AOCCLOSEZ. You can define symbol descriptions at the ENTerprise and SYStem levels. You can define symbol values only at the SYStem level. In this example, there are no symbol descriptions defined at the ENTerprise level. The AOCCLOSE and AOCCLOSE1 system automation symbols are used in the examples shown in this lesson.

## Example: Symbols in the job name

| Application Information          |   | Line 00000001  |
|----------------------------------|---|----------------|
| Command ==> _____                |   | Scroll ==> CSR |
| Entry Type : Application         | PolicyDB Name :                                   | OPSDEMOCOPY    |
| Entry Name : VTAM                | Enterprise Name :                                 | OPSDEMOCOPY    |
| Category . . . . .               | (IBM-defined, user-defined or blank,<br>see help) |                |
| Subcategory . . . . .            | (IBM-defined, user-defined or blank,<br>see help) |                |
| Subsystem Name . . . . .         | VTAM  |                |
| Job Type . . . . .               | (MVS NONMVS TRANSIENT)                            |                |
| Job Name . . . . .               | VTAM&AOCCLOSE.                                    |                |
| Transient Rerun . . . .          | (YES NO)  |                |
| Scheduling Subsystem . . . .     | (MSTR, JES Subsystem)                             |                |
| JCL Procedure Name . . . .       |   |                |
| Job Log Monitor Interval . . . . | (mm:ss NONE)                                      |                |
| Captured Messages Limit . . . .  | (0 to 999)  |                |
| Desired Available . . . .        | (ALWAYS ONDEMAND ASIS)                            |                |

Job name resolves to VTAM1, based on &AOCCLOSE value of 1

© Copyright IBM Corporation 2019

5-6

### Example: Symbols in the job name

Any Application can use any of the system automation symbols in its definition. Those Applications that use symbols as part of the job name are typically referred to as *cloned Applications*.

## Example: Symbols in STARTUP command

| AOFGASTT                   |             | Subsystem Startup Processing              |
|----------------------------|-------------|---|
| Command ==>                |             |   |
| Entry Type :               | Application | PolicyDB Name : OPSDEMOCOPY               |
| Entry Name :               | VTAM        | Enterprise Name : OPSDEMOCOPY             |
| Scheduling Subsystem . . . |             | (MSTR, JES Subsystem)                     |
| JCL Procedure Name . . .   |             |   |
| Startup Parameters . . .   |             |   |
| .,,(LIST=&AOCCLOSE.)       |             |   |
| Line Commands: S/C (Cmd)   |             |   |
| Cmd                        | Phase       | Description                               |
| PRESTART                   |             | Executed before startup is initiated      |
| STARTUP                    |             | Executed to initiate the startup          |
| POSTSTART                  |             | Executed after startup has completed      |
| REFRESHSTART               |             | Executed after recycle or INGAMS REFRESH  |
| ANYSTART                   |             | Executed after POSTSTART and REFRESHSTART |

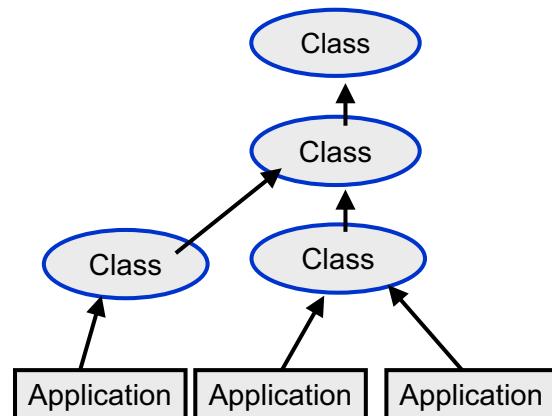
- Start command: S VTAM,,,(LIST=1) where &AOCCLOSE1 resolves to 1

### Example: Symbols in STARTUP command

You can also use the system automation symbols in other parts of the Application specification. The example shows a use for creating a start command. The full stop (period) is required to end the &AOCCLOSE specification. You can use the system automation symbols, for example, in job names, procedure names, start commands, or relationships.

# Application classes and instances

- **Application classes** define common policy for a grouping of Applications; for example, Start commands or relationships.
  - Reduces definition work
  - Reduces maintenance effort
- Common policy definitions are inherited at *instance* level.
- **Nested Application classes** extend the inheritance across multiple levels of a class hierarchy.
- Policy can also be derived from Application defaults (ADF), system defaults (SDF), MVS component defaults (MDF), or sysplex defaults (XDF).



© Copyright IBM Corporation 2019

5-8

## Application classes and instances

An *Application class* is a special form of Application definition that provides a template for a set of Applications to share some common policy specifications. Typically, classes are useful for Applications with similar shutdown policies, parents, and so on. An update to overall policy can be done with one change at the class level. Changes in class policies are propagated to the linked instances.

Definitions are made in the template (class definition) for the common parts of a definition: start commands, delays, and dependencies, for example. The class definition itself does not have a job name specified, and it is not specifically selected for any system. Classes are not actual resources.

Instances are linked to a class definition. If an instance does not have a field defined, but the class does, the instance inherits the definition from the class definition. Classes can be linked to other classes.

Application classes create an extra level of *default* policy definition. You can define the policy at several levels. The search for the policy definition begins at the resource definition until a hardcoded SA z/OS default is found.

The following values are used if they are specified:

- If specified in the policy for the application, that value is used.
- If specified in the class definition for that application, it is used.
- If specified in the Application DEFAULTS policy object, that value is used.
- If specified in the SYSTEM DEFAULTS policy object, that value is used.
- If specified in the MVS Component defaults, that value is used.
- If specified in the sysplex defaults, that value is used.
- Sometimes, a default value hard-coded in SA z/OS is used.

## Nested Application classes

- Extend the inheritance across multiple levels of a tree
  - UPWARD CLASS policy links Application instance or class to an Application class
  - DOWNWARD CLASS/INST policy links Application class to multiple classes or instances
- Customization dialog displays and highlights policy that is inherited from application classes
- Inherited policy is displayed in yellow by default
  - Color can be customized by using the customization dialog Settings Menu
  - Changes are saved across sessions

© Copyright IBM Corporation 2019

5-9

### *Nested Application classes*

The UPWARD CLASS policy replaces the LINK TO CLASS policy for Application instances in previous releases. The DOWNWARD CLASS/INST policy replaces the LINK TO INSTANCES policy for Application classes in previous releases.

# Nested Application classes example

## APPLICATION INFO Policy Item

RV01

|   |                 |                      |
|---|-----------------|----------------------|
| <b>Subsystem Name . . . . .</b>         | <b>RV01</b>     | <b>RV_CLASS</b>      |
| <b>Job Name . . . . .</b>               | <b>RV01</b>     |                      |
| <b>JCL Procedure Name . . . . .</b>     | <b>RUNVAPL</b>  |                      |
| <b>Start Timeout . . . . .</b>          | <b>00:01:00</b> |                      |
| <b>Restart option . . . . .</b>         | <b>ALWAYS</b>   |                      |
| <b>Shutdown Pass Interval . . . . .</b> | <b>00:01:00</b> |                      |
| <b>Inform List . . . . .</b>            | <b>SDF NMC</b>  | <b>RV_MSTR_CLASS</b> |
| ...                                     |                 |                      |
|   |                 | <b>SDF NMC</b>       |

- **RV01** inherits these policies:

- JCL Procedure Name from RV\_CLASS
- Restart option from RV\_CLASS
- Shutdown Pass Interval from RV\_CLASS
- Inform List from RV\_MSTR\_CLASS

- **RV\_CLASS** inherits Inform List from RV\_MSTR\_CLASS

Use **CN** command to display **class name** of inherited field

## Nested Application classes example

In this example, Application RV01 is linked to the RV\_CLASS Application class. RV01 inherits values for several policies from RV\_CLASS:

- JCL Procedure Name
- Restart option
- Shutdown Pass Interval

Because RV\_CLASS is linked to the RV\_MSTR\_CLASS, RV01 also inherits the Inform List policy from the RV\_MSTR\_CLASS.

## Data inheritance rules

Inheritance rules depend on the policy, as follows:

- APPLICATION INFO: Data inherited for each field
- THRESHOLDS: All data inherited, if none defined for the instance
- RELATIONSHIPS: All relationships inherited, if none defined for the instance  
External startup and shutdown inherited individually
- MESSAGES/USER DATA: Data inherited per message ID  
Automation overrides not displayed because they apply to the build of the Automation Table

## Nested Application classes: Notes

- No limit on number of supported class levels
- Application instance linked to one class only
- Application class linked to one upward class only
- Link to same type of Application only, unless class does not specify type
- Inherited policy data is displayed with existing policy data
  - If another user modifies class policy, changes are not automatically refreshed
  - If you modify the inherited value, it is stored at the instance level only
- Use Application Defaults (ADF) and System Defaults (SDF) policy for global defaults
  - Uses fewer NetView global variables

### *Nested Application classes: Notes*

By default, inherited policy data is displayed in yellow. If you modify the inherited data, it is stored at the instance level only. The Application class data is not changed.

## Create an Application class

|                                      |                      |  |
|--------------------------------------|----------------------|--|
| AOFGDYN3                             | Define New Entry     |  |
| Command ==> _____                    | Object Type          |  |
| Define new entry of type Application |                      |  |
| Entry name . . . . .                 | RVCLASS              | • CLASS for Application class                                      |
| Subsystem Name . . . . .             |                      | More: +  |
| Object Type . . . . .                | CLASS                | • INSTANCE for an Application                                      |
| Category . . . . .                   |                      | (CLASS INSTANCE)<br>(IBM-defined, user-defined or blank, see help) |
| Subcategory . . . . .                |                      | (IBM-defined, user-defined or blank, see help)                     |
| Job Type . . . . .                   |                      | (MVS NONMVS TRANSIENT)   |
| Job Name . . . . .                   |                      |  |
| Transient Rerun . . . . .            | (YES NO)             |  |
| AOFGENAM                             | Entry Name Selection | Row 28 to 48 of 50   |
| Command ==> _____                    | Scroll ==> CSR       |  |
| Entry Type : Application             |                      |  |
| Action                               | Entry Name           | PolicyDB Name : OPSDEMOCOPY<br>Enterprise Name : OPSDEMOCOPY       |
| _____                                | RMFGAT               | C Short Description  |
| _____                                | RV_MSTR_CLASS        | RMF Monitor III Data Gatherer                                      |
| _____                                | RVCLASS              | * Master class for RV Applications                                 |
| _____                                | RV01                 | * Class definitions for RV applications                            |
| _____                                | RV02                 | RV01 only in system MVSA   |
| _____                                | RV03                 | RV02 only in system MVSB   |
| _____                                | RV04                 | RV03 RV_BASIC Sysplex APG  |
| _____                                | RV05                 | RV04 only in system MVSA   |
| _____                                | RV06                 | RV05 only in system MVSB   |
| _____                                | RV07                 | RV06 RV_BASIC Sysplex APG  |
| _____                                |                      | RV07 only in system MVSA   |

Remaining policies are defined at the class level, such as APPLICATION INFO policy item.

### Create an Application class

The Define New Entry panel (AOFGDYN3) contains the *Object Type* field. Object Type can be INSTANCE for an Application or CLASS for an Application class. You define a CLASS definition when you first create the Application policy object and overtype the value of INSTANCE with CLASS. Enter a description, if required. Press PF3 to create the class, and define the policy items.

Classes are identified on the *Entry Name Selection* panel with an asterisk (\*) in the column marked **C**. In this case, there are two classes: RV\_MSTR\_CLASS and RVCLASS. Any values provided in the CLASS definition are inherited by its associated instances, if no value is entered specifically for the INSTANCE.

## UPWARD CLASS: Upward link to class

| AOFGXI2C                   |            | Upward Link to Class |          | Row 1 to 11 of 11<br>SCROLL==> CSR |                 |
|----------------------------|------------|----------------------|----------|------------------------------------|-----------------|
| Entry Type                 | Entry Name |                      |          | PolicyDB Name                      | Enterprise Name |
| Action                     | Status     | Entry Name           | Category | Upward Class Link                  |                 |
|                            |            | C_AM                 |          |                                    |                 |
|                            |            | C_APPL               |          |                                    |                 |
|                            |            | C_DVIPA              |          |                                    |                 |
|                            |            | C_JES2               | JES2     |                                    |                 |
|                            |            | C_RUNMODE            |          |                                    |                 |
|                            |            | C_USS_APPL           | USS      | C_RUNMODE                          |                 |
|                            |            | C_USS_FILE           | USS      | C_RUNMODE                          |                 |
|                            |            | C_USS_PORT           | USS      | C_RUNMODE                          |                 |
|                            |            | DEMOCLASS            |          |                                    |                 |
|                            |            | RV_MSTR_CLASS        |          |                                    |                 |
| →                          | SELECTED   | RVCLASS              |          | RV_MSTR_CLASS                      |                 |
| ***** Bottom of data ***** |            |                      |          |                                    |                 |

- **UPWARD CLASS** policy item links Application or Application class (upward) to an Application class
- In this example, you see the following links:
  - APL RV01 is linked to RVCLASS
  - RVCLASS is linked to RV\_MSTR\_CLASS

### UPWARD CLASS: Upward link to class

You can use the UPWARD CLASS policy item to link an Application instance or Application class to an Application class. This slide illustrates how to link the Application instance RV01 to the RVCLASS class. The class object must exist in the PDB for this to be accomplished. RVCLASS is linked to another Application class, RV\_MSTR\_CLASS, and RV01 inherits policy from the RVCLASS and RV\_MSTR\_CLASS.

## DOWNTWARD CLASS/INST: Downward link

| AOFGXC2I<br>Command ==> |  | Downward Link to Classes or Instances                        |                 | Row 25 to 45 of 49<br>SCROLL==> CSR |
|-------------------------|--|--|-----------------|-------------------------------------|
| Entry Type              | Entry Name   | PolicyDB Name  | Enterprise Name |                                     |
|                         | Entry Type : Application<br>Entry Name : RV_MSTR_CLASS | PolicyDB Name : OPSDEMOCOPY<br>Enterprise Name : OPSDEMOCOPY |                 |                                     |
| Action                  | Status   | Entry Name   | C Category      | Upward Class Link                   |
|                         |  | RACF   |                 | C_RUNMODE                           |
|                         |  | RESOLVER   |                 | C_APPL                              |
|                         |  | RMF  |                 | C_APPL                              |
|                         |  | RMFGAT   |                 | C_APPL                              |
|                         | SELECTED   | RVCLASS  | *               | RV_MSTR_CLASS                       |
|                         |  | RV01   |                 | RVCLASS                             |
|                         |  | RV02   |                 | RVCLASS                             |
|                         |  | RV03   |                 | RVCLASS                             |
|                         |  | RV04   |                 | RVCLASS                             |
|                         |  | RV05   |                 | RVCLASS                             |
|                         |  | RV06   |                 | RVCLASS                             |
|                         |  | RV07   |                 | RVCLASS                             |
|                         |  | RV08   |                 | RVCLASS                             |
|                         |  | RV09   |                 | RVCLASS                             |
|                         |  | SDSF   |                 | C_APPL                              |
|                         |  | SYSVAPPL   |                 | C_RUNMODE                           |
|                         |  | SYSVIPLC   |                 | C_RUNMODE                           |
|                         |  | SYSVSSI  |                 | C_APPL                              |
|                         |  | TCPPIP   |                 | C_APPL                              |
|                         |  | TN_PORT  | USS             | C_USS_PORT                          |
|                         |  | TN3270   |                 | C_APPL                              |

- **DOWNTWARD CLASS/INST** policy item links Application class (downward) to an Application or another Application class
- In this example, RV\_MSTR\_CLASS is linked downward to RVCLASS

### DOWNTWARD CLASS/INST: Downward link

You can use the DOWNTWARD CLASS/INST policy item to link an Application class to an Application instance or another Application class. On this slide, you see an example of how to link the RV\_MSTR\_CLASS Application class to another Application class, RVCLASS, using the DOWNTWARD CLASS/INST policy item.

## Implement planned suspend capability

- Problem: Add resources to the policy, which are not ready to be used
- Solution: Planned suspend capabilities of new **suspend file**
- Use SUSPENDFILE parameter in the AM PARMLIB member HSAPRMxx
- Suspend file is processed during a:
  - Manager **COLD/WARM start**
  - Configuration **REFRESH**. Only when configuration file has changed
- **Suspend requests** are injected for all resources in the suspend file with the parameters defined in the file as well
- All lines that have been processed are **commented out** using the line comment character in column 1

```
BROWSE   SAZOS.USER.SUSPEND
Command ==> █
***** Top of Data *****
ÜHZSPROC/APL/MVSA SCOPE=ONLY REMOVE=NO
ÜHZSPROC/APL/MVSB SCOPE=ONLY REMOVE=NO
ÜHZSPROC/APL/MVSC SCOPE=ONLY REMOVE=NO
Ü Suspend health checker
```

© Copyright IBM Corporation 2019

5-16

### *Implement planned suspend capability*

Problem: Add resources to the policy, which are not ready to be used.

Solution: Leverage the planned suspend capabilities of so called suspend file.

Suspend file is processed during a:

- Manager COLD/WARM start.
- Configuration REFRESH, only when configuration file has changed.

Suspend requests are injected for all resources in the suspend file with the parameters defined in the file as well. All lines that have been processed are commented out using the line comment character in column 1. The result is that after the next Manager COLD/WARM start the commented out resources in the suspend file are no longer suspended. The suspend request will “survive” a Manager HOT start.

The automation manager must have RACF UPDATE access to the suspend file.

Specify the name of the resource to be processed, the SCOPE of the suspend request and whether the request is automatically removed after SYSGONE, example:

```
HZSPROC/APL/MVSB SCOPE=ONLY REMOVE=NO
```

# Suspended resources

If suspend request succeeds and automation status is idle:

- Automation manager suspend flag is set to suspended
- Agent automation flag turned off

Implications:

- INGSET/INGAUTO can not be used for suspended resources
  - Except to set automation status to IDLE
- Suspended group member behaves like a passive member
  - It can not be selected by the group
- Suspended group behaves like a passive group
  - No votes are generated
  - No members can be selected
- A suspended MTR does not monitor anymore (Timer is cancelled)
- Job Log Monitoring is stopped for suspended resources
- Still performed: OUTREP, Status Observer, special resources, message capturing, threshold handling, WLM resources management, IPL complete notification

## *Suspended resources*

If suspend request succeeds and automation status is idle:

- Automation manager suspend flag is set to suspended
- Agent automation flag turned off. To differentiate between the turned off automation agent flag or a suspended APL, you can refer to the task global variables SUB\*SUSPEND provided with AOCQRY. All automation flags (Automation, Initstart, Start, Recovery, Terminate, and Restart) are set to 'S' and that setting is also propagated down to minor resources, if applicable. While being suspended, DISPFLGS and DISPSTAT will not allow you to change the status of any of these automation flags. You must first resume the resource before you can change the status of a flag.

**Implications:**

- INGSET/INGAUTO can not be used for suspended resources
  - Except to set automation status to IDLE
- Suspended group member behaves like a passive member
  - It can not be selected by the group
- Suspended group behaves like a passive group
  - No votes are generated
  - No members can be selected
- A suspended MTR does not monitor anymore (Timer is canceled)
- Job Log Monitoring is stopped for suspended resources. To enable Job Log Monitoring again, you must do it manually using INGJLM, or make sure that the resource is first resumed and a start request or vote exists for automation to start it.
- The following capabilities stay in place, even while a resource is suspended:
  - OUTREP processing takes place to save the reply ID, if a WTOR was received that is normally handled for the resource
  - Exit routines registered at the Status Observer by a user script or registered internally by automation are called when the resource changes
  - The status of a Workload Scheduler special resource follows any status change of the resource
  - Message capturing, threshold handling, and WLM resources management are done by default
  - The resource is not exempted from IPL-complete monitoring

# Lesson 2. Additional policy definitions

## Lesson 2: Additional policy definitions

- Notify Operators
- Timers
- Spool management
- MVS automation
- Miscellaneous

### What this lesson is about

This lesson describes how to define additional non application policy items.

### Objectives

After completing this lesson, you should be able to define additional policy items such as notify operators and timers.

### References

*SC34-2717 System Automation for z/OS Version 4  
Release 1 Defining Automation Policy*

## Notify operators

- Notify Operators receive copies of SA z/OS messages based on selectable message classes
  - The messages can be held on the operator screen based on message type
  - Can be logged-on operators and automation tasks
- You use the NOTIFY OPERATORS entry type (NFY, option 38)

### Notify operators

You can use the NOTIFY OPERATORS policy to specify the names of NetView operators who receive copies of messages that are issued by SA z/OS. Most SA z/OS messages are issued by automation tasks (autotasks) and are seen only by operators who are defined as *Notify Operators*.

SA z/OS assigns one or more internal message classes to each message that is issued. You can identify which classes of messages each operator can receive and which types of messages to hold on the operator screen. In this case, the Notify messages can be displayed and held on the operator screen. Use INGNTFY to manage Notify Operators from the automation agent.

# Defining Notify Operators

| Operator Notification                 |             |                      |                               | Row 1 to 17 of 21<br>SCROLL==> CSR                            |           |           |        |        |
|---------------------------------------|-------------|----------------------|-------------------------------|---|-----------|-----------|--------|--------|
|                                       |             |                      |                               | Define operator and types of messages to hold on screen.      |           |           |        |        |
| Entry Type : Notify Operators         |             |                      |                               | PolicyDB Name : OPSDEMOCOPY                                   |           |           |        |        |
| Entry Name : NOTIFY_OPERATORS         |             |                      |                               | Enterprise Name : OPSDEMOCOPY                                 |           |           |        |        |
| Line Commands: S (Select), D (Delete) |             |                      |                               | Select an operator to define their message class assignments. |           |           |        |        |
| Cmd                                   | Operator ID | Operator Description | Notify (Yes/No)               | ---   | Message I | Message E | Type D | Type W |
|                                       | SYSOP       | MVS CONSOLE          | YES                           | —   | —         | —         | —      | —      |
|                                       | MASTER1     | Shift leader         | YES                           | —   | —         | —         | —      | —      |
|                                       | MASTER2     | First Operator       | YES                           | —   | —         | —         | —      | —      |
|                                       |             |                      | —                             | —   | —         | —         | —      | —      |
| AOFGXMX4N                             |             |                      |                               | Notify Operator Message Classes                               |           |           |        |        |
| Command ==>                           |             |                      |                               | Row 1 to 18 of 25<br>SCROLL==> CSR                            |           |           |        |        |
| Entry Type : Notify Operators         |             |                      |                               | PolicyDB Name : OPSDEMOCOPY                                   |           |           |        |        |
| Entry Name : NOTIFY_OPERATORS         |             |                      |                               | Enterprise Name : OPSDEMOCOPY                                 |           |           |        |        |
| OperatorID : MASTER1                  |             |                      |                               | Description : Shift leader                                    |           |           |        |        |
| Cmd                                   | Status      | Class                | Description                   | ---   | ---       | ---       | ---    | ---    |
|                                       |             | 0                    | Errors                        |   |           |           |        |        |
|                                       |             | 1                    | Debug                         |   |           |           |        |        |
|                                       | SELECTED    | 40                   | All agent automation messages |   |           |           |        |        |
|                                       |             | 41                   | No longer used                |   |           |           |        |        |
|                                       |             | 42                   | No longer used                |   |           |           |        |        |
|                                       |             | 43                   | Information                   |   |           |           |        |        |
|                                       |             | 44                   | Action                        |   |           |           |        |        |
|                                       |             | 45                   | Status                        |   |           |           |        |        |
|                                       |             | 46                   | User intervention             |   |           |           |        |        |
|                                       |             | 50                   | Critical threshold exceeded   |   |           |           |        |        |

Defining Notify Operators

Notify Operator policies are defined with option **38** (NFY) of the Entry Type Selection panel. Define the operators as shown on the slide. Specifying YES in the Notify column means that the user, when logged on, receives copies of the selected messages. Specifying NO means that the user must invoke the INGNTEFY command to set the notification ON.

There is a special *pseudo operator* called SYSOP. When used, SYSOP indicates that the automation console also receives notification messages. If SYSOP is not specified, notification messages are only sent to the MCS console when none of the Notify Operators are logged on. You enter **S** as an action against an operator ID to select the message classes for that user.

For example, you might want to see all messages from the agent (class 40), or all messages from the manager (class 80), or all messages from the agent that require operator intervention (class 46). Selecting an autotask to be a Notify Operator ensures that the messages are written to the NetView log. The list of classes is in the *IBM System Automation for z/OS: Messages and Codes* manual.

## Define a timer

| AOFPITI0           |                       | Timer Definitions                                 |                               |       | Row 1 to 4 of 20<br>SCROLL==> CSR |           |
|--------------------|-----------------------|---|-------------------------------|-------|-----------------------------------|-----------|
| Command ==>        |                       | PolicyDB Name : OPSDEMOCOPY                       | Enterprise Name : OPSDEMOCOPY |       |                                   |           |
| Entry Type : Timer | Entry Name : PROD_TIM | Line Commands: S (Select), D (Delete), R (Repeat) |                               |       |                                   |           |
| Cmd                | Timer Id              | Execution Time                                    | Task                          | Clock | Catchup                           |           |
| Command text       |                       |   |                               |       |                                   |           |
| <u>MYTIMER</u>     |                       | Every weekday at 16:00:00                         | <u>MYAUTO</u>                 |       |                                   | <u>NO</u> |
| MYREXX A1 A2       |                       |   |                               |       |                                   |           |

| AOFPITI1  |                                 | Timer Definitions                   |                               |  |  |  |
|---|---------------------------------|-------------------------------------|-------------------------------|--|--|--|
| Command ==>   |                                 | PolicyDB Name : OPSDEMOCOPY         | Enterprise Name : OPSDEMOCOPY |  |  |  |
| Entry Type : Timer  | Entry Name : PROD_TIM           | Timer Id : MYTIMER                  |                               |  |  |  |
| Specify one timer definition and optional CHRON parameters.<br>Time, elapsed time and interval time may be hh:mm:ss, hh:mm or mm. |                                 |                                     |                               |  |  |  |
| Number of Days . . .  | After <u>  </u> days (1-365)    | and <u>  </u> (elapsed time).       |                               |  |  |  |
| Day of Week . . . .   | Every <u>  </u> days (1-365)    | and <u>  </u> (elapsed time).       |                               |  |  |  |
|   | On <u>  </u> (Mon-Sun)          | at <u>  </u> (time).                |                               |  |  |  |
|   | Every <u>  </u> (Mon-Sun)       | at <u>  </u> (time).                |                               |  |  |  |
|   | Every Weekday                   | at <u>  </u> (time).                |                               |  |  |  |
|   | Every Weekend                   | at <u>  </u> (time).                |                               |  |  |  |
|   | Every Day                       | at <u>  </u> (time).                |                               |  |  |  |
| Date. . . . .   | On <u>  </u> (mm/dd/yy)         | at <u>  </u> (time).                |                               |  |  |  |
| Time Only . . . . .   | At <u>  </u> (time).            |                                     |                               |  |  |  |
|   | After <u>  </u> (elapsed time). |                                     |                               |  |  |  |
|   | Every <u>  </u> (interval time) | from <u>  </u> to <u>  </u> (time). |                               |  |  |  |
| Notify . . . . .  |                                 |                                     |                               |  |  |  |
| Remark . . . . .  |                                 |                                     |                               |  |  |  |
| CHRON parameters .  |                                 |                                     |                               |  |  |  |

© Copyright IBM Corporation 2019

5-21

### Define a timer

You define timers with option **30** (TMR) of the Entry Type Selection panel (AOFGETYP). You can schedule timer events for these times:

- At set times within a weekly cycle
- At specific dates and times
- At regular intervals

If NetView is not running, the NetView timer cannot run. You must set the Catchup option on the Timers Definitions panel to YES. The timer runs, even after a system has an outage and is restored. If YES is specified for the Catchup option, the timer is saved in the NetView SAVE/RESTORE database. If NO is specified, the timer is not saved or processed if the timer expires while NetView is down. The Catchup option does not work for the EVERY timer. All timers are converted to the NetView CHRON format, including simple AT, EVERY, and AFTER timers.

In this example, the MYREXX command is scheduled under the AUTO2 autotask to be run every weekday at 4:00 p.m. The timer ID is MYTIMER.

To define a new timer, fill in the following fields on panel AOFPITI0 and press Enter:

- Timer ID
- Task, Clock, and Catchup values (optionally)
- The NetView command to be run by the timer

Use the **Task** field to specify which task the timer runs under. Enter the actual NetView task name (not the name of the automation operator). You can enter task names that the agent does not manage as automation operators. If nothing is entered, the command executes under the AUTO2 task, because this task is the one in which the timers are set when initialization happens.

If the **Catchup** field is YES, the timer runs (after the system is working again) if it was missed because of a system outage. Also, the timer is saved in the NetView Save/Restore database. If Catchup is NO, the timer is not saved and not processed if the timer expires while NetView is down. Catchup cannot be used for the EVERY timers.

If GMT is specified in the **Clock** field, the timer runs at absolute (GMT) time calculated for the local time specified when the time was entered. The timer runs regardless of the subsequent local-to-GMT difference settings on the system. If LOCAL is specified, NetView adjusts the time that the command is run to keep it relative to the new local settings.

# JES2 spool monitoring and control

| AOFGEPOL<br>Command ==>                       | Policy Selection                                    | Row 19 to 27 of 27<br>Scroll ==> CSR         |
|---|---|--|
| Entry Type : Application<br>Entry Name : JES2 | PolicyDB Name : NEW_PDB<br>Enterprise Name : SAPLEX |  |
| Action  | Policy Name   | Policy Description                           |
|   |   | -----JES2 SPECIFIC POLICY-----               |
|   | JES2 SPOOLFULL CNTL                                 | Define JES2 spool full specific data         |
|   | JES2 SPOOLFULL CMDS                                 | Define JES2 spool full commands              |
|   | JES2 SPOOLSHORT CNTL                                | Define JES2 spool short specific data        |
|   | JES2 SPOOLSHORT CMDS                                | Define JES2 spool short commands             |
|   | JES2 DRAIN CMDS                                     | Define JES2 drain commands                   |
| WHERE USED                                    |   | List application groups linked to this entry |
| COPY  |   | Copy data from an existing entry             |
| *****   |   | Bottom of data *****                         |

- Driven by spool threshold messages (\$HASP050)
  - Updates SDF display
  - Runs installation-defined commands
  - Ensures efficient processing in MASplex
- DRAIN: Specifies which JES resources are to be drained and how they are to be drained if a JES2 shutdown fails
- SPOOLSHORT: Provides for recovery from JES spool shortage condition (80%)
- SPOOLFULL: Provides for recovery from a JES spool full condition (100%)
- Spool also monitored from JES2MON MTR, setting Health Status
  - SPOOLSHORT sets WARNING.
  - SPOOLFULL sets CRITICAL.

**Note:** Do not define AUTO(N) for the \$HASP050 message

## JES2 spool monitoring and control

The SA z/OS agent contains facilities to monitor and manage spool shortage conditions. These facilities are initiated by a \$HASP050 message, indicating a spool shortage. The JES2MON MTR also sets the Health Status based on the \$HASP050 message. When you specify an Application type of JES2, you see an additional set of policy items as shown on this slide. The Application type is defined when you create a new Application policy object.

Typically, the initial \$HASP050 occurs when the spool reaches a threshold of 80% full, known as TGWARN. When the spool reaches this threshold, SA z/OS uses the SPOOLSHORT policy for these purposes:

- Set the SPOOL indicator on SDF to yellow.
- Optionally execute user commands to reduce spool usage.
- Ensure that spool recovery only runs on one processor in the sysplex.

SPOOLFULL provides a similar set of specifications that are invoked when the spool goes to 100% utilization. For example, the SPOOL indicator on SDF changes to red.

You can issue passes of commands so that more drastic measures are used while the over utilization continues. You can also avoid wasteful issuing of commands in conditions where the spool is above and below the threshold repeatedly.

# SPOOLSHORT processing

SPOOLFULL  
processing is  
similar.

```
AOFGDYNA          JES2 SPOOLSHORT Specification
Command ===> _____
```

|                          |                           |
|--------------------------|---------------------------|
| Entry Type : Application | PolicyDB Name : NEW_PDB   |
| Entry Name : JES2        | Enterprise Name : MASPLEX |

Specify system names for spool recovery priority.  
(hh:mm:ss)  
(0 to 50)  
(hh:mm:ss)

|                                     |                 |
|-------------------------------------|-----------------|
| Spool Shortage Retry Time . . . . . | <u>00:02:00</u> |
| Spool Shortage Buffer . . . . .     | <u>5</u>        |
| Spool Shortage Reset Time . . . . . | <u>00:15:00</u> |

Enter priority of systems for spool recovery:  
Spool Shortage Recovery SMFID 1 . . SYS1  
Spool Shortage Recovery SMFID 2 . . SYS2  
Spool Shortage Recovery SMFID 3 . . SYS3  
Spool Shortage Recovery SMFID 4 . . \_\_\_\_\_  
Spool Shortage Recovery SMFID 5 . . \_\_\_\_\_  
Spool Shortage Recovery SMFID 6 . . \_\_\_\_\_

Define commands and passes.  
Interval between passes is defined by Retry Time: 000001 C  
Serial

```
AOFGDYNC      Command Processing : SPOOLSHORT
Command ===> _____
```

Mixed case . . . NO (YES NO)

|                              |                                |
|------------------------------|--------------------------------|
| Cmd Ps AutoFn/* Command Text |                                |
| <u>1</u>                     | <u>MVS \$PO,Q=A,A=3</u>        |
| <u>1</u>                     | <u>MVS \$OO,Q=A,A=3,CANCEL</u> |
| <u>2</u>                     | <u>MVS \$PO,ALL,A=4</u>        |
| <u>2</u>                     | <u>MVS \$OO,ALL,A=4,CANCEL</u> |

## SPOOLSHORT processing

To avoid oscillation around the TGWARN level, spool recovery continues until the utilization is far below the TGWARN level. A *Buffer* of 5 indicates that Passes of commands are issued until the spool utilization falls 5% below TGWARN.

The *Reset Time* indicates that if spool short recovery is entered again (after reaching the buffer) within this time, the previous recovery continues from where it ended. It does not start a new recovery process from PASS1.

In a MASplex, all systems receive the \$HASP050 message at the same time. To avoid the same routines running on all systems at the same time, you can provide a priority list of systems in the MASplex. The routine is run only on the highest system in the list that is active in the sysplex. When this happens, the SDF detail record for SPOOL on the other systems indicates that recovery is occurring elsewhere. Enter YES to edit the commands that are used to reduce spool usage.

Recovery actions for spool shortage are scheduled as command passes, similar to the start and stop phases. More information about this topic is in Unit 4, [“Defining applications and building automation configuration files”](#) on page 197. The *Retry Time* is the interval between passes. Most likely, you will code JES commands to handle the SPOOLSHORT processing. You can code REXX

EXECs if you have to perform more complex processing. You can use &EHKVAR variables to pass the current utilization to your program. EHKVAR1 contains the current spool utilization, and EHKVAR2 contains the recovery target. If all passes are exhausted, on subsequent retries, all commands from all passes are run in turn.

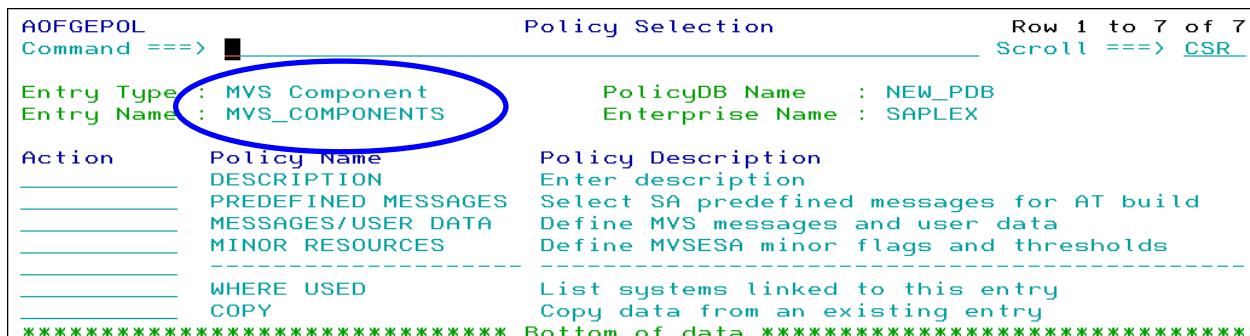
## SPOOL recovery considerations

- Avoid violent reactions
  - Use frequent, low-impact actions at first
  - Aim to reduce by a reasonable amount before starting again
- Alert operations staff for typical actions they take
- Spool recovery is controlled by recovery automation flag for JES2

*SPOOL recovery considerations* Try to make the actions you specify gradual and progressive, but frequent. Avoid violent reactions, which might remove too much and result in user dissatisfaction. SPOOL usage monitoring is controlled by the RECOVERY automation flag of the JES.

## MVS automation

- MVS Component (MVC, option 33) policy defines actions to take when MVS data sets fill: MVSDUMP, SMFDUMP, LOGREC, and SYSLOG
  - Defined as minor resources of MVS\_COMPONENTS
  - Thresholds defined in MINOR RESOURCES policy item
  - Actions defined in MESSAGES/USER DATA policy item
- Specify actions when MVS console buffers too high



© Copyright IBM Corporation 2019

5-25

### MVS automation

*MVS component automation* refers to a set of automation routines that manage the MVS data sets and console WTO buffers. For the MVS data sets, you specify installation-dependent commands that are invoked when the data sets are filling or are nearly full. For example, LOGREC corresponds to system messages about the logrec data set being full or nearly full (IFB040I, IFB060E, IFB080E, IFB081I, and IFC001I).

Most of the definitions follow a similar pattern. You can provide a set of thresholds so that alert messages can be sent to Notify Operators when the frequency rises. You can also specify automation flags to control the activation of this automation or specify a set of commands to be issued when the appropriate condition occurs. The &EHKVAR1 variable contains the data set name and other useful information. Not all of these routines are required. Some z/OS facilities simplify MVS dump data management, and some vendor products automatically handle SYSLOG and other data set offloads. The &EHKVARn variables are described in the *IBM System Automation for z/OS: Customizing and Programming* manual.

## MVSDUMP thresholds specification

| AOFGDYNA                           | Thresholds Definition                      | Line 00000001<br>Scroll ==> CSR |
|------------------------------------|--|---------------------------------|
| Command ==>                        |  |                                 |
| Entry Type : MVS Component         | PolicyDB Name : NEW_PDB                    |                                 |
| Entry Name : MVS_COMPONENTS        | Enterprise Name : SAPLEX                   |                                 |
| Resource : MVSESA.MVSDUMP          |  |                                 |
| Critical Number . . . 6            | (1 to 50)                                  |                                 |
| Critical Interval . . . 00:20:00   | (hh:mm:ss or hhmmss, 00:00:01 to 24:00:00) |                                 |
| Frequent Number . . . 4            | (1 to 50)                                  |                                 |
| Frequent Interval . . . 00:20:00   | (hh:mm:ss or hhmmss, 00:00:01 to 24:00:00) |                                 |
| Infrequent Number . . . 2          | (1 to 50)                                  |                                 |
| Infrequent Interval . . . 00:20:00 | (hh:mm:ss or hhmmss, 00:00:01 to 24:00:00) |                                 |

- MVSESA.MVSDUMP MINOR RESOURCES thresholds
- Actions defined under MESSAGES/USER DATA policy
  - Message ID (Pseudo) is MVSDUMP
  - Commands for INFR, FREQ, and CRIT

### MVSDUMP thresholds specification

On this slide, you see an example of the MVSESA.MVSDUMP minor resource thresholds that are defined with the MVS Component MINOR RESOURCES policy item. The MESSAGES/USER DATA policy item for message ID MVSDUMP defines the commands to issue for each minor resource threshold:

- **INFR:** MVS DD ALLOC=ACTIVE
- **FREQ:** MVS DD ALLOC=INACTIVE
- **CRIT:** MVS DD ALLOC=INACTIVE

## MVS component: MVSDUMPTAKEN message

| AOFGDYNM  |                                   | Message Processing | Line 00000001 Col 001 075 |
|---|-----------------------------------|--------------------|---------------------------|
| Command ==> _____   |                                   | Scroll ==> CSR     |                           |
| Entry Type : MVS Component  | PolicyDB Name : NEW_PDB           |                    |                           |
| Entry Name : MVS_COMPONENTS   | Enterprise Name : SAPLEX          |                    |                           |
| Line Commands: S/C (Cmd), R (Rep), K (Cod), U (Usr), A,M (AT, MRT, MPF) |                                   |                    |                           |
| I, D (insert or delete lines)   |                                   |                    |                           |
| Message ID field length . 16 (1 - 32)                                   |                                   |                    |                           |
| <b>Cmd Message ID</b>   | <b>Description</b>                | <b>C</b>           | <b>R</b>                  |
| IEA793A   | No dump datasets available        | 1                  |                           |
| IEC613A   | Unable to position a tape         | 1                  |                           |
| IEE767A   | Buffer shortage on system log     | 1                  |                           |
| IEE768I   | Buffer on system log relieved     | 1                  |                           |
| IEF824E   | Catch WTORs after console failure | 1                  |                           |
| IEF238D   | Cannot complete device allocation | 4                  | P                         |
| IEF402I   | Job failed                        | 8                  | P                         |
| IEF450I   | Job abended                       | 8                  | P                         |
| LOGREC  | LOGREC commands                   | 3                  | P P P                     |
| MVSDUMP   | MVSDUMP Threshold Commands        | 3                  | P P P                     |
| MVSDUMPRESET  | MVSDUMP Reset Commands            | 1                  | P P P                     |
| <b>S MVSDUMPTAKEN</b>   | <b>MVSDUMPTAKEN commands</b>      | 1                  | P P P                     |
| SMFDUMP   | SMFDUMP commands                  | 1                  | P P P                     |
| SYSLOG  | SYSLOG commands                   | 1                  | P P P                     |
| WTORS   | Classification of WTORs           | 1                  | P P P                     |

| AOFGDYNM          |                           | Command Processing : MVSDUMPTAKEN | Line 00000001 Col 001 075 |
|-------------------|---------------------------|-----------------------------------|---------------------------|
| Command ==> _____ |                           | Scroll ==> CSR                    |                           |
| Mixed case . . .  | <u>NO</u> (YES NO)        |                                   |                           |
| Cmd Ps>Select     | AutoFn/* Command Text     |                                   |                           |
|                   | MVS DD CLEAR,DSN=&EHKVAR1 |                                   |                           |
| *****             | Bottom of data *****      |                                   |                           |

Period is required

© Copyright IBM Corporation 2019

5-27

### MVS component: MVSDUMPTAKEN message

You can use the MESSAGES/USER DATA policy item to specify message IDs, including special ones, like MVSDUMPTAKEN, MVSDUMPRESET, MVSDUMP, SMFDUMP, SYSLOG, LOGREC, and so on.

On this panel, you can define how SA z/OS reacts to an issued message. Message processing provided here is the same as that provided for Applications. The messages can be predefined strings, such as MVSDUMPTAKEN, or an actual message ID.

The actions taken are defined as commands that are run when special messages occur. For example, the commands for the special message SYSLOG identify commands to be run when a SYSLOG data set is created on spool. The MVSDUMPTAKEN message issues an MVS DD CLEAR,DSN=&EHKVAR1. command. Automation flags that are used to control these actions are defined in the MVS Component MINOR RESOURCES policy item for the same system.

## MVSDUMPTAKEN automation flags

| Automation Flag Processing                      |                |                          |       |              |
|---|----------------|--------------------------|-------|--------------|
| Command ==> _____                               |                |                          |       |              |
| Entry Type : MVS Component                      |                | PolicyDB Name : NEW_PDB  |       |              |
| Entry Name : MVS_COMPONENTS                     |                | Enterprise Name : SAPLEX |       |              |
| Resource : MVSESA.MVSDUMPTAKEN                  |                |                          |       |              |
| Line Commands: Exi (Exits), Dis (Disable Times) |                |                          |       |              |
| Automation Level: YES, NO, LOG, EXITS           |                |                          |       |              |
| Cmd   | Flag           | Auto                     | Exits | DisableTimes |
| <u>dis</u>                                      | Automation (A) | _____                    |       |              |
|   | Recovery (R)   | _____                    |       |              |

Enter **dis** to set  
disable times.

| Command ==> _____   |          |       |       |  | Scroll |
|---|----------|-------|-------|--|--------|
| Automation Flag: AUTOMATION Resource: MVSESA.MVSDUMPTAKEN |          |       |       |  |        |
| Cmd   | Day      | Begin | End   |  |        |
|   | SATURDAY | 00:00 | 23:50 |  |        |

Example: Disable MVSDUMPTAKEN automation all day on Saturdays

### MVSDUMPTAKEN automation flags

The flag settings that control the *MVS Component* automation are set by selecting the MINOR RESOURCES policy item. This example disables MVSDUMPTAKEN processing on Saturdays.

You can use the option to run installation exits when you must determine a flag setting. You can set the flag to be dynamic, and automation can be done more selectively. You can set a number of exits for any flag. Use the **exi** command on panel AOFGAFL2 to specify exit names.

You enter **dis** in the cmd column to set disable times for the automation or recovery flags. This causes the Disable Time Processing panel to be displayed. You can view or modify the disable days and times from that panel. Similar to the automation and assist flags, only the *Auto* and *Recover* flags can be selected when working with MVS components.

## MVS console buffer recovery

Two-stage process:

- Stage 1:
  - Extend buffer limit by 20% or to 2500
  - Change console characteristics (K S,DEL=RD)
- Stage 2:
  - Cancel jobs creating WTOs and WTORs
  - Installation filter for cancellable jobs

Routines can handle situations where console buffer queues become too high. The automation routines determine which console has the queue. The process to recover the console has two stages. In the first stage, the console buffer queue is extended, and an MVS CONTROL command (K S,DEL=RD) is issued. This action attempts to relieve any problems that might exist at the console.

## MVS console buffer recovery (continued)

© Copyright IBM Corporation 2019

5-30

This automation policy is defined to cancel jobs if they are creating too many WTOs or WTORs. Create a MESSAGES/USER DATA policy entry for the special message **WTOBUF**. Then, create CODE settings for it. In this example, CODE1 is the job name, and CODE2 is the message type (WTO or WTOR). The Value Returned indicates whether the job can be canceled.

In this example, the following information applies:

- If JOB1 is creating many WTOs (but not WTORs), it can be canceled.
  - Any job that has a job name with a prefix TEST can be canceled if it is issuing many WTOs.
  - By default, all other jobs remain because CODE1=\* and CODE2=\*.

You can disable the WTO buffer recovery by setting the recovery flag to N, as shown. Often the default flag settings are sufficient to create the effective setting of this flag.

# Automation operators policy

| AOFGENAM                          | Entry Name Selection     | Row 1 to 3 of 3              |
|-----------------------------------|--------------------------|------------------------------|
| Command ==>                       |                          | Scroll ==> CSR               |
| Entry Type : Automation Operators | PolicyDB Name : NEW_PDB  |                              |
|                                   | Enterprise Name : SAPLEX |                              |
| Action                            | Entry Name               | Short Description            |
|                                   | BASE_AUTOOPS             | Default Automation Operators |
|                                   | GATEWAY_AUTOOPS          | Gateway Automation Operator  |
|                                   | WORK_AUTOOPS             | Automation Work Operators    |
| ***** Bottom of data *****        |                          |                              |

- Include standard set in each system
  - Customize task names for your environment
  - Customize GATEWAY\_AUTOOPS if you run gateways
- If you add your own autotasks, SA z/OS automatically starts and recovers them
- The sample \*BASE policy contains BASE\_AUTOOPS, GATEWAY\_AUTOOPS, and WORK\_AUTOOPS OPS
- The CMD\_RECEIVER\_AUTOOPS is defined in the \*IBMCOMP add-on policy

## Automation operators policy

Automation operators represent the autotasks that are used when automation processing occurs. You define the automation operators by using the customization dialog option 37 (AOP). The standard set of operators must be specified for all systems. This set is defined in the sample system definitions that are included with the product. However, each system might need some tailoring. The Gateway operator (GATOPER) is a unique task in each system and might require specific definitions for each system. Special console definitions for MVS CONSi might be unique to each system.

Internally, within NetView, the SA z/OS agent processing is spread across many NetView automation tasks (autotasks) to provide a high degree of parallel activity and ensure that messages from MVS and other events are processed in the correct sequence. You can also add your own autotask definitions so that SA z/OS starts, monitors, and recovers them.

Several base automation operators are provided with the \*BASE sample PDB. The CMD\_RECEIVER\_AUTOOPS definition which is provided in the add-on policy \*IBMCOMP is required for the command receiver function.

# Base automation operators policy

| AOFGDYN0 Automation Operator Definitions   |           |                  | Line 00000001 Col 001 075 |
|--|-----------|------------------|---------------------------|
| Command ==> _____  |           |                  | Scroll ==> CSR            |
| Entry Type : Automation Operators PolicyDB Name : NEW_PDB  |           |                  |                           |
| Entry Name : BASE_AUTOOPS Enterprise Name : SAPLEX   |           |                  |                           |
| Line Commands: S (Select), M (Move), B (Before), A (After), R (Repeat)<br>I (Insert lines), D (Delete lines) |           |                  |                           |
| Cmd  | AutoFunc  | Primary Operator | Backup Operator           |
| INITOPR1   | AUTINIT1  |                  |                           |
| INITOPR2   | AUTINIT2  |                  |                           |
| ALRTOPER   | AUTALERT  |                  |                           |
| BASEOPER   | AUTBASE   |                  |                           |
| GSSOPER  | AUTGSS    |                  |                           |
| JESOPER  | AUTJES    |                  |                           |
| LOGOPER  | AUTLOG    |                  |                           |
| MONOPER  | AUTMON    |                  |                           |
| MSGOPER  | AUTMSG    |                  |                           |
| MVSCONS  | AUTCON    |                  |                           |
| NETOPER  | AUTNET1   | AUTNET2          |                           |
| PPIOPER  | AOFARCAT  |                  |                           |
| RECOPER  | AUTREC    |                  | IEA*, IEE*, IOS*          |
| RPCOPER  | AUTRPC    |                  |                           |
| SHUTOOPER  | AUTSHUT   |                  |                           |
| SYSOPER  | AUTSYS    |                  | IEE400I, IEE600I          |
| XCFOPER  | AUTXCF    |                  |                           |
| XCFOPER2   | AUTXCF2   |                  |                           |
| FA-FIELD   | FA-COMMIT | FA-END           | FA-DEFUN                  |
|  |           |                  | FA-PERIOD                 |
|  |           |                  | FA-PERIODIC               |
|  |           |                  | FA-PERFORMANCE            |

Automated function name  
equates to a common global  
variable whose value is the  
*automation operator name*.

Automation operators are started when agent initializes, and restarted if they fail

## Base automation operators policy

On this slide is an example of the BASE\_AUTOOPS policy item. Each entry is an *automated function* that represents one or more *automation operators*. Automation operators are the autotasks that perform the automation. Automation operators are NetView tasks that are defined in xxxOPF or SAF. The automation operators are grouped together based on function, called automated functions, with names like JESOPER, RECOPER, or NETOPER. Each of the automation operators is like a member of an operations team with specific responsibilities. For example, the RECOPER handles recovery actions, the NETOPER handles network functions, and so on.

The automation function name is used to build a common global variable. The value of the common global variable is the automation operator. Automation can be routed to an automated function instead of the automation operator, which might be different from installation to installation.

For example, SA z/OS can route actions to MONOPER (automated function name). MONOPER might be AUTMON (the default automation operator name) or MONAUTO if you customized the AOP policy. Typically, the defaults provided by IBM are used.

Some of the operators have messages assigned to them. For example, all IEA\* messages, unless otherwise indicated, are assigned to the RECOPER operator. Automated functions are vital to SA z/OS operation. Special processing is performed for them as follows:

- When the automation agent starts, each defined automated function autotask is started.
- If a defined autotask fails, it is automatically restarted. (Autotasks rarely fail, however.)
- If an autotask fails permanently, another defined automation task takes over. The BASEOPER is the last fall back option for all other automated functions.

The complete set of SA z/OS automated functions is documented in *IBM System Automation for z/OS Defining Automation Policy*.

# Important operators

| Function Name   | Description  |
|-----------------|--|
| <b>MONOPER</b>  | Overall monitoring                                       |
| <b>GATOPER</b>  | Gateway processes  |
| <b>RECOPER</b>  | Recovery actions   |
| <b>AOFCMDnn</b> | Command Receiver Autoops                                 |
| <b>AOFWRKn</b>  | Work operators used to start, stop, and process messages |
| <b>SYSOPER</b>  | Selected MVS messages                                    |
| <b>LOGOPER</b>  | Default message handlers                                 |
| <b>PLEXOPER</b> | Operator that handles sysplex automation requests        |
| <b>E2EOPER</b>  | Operator that handles requests from end-to-end manager   |
| <b>RPCOPER</b>  | Operator that handles XCF requests                       |
| <b>EVTOPER</b>  | Operator that handles processing for Inform List policy  |
| <b>BASEOPER</b> | Backup operator  |

© Copyright IBM Corporation 2019

5-33

## Important operators

This slide shows some of the many important operators and their roles. These automated functions and operators are defined in the automation policy under option 37 (AOP).



**Hint:** You can use the NetView QRYGLOBL command to display the common global variables (automated functions) and their values (automation operators):

```
QRYGLOBL COMMON VARS=AOF.0AUTOOPS.*
```

## Work operators (AOFWRKxx)

- The samples include 20 work operators.
  - AUTWRK01 - AUTWRK20
  - SA z/OS does not initialize unless a minimum of three AOFWRKxx operators are defined
- All activity is directed to a work automation task (autotask)
  - Starting
  - Stopping
  - Message processing
- Operators are assigned at ACF load time
- Allocating more work operator tasks might improve overall SA z/OS performance, especially during system start and shutdown

© Copyright IBM Corporation 2019

5-34

### Work operators (AOFWRKxx)

The automation agent uses work operators to manage the starting, stopping, and monitoring of Applications. By default, there are twenty work operators provided in the \*BASE sample PDB.



**Note:** A minimum of three work operators is required. Fewer than three causes a warning message to be issued. AUTINIT1 and AUTINIT2 are also important during initialization and ACF refresh.

The maximum number of work operators that you define must handle a system initialization or shutdown. In those environments, a subset of the work operators are typically idle. At ACF load time, each Application is associated with a work operator. This selection is for spreading the workload at any time over all available tasks. After the association is made, it is fixed until the ACF is reloaded.

Processing of ACTIVMSG, TERMMMSG, and other generic routines for an Application is routed to its associated work operator. All requests for any one subsystem are serialized. Do not route REXX EXECs to the work operators for processing. The EXECs can prevent the work operators from

processing important messages if the EXECs wait and trap messages, for example. In that case, you route the EXEC to another task to prevent the work operator from being busy.

## Message assignment

- ALL messages from a subsystem are assigned to the designated work operator, serializing all processing for the subsystem
- Messages are assigned with the NetView ASSIGN command
- Additional message IDs can be directed to the same task as the one that is used by a specific subsystem

### Message assignment

The SA z/OS agent uses the NetView ASSIGN command to route unsolicited messages to specific operators. All messages from a subsystem are directed to one designated work operator. The designation of the work operator and the assignment occur at the time that the ACF is loaded. All commands to the subsystem are issued by the work operator, and all messages from the subsystem are assigned to that same work operator. This practice helps to ensure serialization of automation actions for that subsystem. You can also make a specification so that any message in the system that has a specific prefix is routed to the work operator of a designated Application.

# Routing specific messages

| AOFGDYNA                         | Application Information                | Line 00000024<br>Scroll ==> CSR |
|----------------------------------|--|---------------------------------|
| Command ==>                      |  |                                 |
| Entry Type : Application         | PolicyDB Name : NEW_PDB                |                                 |
| Entry Name : VTAM                | Enterprise Name : SAPLEX               |                                 |
| Startup Parameters . . . . .     | ... (LIST=00)                          |                                 |
| External Shutdown . . . . .      | (FINAL ALWAYS NEVER)                   |                                 |
| Shutdown Pass Interval . . . . . | (hh:mm:ss)                             |                                 |
| Cleanup Delay . . . . .          | (hh:mm:ss)                             |                                 |
| Command Prefix . . . . .         |  |                                 |
| Message Prefix . . . . .         | IST                                    |                                 |
| Sysname . . . . .                |  |                                 |
| Monitor Routine . . . . .        | (name NONE)                            |                                 |
| Monitor Interval . . . . .       | (hh:mm NONE)                           |                                 |
| Inform List . . . . .            | (SDF EIF E2E IOM ITM SMF TTT USR NONE) |                                 |
| ARM Element Name . . . . .       |  |                                 |
| WLM Resource Name 1 . . . . .    |  |                                 |
| WLM Resource Name 2 . . . . .    |  |                                 |
| WLM Resource Name 3 . . . . .    |  |                                 |
| Owner . . . . .                  |  |                                 |
| Info Link . . . . .              |  |                                 |
| Runtokens . . . . .              |  |                                 |

© Copyright IBM Corporation 2019

5-36

## Routing specific messages

On this slide is a way to specify the routing of messages for an Application to its work autotask. In the APPLICATION INFO policy item for VTAM, you can specify one or more message prefixes (in this case, IST). Any messages with this prefix, from any address space, are assigned to the same work autotask as that used for the VTAM Application.

## Notifying the MVS console

```

AOFPIEC0                               Automation Console
Command ==> _____
Entry Type : System
Entry Name : MVS
Console Name . . . . .
Select the route codes:
A Code      Definition
--- All Display ALL messages
* 1 Master console action
* 2 Master console information
--- Tape pool
--- Direct access pool
--- Tape library
--- Disk library
--- Unit record pool
--- Teleprocessing control
--- System security
--- System error or maintenance
--- Programmer information
--- Emulators
--- User
--- User
--- User
Select the MCSFLAGS:
A Flag      Definition
More: +
--- 1 Send conditionally to console (REGO)
--- 2 Send unconditionally (QREGO)
--- 3 Immediate command response (RESP)
--- 4 Reply to a WTOR (REPLY)
--- 5 Broadcast to all active consoles (BRDCST)
--- 6 Send to hard copy only (HDRCPY)
--- 7 Do not append time to the message (NOTIME)
--- 8 No hardcopy if issued from supervisor state (NOCPY)

```

AUTOMATION CONSOLE policy item in SYStem policy defines the following for SA z/OS messages:

- Console name
- Routecodes
- MCSFLAGS

### Notifying the MVS console

You can use the AUTOMATION CONSOLE policy item of the SYStem object to specify an MVS console to which agent messages are routed when there is not a Notify Operator logged on. This situation is likely to be relevant during system initialization, before VTAM is active. It also applies if you specify the NetView pseudo-operator SYSOP as a Notify Operator.

When initializing, if SA z/OS cannot send messages to the console identified by this definition, then it broadcasts its messages to all active consoles in the sysplex. This includes all NetView users logged on throughout the sysplex. This example defines route codes 1 and 2 and MCSFLAGS 1 for the MASTMVSA console.

## MVSCONS automated operator

```
A0FGDYN7          Automation Operator Definitions Processing : line 10
Command ==> _____ Scroll ==> CSR

This panel displays the complete input fields of a single data line.

Automated Function . . . . . MVSCONS
Primary Operator . . . . . AUTCON
Backup Operator . . . . . _____
Messages for this Operator .
MVS Console Name . . . . . MASTMVSA
NetView Operator 1 . . . . . _____
NetView Operator 2 . . . . . _____
NetView Operator 3 . . . . . _____
NetView Operator 4 . . . . . _____
NetView Operator 5 . . . . . _____
NetView Operator 6 . . . . . _____
```

- Select Auto Operators (AOP, option 37).
- Select BASE\_AUTOOPS.
- Add MVSCONS.
- Define MASTMVSA console to be associated to the AUTCON operator.

### MVSCONS automated operator

Select the Auto Operators policy object (AOP, option 37). Select one of the policy objects, such as BASE\_AUTOOPS. Select the OPERATORS policy item, and then ADD an automation operator with a name MVSCONSi, where *i* represents:

- A through Z
- 0 through 9
- Blank (used in this example)

You can use a name that is relevant for the system, or \*MASTER\* (the current master console), or \*ANY\* (any console, meaning that this autotask serves every console). A console can be assigned for each autotask.

## Commands from MVS consoles

SA z/OS Automation Agent automatically checks for physical MCS consoles on a system

- For each console, an autotask is started so that the operator can enter NetView commands
- The SA z/OS agent must be initialized
- The function is not available for the Hardware Management Console (HMC)

### *Commands from MVS consoles*

A standard SA z/OS agent feature regularly checks for all consoles on the host system so that operators at physical MVS consoles can enter commands to NetView or SA z/OS. It then ensures that each physical console has an associated autotask. Commands can then be entered from each console. If a physical console is on the system, these specifications ensure that line-mode commands such as DISPSTAT and INGLIST can always be entered.

However, SA z/OS must be initialized. Problems that occur when commands can only be entered from consoles are typically related to initialization. Ensure that at least one console has such facilities as soon as NetView starts by using the appropriate coding in the NetView CNMSTYLE.

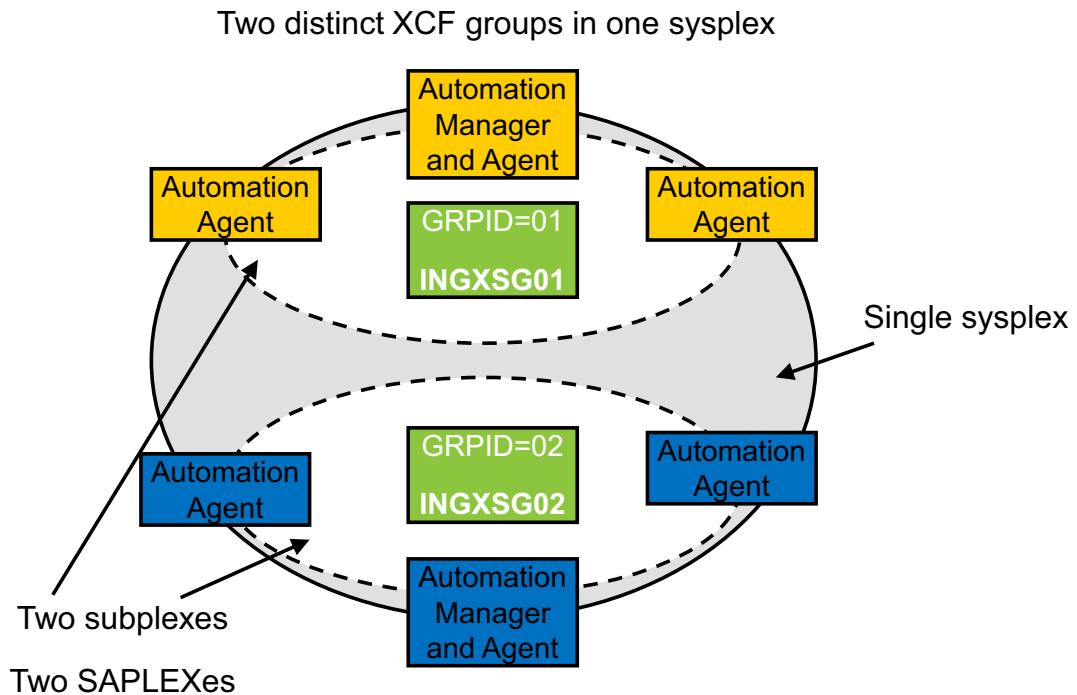
The NetView SSI must be running to use the command prefix. If it is not, you can use the MVS MODIFY command, for example, **MVS F AUTOSSI,DISPSTAT**. You must have an autotask associated with that MCS console.

The automatic association only happens for consoles that are physically attached to this image. It does not include the Hardware Management Console (which is strictly called the System Console). However, you can associate this console to an autotask using CNMSTYLE or by programming an exit, such as AOFEXDEF.

You can also use the AUTOTASK statement in CNMSTYLE to define the console:

- AUTOTASK.?Master.Console = \*MASTER\*
- AUTOTASK.?Master.Console = \*ANY\*

## Sysplex with subplex example



© Copyright IBM Corporation 2019

5-40

### Sysplex with subplex example

You can divide your real sysplexes into several logical SA z/OS subplexes. You must define a specific XCF group suffix and a specific group policy object for each subplex. Each SA z/OS subplex must have its own automation manager running. Using SA z/OS subplexes, you can run the automation on systems of sysplexes as on single systems. SA z/OS subplexes are required if you do not have shared disks for all your systems in the sysplex. A practical example is the separation into a test and production sysplex.

In this example, two XCF groups are in use: INGXSG01 and INGXSG02. The group is defined in the INGXPDST task initialization member, INGXINIT. The group must match the XCF group ID defined in the automation manager HSAPRMxx member.

# Sysplex policy items

| AOFGEPOL                              |                     | Policy Selection                           | Row 1 to 20 of 20 |
|---------------------------------------|---------------------|--|-------------------|
| Command ==>                           |                     |  | Scroll ==> CSR    |
| Entry Type                            | : Group             | PolicyDB Name                              | : OPSDEMO         |
| Entry Name                            | : SYSPLEX1          | Enterprise Name                            | : OPSDEMO         |
| Action                                | Policy Name         | Policy Description                         |                   |
| DESCRIPTION                           |                     | Enter description                          |                   |
| GROUP INFO                            |                     | Define group information                   |                   |
| SUBGROUPS                             |                     | Select subgroups for group                 |                   |
| SYSTEMS                               |                     | Select systems for group                   |                   |
| -----SYSPLEX SPECIFIC POLICY-----     |                     |  |                   |
| S                                     | SYSPLEX             | Define sysplex policy                      |                   |
|                                       | APPLICATION GROUPS  | Select application groups for sysplex      |                   |
|                                       | SYSPLEX DEFAULTS    | Select sysplex resource defaults for group |                   |
|                                       | DOMAINS             | Select domains for group                   |                   |
| -----LOCAL PAGE DATA SET POLICY-----  |                     |  |                   |
|                                       | LOCAL PAGE DATA SET | Define local page data set recovery        |                   |
|                                       | JOB DEFINITIONS     | Define handling of jobs                    |                   |
| -----LONG RUNNING ENQUEUE POLICY----- |                     |  |                   |
| JOB/ASID                              | A0FGXC4E            | Sysplex ApplGroups for Sysplex             |                   |
| COMMAND                               | Command ==>         |  |                   |
| COMMAND                               |                     |  |                   |
| RESOURCE                              |                     |  |                   |
| RECOVERY                              |                     |  |                   |
| COPY                                  |                     |  |                   |
| *****                                 |                     |  |                   |
| Action                                | Status              | PolicyDB Name                              |                   |
|                                       | SELECTED            | Enterprise Name                            |                   |
|                                       | SELECTED            |  |                   |
|                                       | SELECTED            |  |                   |
|                                       | SELECTED            |  |                   |
|                                       |                     | Sysplex ApplGroup                          |                   |
|                                       |                     | AM_X                                       |                   |
|                                       |                     | RV_BASIC                                   |                   |
|                                       |                     | RV_MOVE                                    |                   |
|                                       |                     | RVPLEX                                     |                   |

Four sysplex Application Groups are included in SYSPLEX1

© Copyright IBM Corporation 2019

5-41

## Sysplex policy items

Use the Group policy object (GRP, option 2) to define a sysplex. On this slide is an illustration of the sysplex policy items for SYSPLEX1. A sysplex group represents the set of systems, in the same logical sysplex (same XCF group ID), that are operated together under the control of a single Primary automation manager (PAM). The group does not need to include all systems in the actual sysplex. Selecting the SYSTEMS policy item displays the systems that are part of this sysplex.

Selecting the APPLICATION GROUPS policy item displays the sysplex Application Groups that are part of this sysplex. In this case, three sysplex Application Groups are selected for the SYSPLEX1 sysplex entry. You can define default policy for sysplex Application Groups by using the XDF (Sysplex Defaults, option 40) default policy; for example, Inform List. In the case of GDPS, define a STANDARD group and select all GDPS production and controlling systems and any dummy systems.

# Lesson 3. Defining Service Periods, Events, Triggers and Pacing Gates

## Lesson 3: Defining Service Periods, Events, Triggers, and Pacing Gates

- Service Periods (option 8, SVP):
  - Defines windows for the availability of resources
  - Creates MakeAvailable and MakeUnavailable requests at start of window and removes them at end of window
- Events (option 7, EVT):
  - Events are conditions that occur that can be recognized and marked as set in the NetView AT or by an operator
  - Events are set with the INGEVENT command
- Triggers (option 9, TRG):
  - Used in combination with Events
  - Used to control the starting and stopping of applications
- Pacing Gates (option 13, PAC):
  - Specify the number of applications which are allowed to be started or stopped concurrently

© Copyright IBM Corporation 2019

5-42

### What this lesson is about

This lesson describes how to define service period, event, and trigger and pacing gate policy definitions.

### Objectives

After completing this lesson, you should be able to define service period, event, and trigger and pacing gate policy definitions.

### References

*SC34-2717 System Automation for z/OS Version 4  
Release 1 Defining Automation Policy*

You can use *Service Periods* and *Triggers* to control the starting and stopping of resources. *Service Periods* are based on a time schedule. *Triggers* work in conjunction with *Events*.

An *Event* is a switch that is set when some condition is detected. Typical examples include a housekeeping job for an online Application ending successfully or a monitoring routine indicating that there is enough free DASD space for an Application to proceed.

You define Service Periods with option **8** (SVP), and Events with option **7** (EVT) of the Entry Type Selection panel. You define Triggers with option **9** (TRG).

You define Pacing Gates with option 13 (PAC) of the Entry Type Selection panel: Specify the number of applications which are allowed to be started or stopped concurrently

## Define a Service Period

| AOFGEPOL   |                | Policy Selection        |                          | Row 1 of 5           |       |       |       |       |      |    |      |    |      |    |          |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |           |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |             |       |       |                        |       |       |       |       |       |   |       |       |                          |       |       |       |       |       |            |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |
|--|----------------|-------------------------|--------------------------|----------------------|-------|-------|-------|-------|------|----|------|----|------|----|----------|-------|-------|-------|-------|-------|-------|-------|-------|---|-------|-------|-------|-------|-------|-------|-------|-------|-----------|-------|-------|-------|-------|-------|-------|-------|-------|---|-------|-------|-------|-------|-------|-------|-------|-------|-------------|-------|-------|------------------------|-------|-------|-------|-------|-------|---|-------|-------|--------------------------|-------|-------|-------|-------|-------|------------|-------|-------|-------|-------|-------|-------|-------|-------|---|-------|-------|-------|-------|-------|-------|-------|-------|
| Command ==> _____  |                |                         |                          | SCROLL==> <u>CSR</u> |       |       |       |       |      |    |      |    |      |    |          |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |           |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |             |       |       |                        |       |       |       |       |       |   |       |       |                          |       |       |       |       |       |            |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |
| Entry Type   | Service Period | PolicyDB Name : PROD    |                          |                      |       |       |       |       |      |    |      |    |      |    |          |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |           |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |             |       |       |                        |       |       |       |       |       |   |       |       |                          |       |       |       |       |       |            |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |
| Entry Name   | RVSPV          | Enterprise Name : PLEX1 |                          |                      |       |       |       |       |      |    |      |    |      |    |          |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |           |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |             |       |       |                        |       |       |       |       |       |   |       |       |                          |       |       |       |       |       |            |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |
| Action   | Policy Name    | Policy Description      |                          |                      |       |       |       |       |      |    |      |    |      |    |          |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |           |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |             |       |       |                        |       |       |       |       |       |   |       |       |                          |       |       |       |       |       |            |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |
| S  | DESCRIPTION    | Enter description       |                          |                      |       |       |       |       |      |    |      |    |      |    |          |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |           |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |             |       |       |                        |       |       |       |       |       |   |       |       |                          |       |       |       |       |       |            |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |
|  | SERVICE WINDOW | Define service windows  |                          |                      |       |       |       |       |      |    |      |    |      |    |          |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |           |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |             |       |       |                        |       |       |       |       |       |   |       |       |                          |       |       |       |       |       |            |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |
| WHERE USED   |                | AOFGAPSP                |                          |                      |       |       |       |       |      |    |      |    |      |    |          |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |           |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |             |       |       |                        |       |       |       |       |       |   |       |       |                          |       |       |       |       |       |            |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |
| COPY   |                | Service Windows         |                          |                      |       |       |       |       |      |    |      |    |      |    |          |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |           |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |             |       |       |                        |       |       |       |       |       |   |       |       |                          |       |       |       |       |       |            |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |
| <p>Note: Review help panel for complete description of data entry fields.<br/>Valid times are 00:00-24:00, valid priorities (P) are L and H.</p> <table border="1"> <thead> <tr> <th>P</th> <th>From</th> <th>To</th> <th>From</th> <th>To</th> <th>From</th> <th>To</th> <th>From</th> <th>To</th> </tr> </thead> <tbody> <tr> <td>Monday U</td> <td>00:00</td> <td>06:00</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> </tr> <tr> <td>D</td> <td>06:00</td> <td>18:00</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> </tr> <tr> <td>Tuesday U</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> </tr> <tr> <td>D</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> </tr> <tr> <td>Wednesday U</td> <td>_____</td> <td>_____</td> <td>Up from 00:00 to 06:00</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> </tr> <tr> <td>D</td> <td>_____</td> <td>_____</td> <td>Down from 06:00 to 18:00</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> </tr> <tr> <td>Thursday U</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> </tr> <tr> <td>D</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> <td>_____</td> </tr> </tbody> </table> |                |                         |                          |                      |       | P     | From  | To    | From | To | From | To | From | To | Monday U | 00:00 | 06:00 | _____ | _____ | _____ | _____ | _____ | _____ | D | 06:00 | 18:00 | _____ | _____ | _____ | _____ | _____ | _____ | Tuesday U | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | D | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | Wednesday U | _____ | _____ | Up from 00:00 to 06:00 | _____ | _____ | _____ | _____ | _____ | D | _____ | _____ | Down from 06:00 to 18:00 | _____ | _____ | _____ | _____ | _____ | Thursday U | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | D | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ |
| P  | From           | To                      | From                     | To                   | From  | To    | From  | To    |      |    |      |    |      |    |          |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |           |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |             |       |       |                        |       |       |       |       |       |   |       |       |                          |       |       |       |       |       |            |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |
| Monday U   | 00:00          | 06:00                   | _____                    | _____                | _____ | _____ | _____ | _____ |      |    |      |    |      |    |          |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |           |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |             |       |       |                        |       |       |       |       |       |   |       |       |                          |       |       |       |       |       |            |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |
| D  | 06:00          | 18:00                   | _____                    | _____                | _____ | _____ | _____ | _____ |      |    |      |    |      |    |          |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |           |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |             |       |       |                        |       |       |       |       |       |   |       |       |                          |       |       |       |       |       |            |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |
| Tuesday U  | _____          | _____                   | _____                    | _____                | _____ | _____ | _____ | _____ |      |    |      |    |      |    |          |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |           |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |             |       |       |                        |       |       |       |       |       |   |       |       |                          |       |       |       |       |       |            |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |
| D  | _____          | _____                   | _____                    | _____                | _____ | _____ | _____ | _____ |      |    |      |    |      |    |          |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |           |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |             |       |       |                        |       |       |       |       |       |   |       |       |                          |       |       |       |       |       |            |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |
| Wednesday U  | _____          | _____                   | Up from 00:00 to 06:00   | _____                | _____ | _____ | _____ | _____ |      |    |      |    |      |    |          |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |           |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |             |       |       |                        |       |       |       |       |       |   |       |       |                          |       |       |       |       |       |            |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |
| D  | _____          | _____                   | Down from 06:00 to 18:00 | _____                | _____ | _____ | _____ | _____ |      |    |      |    |      |    |          |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |           |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |             |       |       |                        |       |       |       |       |       |   |       |       |                          |       |       |       |       |       |            |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |
| Thursday U   | _____          | _____                   | _____                    | _____                | _____ | _____ | _____ | _____ |      |    |      |    |      |    |          |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |           |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |             |       |       |                        |       |       |       |       |       |   |       |       |                          |       |       |       |       |       |            |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |
| D  | _____          | _____                   | _____                    | _____                | _____ | _____ | _____ | _____ |      |    |      |    |      |    |          |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |           |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |             |       |       |                        |       |       |       |       |       |   |       |       |                          |       |       |       |       |       |            |       |       |       |       |       |       |       |       |   |       |       |       |       |       |       |       |       |

Animated

© Copyright IBM Corporation 2019

5-43

### Define a Service Period

A Service Period (also called a *schedule*) has several windows extending over a 1-week period. Each service window represents a time interval during which an Application is up (available) or down (unavailable). Each set of up or down windows for a day has a priority that affects the priority of the votes that are created.

Select the SERVICE WINDOW policy item to define the up and down windows. You can specify up to five service windows for up and down periods within each day. Each set of windows can also have a HIGH or LOW priority (LOW is the default). There are specifications for WEEKDAY, WEEKEND, and DAILY. These specifications can simplify your Service Period definitions. You can specify continuous service over midnight by specifying a window that ends at 24:00 one day, and a similar window starting at 00:00 on the next day.

The start time of a window is when the manager creates a vote for the resource. A MakeAvailable vote is generated for an up window. A MakeUnavailable vote is generated for a down window. The vote might not be successful until some time later, or even not at all. The vote is withdrawn at the end of the window.

In the example on the slide, a LOW priority MakeAvailable vote is generated every Monday at midnight because of the up window. At 6:00 a.m. (06:00), the MakeAvailable vote is withdrawn and a MakeUnavailable vote is generated from the down window. At 6:00 p.m. (18:00), the MakeUnavailable vote is withdrawn.

You must link the Service Period to the resources that use that Service Period:

- Use the WHERE USED policy item within the Service Period definition.
- Use the SERVICE PERIOD policy item within APL, APG, or MTR policies.

# Service Period resources

| AOFGXWHU     |                | Where Used        | Row 47 of 85  |
|--------------|----------------|-------------------|---------------|
| Command ==>  |                |                   | SCROLL==> CSR |
| Entry Type : | Service Period | PolicyDB Name :   | PROD          |
| Entry Name : | RVSVP          | Enterprise Name : | PLEX1         |
| Action       | Status         | Name              | Type          |
| _____        | _____          | OMXE_GROUP        | APG           |
| _____        | _____          | RACF              | APL           |
| _____        | _____          | RESOLVER          | APL           |
| _____        | _____          | RMF               | APL           |
| _____        | _____          | RMFGAT            | APL           |
| _____        | _____          | RRS               | APL           |
| _____        | _____          | RV_GROUP          | APG           |
| _____        | _____          | RV_MOVE           | APG           |
| _____        | _____          | RV_MSTR_CLASS     | APL           |
| _____        | _____          | RV_PLEX           | APG           |
| _____        | _____          | RVCLASS           | APL           |
| _____        | _____          | RV01              | APL           |
| _____        | _____          | RV02              | APL           |
| _____        | _____          | RV03              | APL           |
| _____        | _____          | RV04              | APL           |
| _____        | _____          | RV05              | APL           |
| _____        | _____          | RV06              | APL           |
| _____        | _____          | RV07              | APL           |
| _____        | _____          | RV08              | APL           |

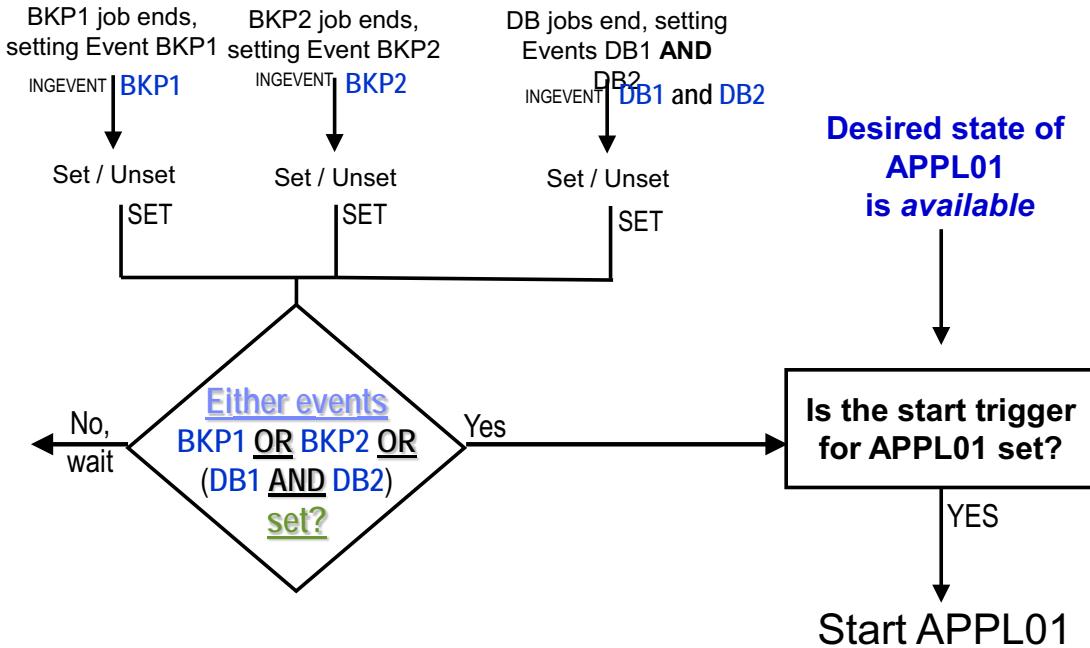
WHERE USED policy item for RVSVP Service Period

Select the Applications and press Enter to link them to the Service Period

## Service Period resources

Link the resource to the Service Period with the WHERE USED policy item. The resources inherit all up and down window specifications. In this example, Applications RV01, RV02, RV03, and RV04 are linked to the RVSVP Service Period.

## Example: One Trigger with several Events



There is one start trigger for APPL01, but several events are attached to this one trigger

### Example: One Trigger with several Events

Events are binary: they are *set* or *unset*. Events are typically set when something occurs; for example, when a database backup job is completed. Events can be set from the Automation Table or an operator action. Events are typically set by the Automation Table. The Automation Table entries are defined with the OVR action for the expected message defined in the MESSAGES/USER DATA policy item. In this case, the state is for APPL01.

Using multiple Events with different conditions equates to an OR relation. Using multiple Events within one condition equates to an AND relation. On the next several slides, you see ways to define the BKP1 Event as a start Trigger for APPL01.

## Define an Event

| Policy Selection                          |   |   | Row 1 of 5           |
|---|---|---|----------------------|
| Command ===> _____                        |   |   | SCROLL==> <u>CSR</u> |
| Entry Type : Event<br>Entry Name : BKP1EV |   | PolicyDB Name : PROD<br>Enterprise Name : PLEX1                   |                      |
| Action<br><u>S</u>                        | Policy Name<br>DESCRIPTION<br>CONDITION | Policy Description<br>Enter description<br>Define unset condition |                      |
| WHERE USED<br>COPY                        |   |   |                      |
| *****                                     |   |   |                      |

**A0FGAPEV** Unset Condition

| Command ===> _____                                      |   |
|---|---|
| Entry Type : Event<br>Entry Name : BKP1EV               | PolicyDB Name : PROD<br>Enterprise Name : PLEX1 |
| Verify the following condition:                         |   |
| Unset Condition . . . <b>START</b> (DOWN NONE START UP) |   |

Select the **Unset Condition** (START) and press PF3 to save.

BKP1EV Event: Select CONDITION and press Enter to define *unset* condition for the Event

### Define an Event

This example shows the BKP1EV Event. Select the CONDITION policy item and press Enter to define the *unset* condition for the Event. All that is required for an Event is a description and an automatic unset value. In this case, the BKP1EV Event is unset when the Application is started. The following values for the unset condition are valid:

- DOWN (**D**) to unset the Event when the stop is complete
- NONE (**N**) to avoid any automatic unsets
- START (**S**) to unset the Event when the start is initiated
- UP (**U**) to unset the Event when the start is complete

# Define a Trigger

| AOFGEPOL   |   | Policy Selection   |                                       | Row 1 of 5  |
|--|---|--|---------------------------------------|---|
| Command ==> _____  |   |  |                                       | SCROLL==> CSR   |
| Entry Type : Trigger<br>Entry Name : APPL01TR  |   | PolicyDB Name : PROD<br>Enterprise Name : PLEX1                    |                                       |   |
| Action<br><b>S</b><br>-----<br>WHERE USED<br>COPY                                      | Policy Name<br>DESCRIPTION<br>CONDITION | A0FGAPTO   | Command ==> _____                     |   |
| *****  |   | Entry Type : Trigger<br>Entry Name : APPL01TR                      |                                       | linked to this entry  |
| *****  |   | Action<br><b>S</b><br>-----<br>STARTUP 1<br>STARTUP 2<br>STARTUP 3 | Condition                             | entry   |
| *****  |   | *****  |                                       | *****   |
| <b>APPL01 Trigger:</b><br>Select<br>CONDITION to<br>define start or<br>stop conditions |   | A0FGAPTO   |                                       | Row 1 of 2  |
|  |   | STARTUP Condition for Trigger                                      |                                       | SCROLL==> CSR   |
|  |   | Entry Type : Trigger<br>Entry Name : APPL01TR                      |                                       | PolicyDB Name : PROD<br>Enterprise Name : PLEX1   |
|  |   | Description. . .   |                                       |   |
|  |   | Action<br>-----<br>SELECTED  | Status<br>-----<br>BKP1EV<br>CICS01EV | Event<br>-----<br>BKP1 event - unset when APPL01 starts<br>CICS01 Event - Unset when CICS01 started |
|  |   | *****  |                                       | Bottom of data *****  |

Select **STARTUP 1**  
to define the first  
start condition

Select the BKP1EV Event to satisfy the Trigger. Press PF3 to save

## Define a Trigger

A Trigger is some combination of Events. A Trigger can be a start (condition=STARTUP) or a stop (condition=SHUTDOWN) Trigger. A Trigger can be a single Event or a combination of several Events.

A Trigger is considered satisfied when all of its Events are set. The Events for a Trigger must already be defined. In this case, the BKP1EV is selected for the first STARTUP Trigger of APPL01TR.

To define a trigger for APPL01, perform these tasks:

1. Select the CONDITION policy item.
2. Select the Trigger type: *startup* or *shutdown*. When you press Enter, you see a list of Events to select for the Trigger.
3. Select the events to be combined as part of the Trigger definition and press PF3 to save.

After saving the Trigger definition, you see the name of the Event on the Trigger Conditions panel.

Repeat this process for each Event that is part of the Trigger. If additional start Triggers are required, select the STARTUP 2 Trigger condition and repeat these steps. Stop (SHUTDOWN)

Triggers are defined exactly the same as start Triggers. After you link a Trigger with one or more Events, you must select the resources that are associated with the Trigger by using these items:

- WHERE USED policy item for the Trigger definition
- TRIGGER policy item for the APL or APG policy

## Set Event for BKP1

- Set BKP1 Event when job BKP1 ends
- Select MESSAGES/USER DATA policy item for APL APPL01
- Define a user Automation Table entry for message IEF404I to call INGEVENT to set the Event, BKP1EV

```
IF MSGID = 'IEF404I' & JOBNAME = 'BKP1' THEN  
    EXEC (CMD('INGEVENT BKP1EV,REQ=SET'))  
    CONTINUE(Y);
```

### Set Event for BKP1

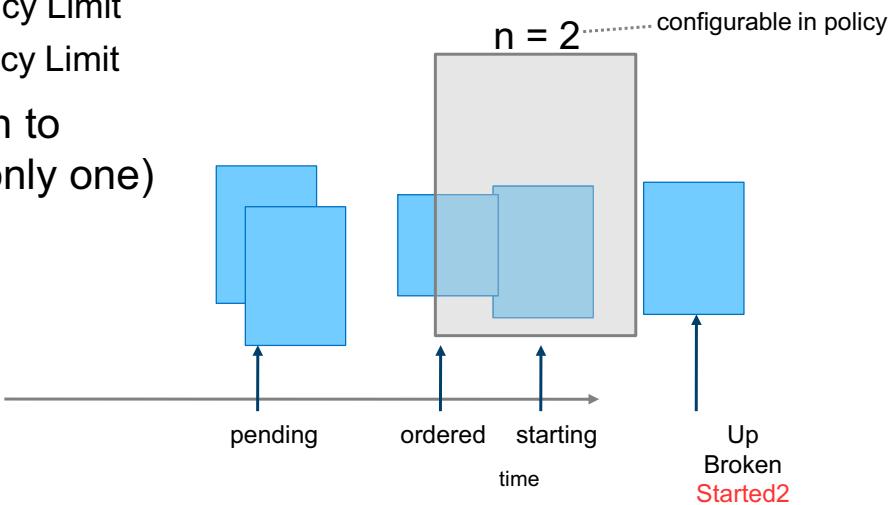
Using the previous example of one Trigger with several Events, this slide illustrates how to set the BKP1 Event when the BKP1 job ends. You must use the INGEVENT command to set the Event. Typically, you use INGEVENT from an Automation Table statement like the one shown on the slide. The Automation Table statement is defined by using the OVR action of the MESSAGES/USER DATA policy item for the Application.



**Hint:** The setting of an Event is a sysplex-wide action. A job might run on any system of the sysplex group and be a Trigger for a start or stop of an Application on another system.

# Pacing Gates

- Used to throttle the starting and stopping of applications
- Create Pacing Gate entry type
- Use Pacing Gate to define the number of applications which are allowed to be started or stopped concurrently:
  - Start Concurrency Limit
  - Stop Concurrency Limit
- Link application to Pacing Gate (only one)



© Copyright IBM Corporation 2019

5-49

## Pacing Gates

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

To facilitate the management of those applications aiming for efficient resource utilization while avoiding customization efforts, the application pacing capability can be used to control how many applications of a kind can be started or stopped at the same time. For this, any application that is eligible to receive a start or stop order from the automation manager has to 'transit' through a defined Pacing Gate.

If the number of applications that are currently in transition reaches the maximum concurrency level, additional applications are held back (waiting) until another application finished this transition and reached the final desired status or terminated during that transition.

In order to use application pacing, you need to define one or more Pacing Gate entries using the customization dialog, option 13 (PAC) and specify the concurrency level of starting and stopping applications. Next, you assign those applications that you want to more tightly control to a Pacing Gate. This is done by selecting a pacing gate using the PACING GATE policy.

Gate mechanism ensures that only up to n automation resources can pass the gate at any given time. The automation manager delays the order of the applications waiting in front of the gate. When the gate is open a start order is sent to the agent and the status becomes starting.

The gate will be opened again when the startup is done successfully, indicated by an UP status or unsuccessfully indicated by a STARTED2 or BROKEN agent status.

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

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

# Create Pacing Gates

- Policy selection panel for Pacing Gates

| AOFGEPOL                 |                  | Policy Selection                          | Row 1 to 5 of 5<br>Scroll ==> CSR |
|--------------------------|------------------|---|-----------------------------------|
| Command ==>              |                  |   |                                   |
| Entry Type : Pacing Gate |                  | PolicyDB Name : OPSU4EX3                  |                                   |
| Entry Name : RV          |                  | Enterprise Name : OPSU4EX3                |                                   |
| Action                   | Policy Name      | Policy Description                        |                                   |
|                          | DESCRIPTION      | Enter description                         |                                   |
|                          | PACING GATE INFO | Enter and display pacing gate information |                                   |
|                          | -----            | -----                                     | -----                             |
|                          | WHERE USED       | List applications linked to this entry    |                                   |
|                          | COPY             | Copy data from an existing entry          |                                   |

| AOFGPAC0                      |    | Pacing Gate Information    |
|-------------------------------|----|----------------------------|
| Command ==>                   |    |                            |
| Entry Type : Pacing Gate      |    | PolicyDB Name : OPSU4EX3   |
| Entry Name : RV               |    | Enterprise Name : OPSU4EX3 |
| Start Concurrency Limit : . . | 1  | (0-9999 or NOLIMIT)        |
| Stop Concurrency Limit : . .  | 99 | (0-9999 or NOLIMIT)        |

- PACING GATE INFO policy item

© Copyright IBM Corporation 2019

5-50

## Create Pacing Gates

The unique policy item for Pacing Gates is:

PACING GATE INFO

Using this policy, you can specify and display Pacing Gate information.

A Pacing Gate consists of the following:

### Start Concurrency Limit

Specifies the number of applications which are allowed to start concurrently by SA z/OS during system startup. If an unlimited number of applications is allowed, specify NOLIMIT which is also the default value. Alternatively exactly one system automation symbol may be specified.

### Stop Concurrency Limit

Specifies the number of applications which are allowed to be stopped concurrently by SA z/OS during system shutdown. If an unlimited number of applications is allowed, specify NOLIMIT which is also the default value. Alternatively exactly one system automation symbol may be specified.

## Policy selection panel for Applications

- Link application to one Pacing Gate

| ACTIONS HELP             |             | Policy Selection                              | Row 1 to 21 of 21<br>Scroll ==> CSR |
|--------------------------|-------------|---|-------------------------------------|
| AOFGEPOL                 | Command ==> |   |                                     |
| Entry Type : Application |             | PolicyDB Name : NEW_PDB                       |                                     |
| Entry Name : AUTOSSI     |             | Enterprise Name : SAPLEX                      |                                     |
| Action                   | Policy Name | Policy Description                            |                                     |
| DESCRIPTION              |             | Enter description                             |                                     |
| UPWARD CLASS             |             | Select a class to inherit data from           |                                     |
| APPLICATION INFO         |             | Define application information                |                                     |
| AUTOMATION FLAGS         |             | Define application automation flags           |                                     |
| APPLICATION SYMBOLS      |             | Define application symbols                    |                                     |
| TRIGGER                  |             | Select trigger                                |                                     |
| SERVICE PERIOD           |             | Select service period                         |                                     |
| PACING GATE              |             | Select pacing gate                            |                                     |
| RELATIONSHIPS            |             | Define relationships                          |                                     |
| MESSAGES/USER DATA       |             | Define messages and user data                 |                                     |
| STARTUP                  |             | Define startup procedures                     |                                     |
| SHUTDOWN                 |             | Define shutdown procedures                    |                                     |
| THRESHOLDS               |             | Define error thresholds                       |                                     |
| MINOR RESOURCES          |             | Define application minor flags and thresholds |                                     |
| SYSTEM ASSOCIATION       |             | Define primary and secondary associations     |                                     |
| -----RESOURCES-----      |             |   |                                     |
| GENERATED RESOURCES      |             | List resources generated for this entry       |                                     |
| MEMBER OF                |             | List resources where this entry is a member   |                                     |
| WHERE USED               |             | List application groups linked to this entry  |                                     |
| COPY                     |             | Copy data from an existing entry              |                                     |

© Copyright IBM Corporation 2019

5-51

*Link application to one Pacing Gate*

You also can use the Policy selection panel for Applications to link an application to one Pacing Gate.

# Lesson 4. Defining USS and MTR resources

## Lesson 4: Defining USS and MTR resources

- USS Applications
  - Application type is USS
  - Use the USS CONTROL policy item to define USS-specific data
  - Import \*USS add-on policies for sample SA z/OS automation
- MTR resources
  - Monitor the Health Status of a resource
    - Active monitor: Health Status set by polling for resource information
    - Passive monitor: Health Status set by processing events
  - HasMonitor relationship associates MTR with an Application or Application Group
  - Base for performance-driven automation with OMEGAMON

### What this lesson is about

This lesson describes how to define USS and MTR resource policy definitions.

### Objectives

After completing this lesson, you should be able to define USS and MTR resource policy definitions.

### References

*SC34-2717 System Automation for z/OS Version 4  
Release 1 Defining Automation Policy*

You can use USS applications for starting and stopping applications and for monitoring. Monitor Resources (MTR) are used for monitoring application-specific performance and health.

## USS Applications

With SA z/OS UNIX Applications, you can start and stop applications and monitor the following items:

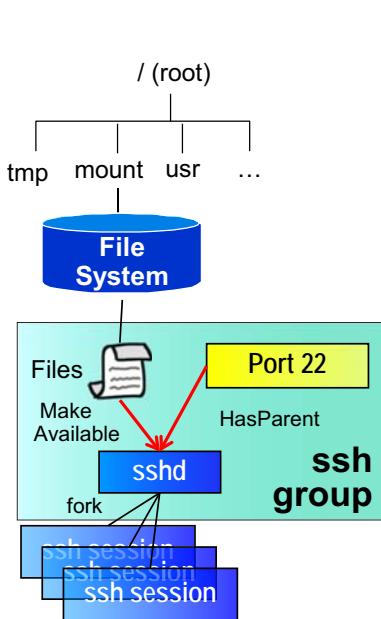
- Processes (represented by the command or path and user ID)
- TCP ports
- Files and file systems
- Generic User Monitoring (the user supplies a z/OS UNIX monitoring routine or script)

You can also run z/OS UNIX commands with the INGUSS command.

## Monitor resources

Application-specific performance and health monitoring can be accomplished with monitor resources (MTRs). MTRs can influence the automation manager Compound Status through the setting of a Health Status for an Application or Application Group. This Health Status can be used for information, or by the automation manager to make decisions and, if necessary, invoke automation for the Application or Application Group.

# UNIX System Services (USS) automation



- Automation policy for USS resources
  - Processes
  - Files and file systems
  - TCP Ports
- **Best practices policy** for ftp, ssh, inetd, syslogd, nfs, webserver...
- UNIX automation infrastructure to
  - Start and stop
  - Monitor using start command, user ID, and a filter (for uniqueness)
- Solution to fork problem uses process ID, fast active monitor and z/OS events to identify the right process
- Trap UNIX syslogd messages
- Security setup

5-53

## UNIX System Services (USS) automation

IBM automation provides comprehensive UNIX System Services (USS) automation that can solve problems like identifying the right process for monitoring and not getting confused when jobnames are changed by a UNIX fork.

The Automation policy for USS resources covers:

- Processes
- Files and file systems
- TCP Ports

The best practices policy for ftp, ssh, inetd, syslogd, nfs, webserver... makes it easy to automate USS resources w/o any required scripting or definition of messages.

The UNIX automation infrastructure allows to:

- Start and stop resources as a UNIX process rather than as a started task (STC)
- Monitor resources using start command, user ID, and a filter to uniquely identify a UNIX process

The Solution to the „fork problem“ uses process ID, a fast active monitor and z/OS events to identify the right process.

The fast active monitor directly reads UNIX control blocks.

An active event is triggered by the provided Image initiation exit and by a termination event triggered by the Pre-Process Termination exit.

UNIX syslogd messages can be forwarded to the MVS log to be processed by the NetView automation table using

```
*.* /dev/console
```

in the syslogd configuration file.

A role-based security setup is provided, operators can be easily joined to a corresponding security group.

The Configuration Assistant creates RACF commands that are necessary to add a user and set the necessary characteristics.

The UNIX System Services Applications have a USS CONTROL section:

- User ID
- Monitoring Command
- Process Command/Path , for example: /usr/sbin/inetd
- with Filter, for example: /etc/inetd.conf
- File Name
- IP Stack
- Port Number

## Defining a USS Application

| Application Information                      |   | Line 00000001  |
|--|---|----------------|
| Command ==> [ ]                              |   | Scroll ==> CSR |
| Entry Type : Application                     | PolicyDB Name : NEW_PDB                           |                |
| Entry Name : SSHD                            | Enterprise Name : SAPLEX                          |                |
| Category : <b>USS</b>                        | (IBM-defined, user-defined or blank,<br>see help) |                |
| Subcategory . . . . .                        | (IBM-defined, user-defined or blank,<br>see help) |                |
| Subsystem Name . . . . . SSHD                |   |                |
| Job Type . . . . .                           | (MVS NONMVS TRANSIENT)                            |                |
| Job Name . . . . . SSHD                      |   |                |
| Transient Rerun . . . . .                    | (YES NO)  |                |
| Scheduling Subsystem . . . . .               | (MSTR, JES Subsystem)                             |                |
| JCL Procedure Name . . . . .                 |   |                |
| Job Log Monitor Interval . . . . .           | (mm:ss NONE)                                      |                |
| Captured Messages Limit . . . . .            | (0 to 999)  |                |
| Desired Available . . . . .                  | (ALWAYS ONDEMAND ASIS)                            |                |
| Restart after IPL . . . . .                  | (START NOSTART NONE)                              |                |
| Monitor for IPL complete . . . . .           | (YES NO)  |                |
| Start Delay . . . . . 00:00:10               | (time for "UP" status checks, hh:mm:ss)           |                |
| Start External Shutdown . . . . .            | (FINAL ALWAYS NEVER)                              |                |
| UP Shutdown Pass Interval . . . . . 00:00:30 | (hh:mm:ss)  |                |
| Rest Cleanup Delay . . . . . 00:00:04        | (hh:mm:ss)  |                |
| External Shutdown . . . . .                  |   |                |
| Command Prefix . . . . .                     |   |                |
| Message Prefix . . . . .                     |   |                |
| Sysname . . . . .                            |   |                |
| Monitor Routine . . . . . AOFUXMON           | (name NONE)                                       |                |
| Monitor Interval . . . . .                   | (hh:mm NONE)                                      |                |
| Inform List . . . . .                        | (SDF EIF E2E IOM ITM SMF TTT USR NONE)            |                |

Press F8

Animated

© Copyright IBM Corporation 2019

5-54

### Defining a USS Application

This slide shows the APPLICATION INFO policy item for the SSHD Application, imported from the \*USS add-on policy. Notice the *Application Type* is **USS**. The SSHD Application monitors the Secure Shell Daemon. The monitor routine is AOFUXMON. Policy definitions can be inherited from other Application classes.

Many of the policy items for a USS Application are the same as they are for a typical z/OS Application. However, USS Applications have a unique policy item called USS CONTROL. Select USS CONTROL to define USS-specific data.

# USS CONTROL policy

USS Control Specification

Line 00000001  
Scroll ==> CSR

Command ==> \_\_\_\_\_

Entry Type : Application                    PolicyDB Name : NEW\_PDB  
Entry Name : SSHD                         Enterprise Name : SAPLEX

User ID . . . . . . . OMVSKERN \_\_\_\_\_

Monitoring Command . . . . .

---

Specify either a USS Process, a USS File or a USS Port:  
Process Command/Path . . .  
/bin/sh

---

with Filter . . . . .  
-c '/usr/sbin/sshd -D'

---

File Name . . . . . . .

---

IP Stack . . . . . . .

Port Number . . . . . .

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

© Copyright IBM Corporation 2019

5-55

USS CONTROL policy

Enter the policy data and press PF3 to save the following policy information:

- **User ID:** Enter the UNIX user ID that this resource belongs to. You can determine the user by issuing a DISPLAY OMVS,a=ALL command.
  - **Monitoring Command:** Enter the UNIX System Services monitoring command that is used to obtain the status of the resource. This command must be a valid program in the HFS. If the field is empty, the standard monitoring routine supplied by SA z/OS is used.

The most likely scenario is to use the standard USS monitoring routine, AOFUXMON. When the Monitoring Command field is empty, SA z/OS internal monitoring is invoked by AOFUXMON. If a command is specified, this command is issued instead of the SA z/OS internal monitoring. The first part of the command must be an executable program (for example, this can be a shell script, a compiled program written in C, or a REXX program).

The following return codes are possible:

- 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 slash (/), 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.

- **Command/Path:** Enter the name of the command that is running without its arguments, as shown by the UNIX command *ps* with the format specification *comm*.
- **File Name:** Enter the path of a file in the HFS. If the file exists and the specified user ID is the owner of the file, the resource is available; otherwise, the resource is not available.
- **Port Number:** Enter the TCP port number that this resource listens to on the local host.

# USS Application STARTUP

```
Command Processing : STARTUP          Line 00000001 Col 001 075
Command ==> _____                      Scroll ==> CSR
Mixed case . . . YES (YES NO)
Cmd Type      AutoFn/* Command Text
               INGUSS JOBNAME=&SUBSJOB &SUBSPATH &SUBSFILTER
***** Bottom of data *****
```

Define INGUSS command with parameters.

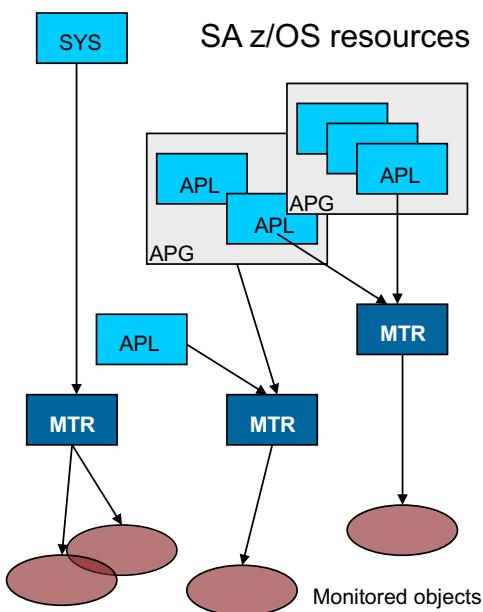
For example, &SUBSPATH points to the *Process/Command Path* definition of the USS CONTROL policy

## USS Application STARTUP

To start a process with the command and job name specified in the customization dialog, enter **INGUSS JOBNAME=&SUBSJOB &SUBSPATH &SUBSFILTER** on the Startup Command Processing panel. This policy is the same STARTUP phase policy that is used for non-USS Applications.

Only the command that was used to start an Application or a process can be monitored. If the same program must be started multiple times, you can use a soft link as PRESTART command to distinguish the processes.

## Monitor resources overview



- Monitor resources are policy objects that are used to maintain the health of other Applications  
Monitor resource name is `monitor_name / MTR / system_name`
- Monitor resources determine the health status of the object that it monitors  
Provides gradients between up and down
- The HasMonitor relationship associates MTR with an Application (APL) or Application Group (APG)
- Health Status is maintained by Automation Manager
- Health Status is obtained either periodically or based on an event
- Health Status is propagated to associated APL and APG
- Monitoring can be active or passive and support Service Periods
- You start and stop with INGREQ
- You manage with DISPMTR command

© Copyright IBM Corporation 2019

5-57

### Monitor resources overview

*Monitor resources* are policy objects that can be used to track the health of Applications and Application Groups. Monitor resources provide status values between up and down. Monitor resources are defined with option **11** of the customization dialog. MTR is the resource type. Monitor resources are like other SA z/OS resources; for example, a monitor resource can be linked to a Service Period definition. Monitor resources can also be stopped and started by using the INGREQ command. Monitor resources set the health status. The health status aggregates up to the compound status, which can be seen on an INGLIST panel. From INGLIST option **M**, you can invoke the **DISPMTR** command to manage your monitor resources and see more detailed status information.

The monitor can be active (Health Status determined by monitor-command polling data for the resources) or passive (Health Status determined by events that cause the automation). The status of the monitor resource is called the *Health Status*. The Health Status aggregates up to the Compound Status, which can be seen on an INGLIST panel. From INGLIST option **M** you can invoke the **DISPMTR** command to manage your monitor resources and see more detailed status information.

## Active monitor versus passive monitor

- Active monitor
  - Status determined by a monitor command
  - Timer scheduled based on defined interval
- Passive monitor
  - Status based on predefined events of the INGMON command
  - No monitor interval defined. No timer scheduled
- Can combine Active monitor with events of the INGMON to update status

### *Active monitor versus passive monitor*

*Active monitors* use a monitor command to periodically retrieve data and set the Health Status based on the results of that data. The monitor command is typically written by the system administrator.

*Passive monitors* wait for events to occur before changing the Health Status of a monitor resource. The Health Status is set through Automation Table logic to call the INGMON command. Typically, a passive monitor has a monitor or activate command defined to set the Health Status initially. You can define an active monitor and also define events to update the Health Status.

## Monitor resource definition

|  |   |
|--|---|
| <b>AOFGLPOL</b><br><b>Action</b><br><b>s</b>         | <b>Policy Selection</b><br><b>Row 1 to 10 of 10</b><br><b>Scroll ==&gt; CSR</b><br><b>Select MONITOR INFO policy item, for example, to define a monitor command or interval.</b>  |
|  | <b>PolicyDB Name : NEW_PDB</b><br><b>Enterprise Name : SAPLEX</b>   |
|  | <b>Policy Description</b><br><b>Enter description</b><br><b>Define monitor information</b>  |
|  | <b>AOFGMTR</b><br><b>Command ==&gt;</b><br><b>Entry Type : Monitor Resource</b><br><b>Entry Name : JES2SPPOOL</b><br><b>Monitored Object. . . . . SPPOOL</b><br><b>Monitored Jobname . . . . . JES2</b><br><b>Activate Command. . . . .</b><br><b>Deactivate Command . . . . .</b><br><b>Monitor Command . . . . . INGRMJSP</b>   |
| <b>Define HasMonitor relationship for APL or APG</b> | <b>Monitor Resource Information</b><br><b>More: +</b><br><b>PolicyDB Name : NEW_PDB</b><br><b>Enterprise Name : SAPLEX</b><br><b>Active Monitor</b>   |
|  | <b>Monitoring Interval . . . . . 00:05</b><br><b>Captured Messages Limit . . . 20</b><br><b>(hh:mm)</b><br><b>(0 to 999)</b><br><b>Desired Available . . . . .</b><br><b>Monitor for IPL complete. . . . .</b><br><b>(ALWAYS ONDEMAND ASIS)</b><br><b>Inform List . . . . .</b><br><b>(YES NO)</b><br><b>Owner . . . . .</b><br><b>Info Link . . . . .</b><br><b>(SDF EIF E2E IOM ITM SMF TTT USR NONE)</b> |

© Copyright IBM Corporation 2019

5-59

### Monitor resource definition

This slide provides an overview of the policy items for monitor resources (MTRs). The MONITOR INFO policy item defines commands to be run when the MTR is started or stopped and a command to use for monitoring. The monitoring interval is used to schedule the monitor command for active monitoring. In this case, INGRMJSP is scheduled for every five minutes.

**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.

**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.

**Activate Command** can be any NetView command. It is executed when the monitor resource is started. In this example, no activate command is defined.

**Deactivate Command** can be any NetView command. It is executed when the monitor resource is stopped. In this example, no deactivate command is defined.

**Monitor Command** can be any NetView command. If a command and a monitoring interval are specified, the command is scheduled as a timer, based on the monitoring interval, while the monitor

is active. If a command is specified and no monitoring interval is specified, the command is executed when the monitor is started. SA z/OS expects the monitor command to return a valid health status code. Additionally the monitor command can issue a message that is then attached to the health status.

*Monitoring Interval* specifies an interval of time for periodic execution of the monitor command. An interval value of zero is forced to blank. If an interval is specified, a monitor command must also be specified. If no interval is specified, the monitor is *passive*.

*Captured Messages Limit* defines the number of messages to save for the monitor. If this field is blank, the default captured messages limit is used. In addition to defining MTR policy, the system administrator must also link the Applications and Application Groups to the MTR by using a HasMonitor relationship.

## Connecting APLs or APGs to MTRs

- MTRs are connected to APLs or APGs by using a HasMonitor relationship
- One MTR can be connected to any number of APLs or APGs (zero to  $n$ )
- One APL or APG can have any number of MTRs connected
- MTRs can have any relationship except HasMonitor
- MTRs cannot be connected to other MTRs
- MTRs cannot be members of APGs

Each monitor resource can be linked to an Application or Application Group by using the HasMonitor relationship defined for the Application or Application Group. This slide illustrates a few basic rules for monitor resources. For example, monitor resources cannot be linked to other monitor resources, nor can they be members of an Application Group.

# System administrator and operator tasks

## System Administrator

- Define MTR policy objects
- Use HasMonitor relationship to associate APL or APG resources to the MTR policy objects
- Code one or more of the following commands per MTR:
  - Monitor command
  - Activate command
  - Deactivate command
  - Healthstate commands
- Define events for status changes.

## Operator

- View Health Status in INGLIST.
- Use DISPMTR command to manage monitoring resources

The system administrator defines the monitor resource (MTR) policy by using the customization dialog (option 11). The administrator can define the MTR policy and optionally use the following commands:

- **Monitor:** Command to be scheduled based on the monitor interval.
- **Activate:** Command to be run every time the monitor is started.
- **Deactivate:** Command to be run every time the monitor is stopped.
- **Healthstate:** Command to use when the monitor status is changed. A different command can be coded for each supported status.

The administrator can optionally define events for automation to call INGMON to update the Health Status. To start the monitor when the resource is available, define MakeAvailable/WhenAvailable and MakeUnavailable/WhenDownOrStopping relationships. Use the HasMonitor relationship to associate Applications or Application Groups with the monitor resource.

## Overview of monitor processes

- The following processes can occur:
  - Start processing
    - Issue activate command, if defined
    - Run initial monitor processing, if monitor command defined
    - Set up timer to do periodic monitor processing, if interval defined
  - Stop processing
    - Cancel timer, if any
    - Issue deactivate command, if defined
  - Monitor processing
    - Issue monitor command
    - Set return code to determine Health Status
    - Issue message (optional)
- Service Periods can be used to start and stop monitor resources

© Copyright IBM Corporation 2019

5-62

### Overview of monitor processes

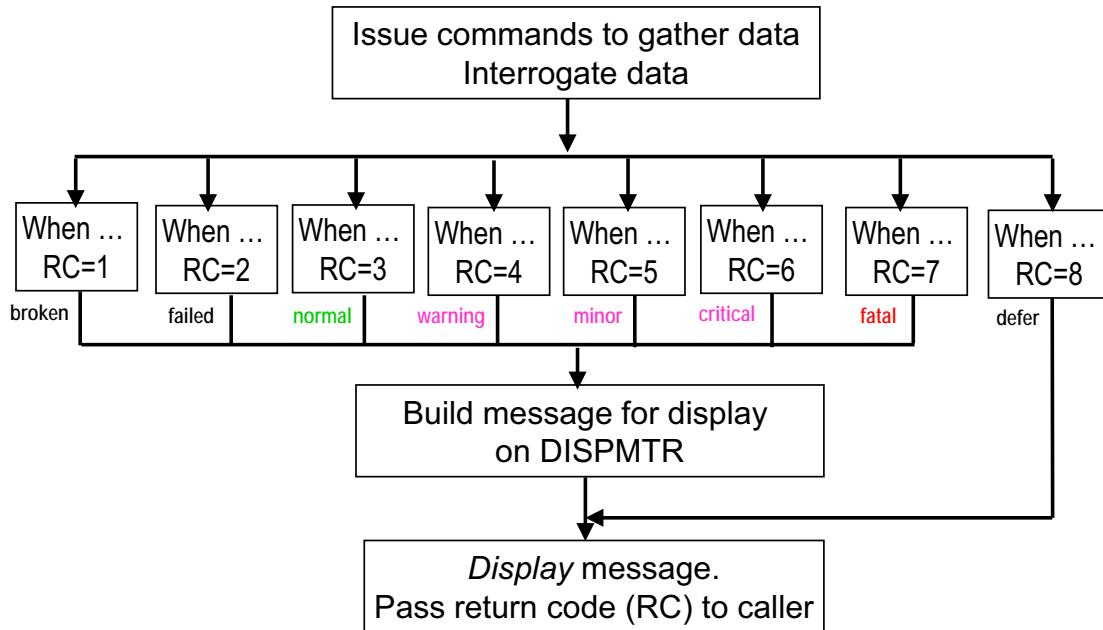
Depending upon your policy, several processes might be invoked at the start or stop of a monitor resource. If it is an active monitor, a monitor command must be defined. Service Periods can also be defined to affect when a monitor resource runs.

For example, when a monitor starts, the following actions occur:

- The activate command, if defined, performs any necessary setup.
- If a monitor command is defined, it is used to set the initial Health Status.
- If a monitor interval is defined, the monitor command is scheduled as a NetView timer based on that interval.

These processes occur every time the monitor is started or stopped, including as a result of a Service Period.

# Monitor command basics



## Monitor command basics

If you define an *active monitor*, you must code a *monitor command*. The purpose of the monitor command is to collect the data necessary to set the Health Status for the Applications or Application Groups being monitored. This can be any NetView command. The monitor command can be coded in any language that is supported by NetView. The SA z/OS Health Status is determined by the return code set by the monitor command.

This high-level flow is for a basic monitor command with four steps:

1. Issue one or more commands to gather data and then interrogate the responses. You might have to issue other commands, based on the response.
2. Set return code of 1 - 8, based on interrogating the response.
3. Optionally, set message text to be displayed on DISPMTR and saved.
4. Pass the message and return code back to the SA z/OS process used for the monitor command. The SA z/OS process checks the return code and sets the Health Status appropriately.

## Health Status aggregation

| Health Status   | RC       | Description  | Compound Status     |
|-----------------|----------|--|---------------------|
| <b>NORMAL</b>   | <b>3</b> | Resource health is good  | <b>SATISFACTORY</b> |
| <b>WARNING</b>  | <b>4</b> | Resource health is becoming degraded   | <b>DEGRADED</b>     |
| <b>MINOR</b>    | <b>5</b> | Similar to WARNING but more severe   | <b>DEGRADED</b>     |
| <b>CRITICAL</b> | <b>6</b> | Similar to MINOR but more severe   | <b>DEGRADED</b>     |
| <b>FATAL</b>    | <b>7</b> | Similar to CRITICAL but more severe<br><b>Automatic ForceDown of related application</b> | <b>PROBLEM</b>      |
| <b>DEFER</b>    | <b>8</b> | Action deferred to automation<br>Used for OMEGAMON exceptions                            | No effect           |
| <b>UNKNOWN</b>  |          | Monitor is not running; might need to be started or fixed                                | No effect           |

Health Status is determined from the return code of the monitor command (active) or set by using INGMON (passive)

© Copyright IBM Corporation 2019

5-64

### Health Status aggregation

The return codes (RC) in this table are set by the monitor command and can be used to affect the Compound Status as shown. Return codes 4 - 6 can be used to indicate a potential problem that is getting worse over time.

The health status values affect the compound status in the automation manager.

**FATAL** is a guaranteed automatic ForceDown, and, if available, failover for the application associated with the monitor.



**Note:** You must use return code **3** when the resource health is good.

You can also use INGMON to set the Health Status. In addition to the Health Status, monitor resources can have several states associated with them, as follows:

- **ACTIVE:** Monitor is running. The Health Status is determined by the monitor command or events driving INGMON.
- **INACTIVE:** Monitor is not running. The Health Status is UNKNOWN.

- **BROKEN:** The monitor command encountered an error and is not capable of running. The monitor command exits with a return code of 1. The Health Status is UNKNOWN.
- **FAILED:** The monitor command encountered an error and will be rescheduled based on the monitor interval. The monitor command must exit with a return code of 2. The Health Status is UNKNOWN.

## HEALTHSTATE policy item

```
A0FGDYN9      Command Processing : HEALTH_RECOVERY      Line 00000001 Col 001 075
Command ===> _____ Scroll ===> CSR

Cmd State      AutoFn/* Command Text
--- FATAL      MSG ALL MTRIP change to FATAL state
--- NORMAL     MSG ALL MTRIP change to NORMAL state - command 1.
--- NORMAL     MSG ALL MTRIP change to NORMAL state - command 2.
***** Bottom of data *****
```

- Healthstate=FATAL: When MTRIP changes to FATAL status, issue **MSG ALL MTRIP change to FATAL state** command: SA will shut down the resource and not allow it to be restarted for 15 minutes to allow for a move
- Healthstate=NORMAL: When MTRIP changes to NORMAL status, issue 2 MSG ALL commands:
  - **MSG ALL MTRIP change to normal state – command 1**
  - **MSG ALL MTRIP change to normal state – command 2**
- All other Health Status changes: Take no action

© Copyright IBM Corporation 2019

5-65

### HEALTHSTATE policy item

The HEALTHSTATE policy defines commands that are executed when the monitored resource changes status. Valid values are NORMAL, WARNING, MINOR, CRITICAL, FATAL, or blank.

The example on the slide shows these commands:

- One command defined to be run every time the monitor status becomes FATAL.
- Two commands defined to be run every time the monitor status becomes NORMAL.

Leaving the *Healthstate* field blank means that the specified commands run for every healthstate change.

# HasMonitor relationship

```

A0FGXREQ          Define Relationship
Command ==> _____
Entry Type : Application      PolicyDB Name   : NEW_PDB
Entry Name  : TCPIP           Enterprise Name : SAPLEX

Subsystem Name     : TCPIP
Description . . . . . TCPIP has a monitor relationship

Relationship Type. . HASMONITOR          MAKEAVAILABLE MAKEUNAVAILABLE
                                         PREPAVAILABLE PREPUNAVAILABLE
                                         HASPARENT HASPASSIVEPARENT
                                         HASMONITOR PEEROF
                                         FORCEDOWN EXTERNALLY

Supporting Resource. MTRIP/MTR/=          Resource Name
Sequence Number. . . . .             Sequence Number (1-99,blank)

Automation . . . . .               ACTIVE PASSIVE
Chaining . . . . .                STRONG WEAK
Condition . . . . .               Satisfy condition
                                   (? for list of possible values)

```

Application TCPIP has  
a monitor called  
MTRIP

© Copyright IBM Corporation 2019

5-66

## HasMonitor relationship

You can define a HasMonitor relationship between the Application or Application Group and the MTR. To define the relationship, use the RELATIONSHIPS policy item of the Application or Application Group being monitored. In this case, the TCPIP Application has a monitor of MTRIP.

## Passive monitor: MESSAGES/USER DATA

| AOFGDYNM  |                               | Message Processing                 | Line 00000001 Col 001 075 |          |          |          |          |
|---|-------------------------------|------------------------------------|---------------------------|----------|----------|----------|----------|
| Command ==> █   |                               |                                    | Scroll ==> CSR            |          |          |          |          |
| Entry Type : Monitor Resource   |                               | PolicyDB Name : NEW_PDB            |                           |          |          |          |          |
| Entry Name : JES2MON  |                               | Enterprise Name : SAPLEX           |                           |          |          |          |          |
| Line Commands: S/C (Cmd), R (Rep), K (Cod), U (Usr), A,M (AT, MRT, MPF) |                               |                                    |                           |          |          |          |          |
|   | I, D (insert or delete lines) |                                    |                           |          |          |          |          |
| Message ID field length. .  | <u>16</u>                     | (1 - 32)                           |                           |          |          |          |          |
| <b>Cmd</b>  | <b>Message ID</b>             | <b>Description</b>                 | <b>C</b>                  | <b>R</b> | <b>K</b> | <b>U</b> | <b>A</b> |
|   | HASP9084                      | JES2_monitor_address_space_started |                           |          |          | *        |          |
|   | HASP9085                      | JES2_monitor_address_space_stopped |                           |          |          | *        |          |
|   | HASP9121                      | No_outstanding_alerts              |                           |          |          | *        |          |
|   | HASP9201                      | Main_task_wait_detected            |                           |          |          | *        |          |
| <b>a</b>  | <b>HASP9202</b>               | Potential_JES2_main_task_loop      |                           |          |          | *        |          |
|   | HASP9203                      | Long_PCE_dispatch                  |                           |          |          | *        |          |
|   | HASP9204                      | Main_task_busy                     |                           |          |          | *        |          |
|   | HASP9207                      | Checkpoint_lock_held               |                           |          |          | *        |          |
|   | HASP9208                      | Main_task_local_lock_wait          |                           |          |          | *        |          |
|   | HASP9209                      | Main_task_non_dispatchable         |                           |          |          | *        |          |
|   | HASP9210                      | Main_task_paging_wait              |                           |          |          | *        |          |
|   | HASP9211                      | Main_task_not_running              |                           |          |          | *        |          |
|   | HASP9212                      | Main_task_not_dispatched           |                           |          |          | *        |          |
|   | HASP9213                      | Long_JES2_command_processing       |                           |          |          | *        |          |
|   | HASP9301                      | Main_task_alerts_cleared           |                           |          |          | *        |          |
|   | HASP9302                      | Checkpoint_lock_released           |                           |          |          | *        |          |
| <b>***** Bottom of data *****</b>                                       |                               |                                    |                           |          |          |          |          |

These messages cause Health Status changes for JES2. Select **A** (AT)

- Use **AS** action to define the type of message; for example, UP message
- Use **AO** action to display Automation Table statement invoking INGMON command to set Health Status

© Copyright IBM Corporation 2019

5-67

### Passive monitor: MESSAGES/USER DATA

For *Passive monitoring*, you must define events that use INGMON to set the Health Status for a resource. You define the events in the MTR policy under the MESSAGES/USER DATA policy. This example shows the JES2MON passive monitor supplied by SA z/OS. There are several JES messages defined that set the Health Status of JES2. To see how the Health Status is set, select **A (AT)** and press Enter for \$HASP9202, for example.

## Passive monitor: Set Health Status

**AOFGMADV**                      **Message Automation Overview**                      **Scroll ==> CSR**

Command ==> \_\_\_\_\_

|                 |                         |                            |
|-----------------|-------------------------|----------------------------|
| AC AT condition | MS MRT action selection | MF MPFLSTxx specifications |
| AS AT status    | MO MRT override         | IG Ignore for AT, MRT, MPF |
| AO AT override  |                         |                            |

Message ID : HASP9202

Generated AT entry :

```
IF MSGID = . 'HASP9202' THEN
  EXEC(CMD('INGMON MON,MSGTYPE=''HASP9202'',STATUS=CRITICAL') ROUTE(ONE
  %AOFGOPGSSOPER%));
```

Generated MRT entry :

UP AOFGMAT1                      AT Status Specification  
SE Command ==> \_\_\_\_\_  
WH \_\_\_\_\_  
RE View or change specifications for message: HASP9202

Ge Application / MVC Component status

|               |                          |  |
|---------------|--------------------------|--|
| HA _ ACTIVE   | Starting                 | Monitor health status                    |
| _ UP          | Available                | _ NORMAL Resource shows good results     |
| _ HALTED      | Degraded                 | _ WARNING Resource shows degradation     |
| _ Terminating | Received STOP command    | _ MINOR More severe than WARNING         |
| _ Terminated  | Terminated normally      | <b>S CRITICAL</b> More severe than MINOR |
| _ ABENDING    | Going to end abnormally  | _ FATAL More severe than CRITICAL        |
| _ Abended     | Ended abnormally         | _ UNKNOWN Health status not available    |
| _ BREAKING    | Non-recoverable abending | _ Check Evaluate health status           |
| _ BROKEN      | Non-recoverable abend    | Monitor status                           |

Capture specification

|           |                 |                                 |
|-----------|-----------------|---------------------------------|
| _ Capture | Capture Message | _ FAILED Monitor failed         |
| DOM . . . |                 | _ BROKEN Monitor finally failed |

When a \$HASP9202 message is received, set the Health Status of the JES2MON monitor resource to **CRITICAL**

© Copyright IBM Corporation 2019

5-68

### *Passive monitor: Set Health Status*

On this slide are the possible values for the Health Status and Application status. In this case, \$HASP9202 indicates JES2 is potentially in a loop. The Health Status is set to CRITICAL.

## INGLIST JES\* example

| SA z/OS - Command Dialogs |           |          |                       |              |           |           |           |            |                    | Line 1    | of 3     |                       |                       |
|---------------------------|-----------|----------|-----------------------|--------------|-----------|-----------|-----------|------------|--------------------|-----------|----------|-----------------------|-----------------------|
| Domain Id . : AOFDA       |           |          | Operator Id : INGC102 |              |           | INGLIST   |           |            | Sysplex = SYSPLEX1 |           |          | Date . . . : 03/22/19 | Time . . . : 08:44:13 |
| A Update                  | B Start   | C Stop   | D INGRELS             | E INGVOTE    | F INGINFO | G Members | H DISPTRG | I INGSCHED | J INGGROUP         | K INGCICS | L INGIMS | M DISPMTR             | P INGPAC              |
| R Resume                  | S Suspend | T INGTWS | U User                | X INGWHY     | / scroll  |           |           |            |                    |           |          |                       |                       |
| CMD Name                  | Type      | System   | Sus                   | Compound     | Desired   | Observed  |           |            |                    |           |          |                       | Nature                |
| — JES2                    | APL       | MVSA     |                       | SATISFACTORY | AVAILABLE | AVAILABLE |           |            |                    |           |          |                       |                       |
| — JES2MON                 | MTR       | MVSA     |                       | SATISFACTORY | AVAILABLE | AVAILABLE |           |            |                    |           |          |                       |                       |
| — JES2SPOOL               | MTR       | MVSA     |                       | SATISFACTORY | AVAILABLE | AVAILABLE |           |            |                    |           |          |                       |                       |

PF11 to scroll right

| SA z/OS - Command Dialogs |           |          |                       |           |           |           |           |            |                    | Line 1    | of 3     |                       |                       |
|---------------------------|-----------|----------|-----------------------|-----------|-----------|-----------|-----------|------------|--------------------|-----------|----------|-----------------------|-----------------------|
| Domain Id . : AOFDA       |           |          | Operator Id : INGC102 |           |           | INGLIST   |           |            | Sysplex = SYSPLEX1 |           |          | Date . . . : 03/22/19 | Time . . . : 08:46:50 |
| A Update                  | B Start   | C Stop   | D INGRELS             | E INGVOTE | F INGINFO | G Members | H DISPTRG | I INGSCHED | J INGGROUP         | K INGCICS | L INGIMS | M DISPMTR             | P INGPAC              |
| R Resume                  | S Suspend | T INGTWS | U User                | X INGWHY  | / scroll  |           |           |            |                    |           |          |                       |                       |
| CMD Name                  | Type      | System   | Automation            | Startable | Health    |           |           |            |                    |           |          |                       | Auto Hold             |
| — JES2                    | APL       | MVSA     | IDLE                  | YES       | NORMAL    |           |           |            |                    |           |          |                       | YES NO                |
| — JES2MON                 | MTR       | MVSA     | IDLE                  | YES       | NORMAL    |           |           |            |                    |           |          |                       | YES NO                |
| — JES2SPOOL               | MTR       | MVSA     | IDLE                  | YES       | NORMAL    |           |           |            |                    |           |          |                       | YES NO                |

© Copyright IBM Corporation 2019

5-69

### INGLIST JES\* example

Using **INGLIST**, you can see the Health Status for resources. In this example, JES2 on MVSA is SATISFACTORY. The Desired Status and Observed Status are AVAILABLE. Press PF11 to scroll to the right to see the remaining status values. The Health Status is NORMAL. You can see more detailed information about the Health Status by selecting option **M** to invoke DISPMTR.

## DISPMTR details example: Active monitor

```

INGKYM01          SA z/OS - Command Dialogs           Line 1  of 35
Domain Id : AOFDA ----- DISPMTR ----- Date . . : 03/22/19
Operator Id : INGC102 Sysplex = SYSPLEX1 Time . . : 08:52:02

Monitor          : JES2SPOOL/MTR/MVSA
System           : MVSA
Description      : JES2 Spool Monitor

Monitored Object : SPOOL
Monitored Jobname: JES2

Inform List       : SDF

Commands...
  Activate       :
  Deactivate     :
  Monitoring    : INGRMJSP
Interval         : 00:05
Last start        : 08:44:00 on 03/22/19
Monitor Status   : ACTIVE at 2019-03-22 08:49:02
Health Status    : NORMAL
                  2.6104% SPOOL UTILIZATION

Policy Definitions for JES2SPOOL ...
  Health State Commands :
    WARNING   'AOFRSD01'
    CRITICAL  'AOFRSD01'

Monitoring History (maximum is 20) ...
  2019-03-22 08:44:00 - STARTING      HEALTH=UNKNOWN
    Monitor starting
  2019-03-22 08:44:02 - ACTIVE       HEALTH=NORMAL
    2.2891% SPOOL UTILIZATION
  2019-03-22 08:49:02 - ACTIVE       HEALTH=NORMAL
    2.6104% SPOOL UTILIZATION

```

Active monitor command INGRMJSP sets the Health Status to NORMAL

### DISPMTR details example: Active monitor

On this slide is a DISPMTR details example for an active monitor, INGRMJSP, that monitors the JES2 SPOOL. The current Health Status for this monitor is NORMAL. Press PF8 to see additional information, such as the healthstate commands and the history of messages over time.

## DISPMTR details example: Passive monitor

```
INGKYM01          SA z/OS - Command Dialogs      Line 1   of 30
Domain Id : AOFDA ----- DISPMTR ----- Date . . : 03/22/19
Operator Id : INGC102           Sysplex = SYSPLEX1 Time . . : 08:54:34

Monitor          : JES2MON/MTR/MVSA
System           : MVSA
Description       : Monitor for JES2

Monitored Object : MON
Monitored Jobname: JES2MON

Inform List       : SDF

Commands...
  Activate       :
  Deactivate     :
  Monitoring    : pipe cc 10 mvs $jd status ! take 1 ! sep ! cons ! collect
                  t ! pipend 8

Interval          :

Last start        : 08:44:00 on 03/22/19

Monitor Status   : ACTIVE at 2019-03-22 08:44:01
Health Status    : NORMAL
                  $HASP9121 NO OUTSTANDING ALERTS

Monitoring History (maximum is 20) ...
  2019-03-22 08:44:00 - STARTING      HEALTH=UNKNOWN
  Monitor starting
  2019-03-22 08:44:01 - ACTIVE       HEALTH=UNKNOWN
  Monitor started
  2019-03-22 08:44:01 - ACTIVE       HEALTH=NORMAL
  $HASP9121 NO OUTSTANDING ALERTS
```

JES2MON passive monitor for JES2  
\$HASP9121 is received, and INGMON set the Health Status to NORMAL

© Copyright IBM Corporation 2019

5-71

### DISPMTR details example: Passive monitor

On this slide is a DISPMTR details example for the JES2MON passive monitor. No monitor command is defined (see **Monitoring** field in Commands section) and no interval is defined. Message \$HASP9121 indicates that JES2 has no outstanding alerts.

# Lesson 5. Automation Tables

## Lesson 5: Automation Tables

- MPFLSTSA: MPF definitions based on automation policies in PDB
- INGEMPF in SINGNSAMP provides default MPF entries
- INGMSGSA: Automation for SA z/OS messages
  - Loaded parallel to INGMSG01
- INGMSG01: Master SA z/OS Automation Table
  - Provides message suppression that is necessary to prevent mismatches and duplicate automation
  - Includes other required AT members
  - AOFMSGSY contains synonyms
  - INGMSGU1 and INGMSGU2 contain user AT code fragments
  - INGMSG02 provides the primary Automation Table logic to trap and process messages for SA z/OS. Built dynamically based on customization dialog policies

### What this lesson is about

This lesson describes the use and purpose of automation tables in System Automation for z/OS.

### Objectives

After completing this lesson, you should be able to describe the use and purpose of automation tables in System Automation for z/OS.

### References

*SC34-2717 System Automation for z/OS Version 4  
Release 1 Defining Automation Policy*

Several Automation Tables are used by SA z/OS. Some are loaded briefly during initialization. Others are built and loaded based on policy definitions.

INGMSG02 is created dynamically by the SA z/OS agent during initialization. AOFMSGSY contains synonyms used by the other Automation Tables. If you make changes to the AOP operator definitions, review AOFMSGSY.

IBM includes versions of INGMSG01, INGMSGSA, AOFMSGSY, INGMSGU1, and INGMSGU2 in the SINGNPRM data set. GDPS provides several Automation Tables: GEOMSGGP and GEOMSGG0. SA z/OS processor operations automation uses the INGMSGHW Automation Table.

## MPFLSTSA and INGEMPF

- MPFLSTSA is dynamically built, based on messages in automation policy:
  - SA z/OS predefined messages and user-defined messages
- INGEMPF provides default MPF entries
- MSG (Messages, option 21) policy:
  - MESSAGES/USER DATA: Entries for specific messages
  - MPF DEFINITIONS: Entries for default processing
    - MPF header: Suppress messages, pass to automation  
.DEFAULT,SUP(YES),RETAIN(I,CE),AUTO(YES)
    - MPF trailer: Display messages, pass to automation  
.NO\_ENTRY,SUP(NO),RETAIN(I,CE),AUTO(NO)

© Copyright IBM Corporation 2019

5-73

### MPFLSTSA and INGEMPF

SA z/OS generates a member in the build output data set called MPSLSTSA. The SA z/OS administrator can use this member as input into z/OS MPF processing. Both MPFLSTSA and INGEMPF are needed for proper SA z/OS automation.



**Hint:** Copy the MPFLSTSA member to a system PARMLIB and activate it with the SET MPF command. Perform this step every time you build the policy.

You can customize the header and trailer lines that are built when MPFLSTSA is generated. Messages can be automated only if they yield an AUTO(YES) when passed through the MPF list members. If your installation requires a default MPF entry to display all messages, you can customize the DEFAULT line to specify **SUP(NO)**. Every time the MPF list is generated, your default definition is included.

# INGMSG01

| System Information                                     |   | Automation Table specified in the SYSTEM INFO policy item for each system |
|--|---|---|
| AOFGSPD0   | PolicyDB Name : NEW_PDB                     |   |
| Command ==>  | Enterprise Name : SAPTEX                    |   |
| Entry Type : System                                    |   |   |
| Entry Name : MVSBR                                     |   |   |
| Operating system : MVS                                 |   |   |
| Image/System name. . . : MVSBR                         |   |   |
| The following specifications are for MVS systems only: |   |   |
| Primary JES. . . . . : JES2                            | Primary JES2/JES3 subsystem name            |   |
| System monitor time. . . : 00:59                       | Time between monitor cycles (hh:mm or NONE) |   |
| Gateway monitor time. . . : 00:15                      | Time between monitor cycles (hh:mm or NONE) |   |
| Automation table(s). . . : INGMSG01                    |   |   |
| SDF root name. . . . . :                               | NetView automation table members            |   |
| Exit name(s) . . . . . :                               | Root of system's SDF tree                   |   |
| USS automation path. . . : /usr/lpp/ing/ussauto/lib    | Environment setup user exit names           |   |
| System Automation UNIX installation path               |   |   |
| SA NetView domain. . . : AOFDB                         | NetView domain ID of SA z/OS                |   |
| Network NetView domain :                               | NetView domain ID of network automation     |   |

- INGMSG01 is default AT
- Loaded when ACF is loaded
- Has the following INCLUDE order:
  - %INCLUDE AOFMSGSY for synonyms
  - %INCLUDE INGMSGU1
  - %INCLUDE INGMSG02 (auto-generated)
  - %INCLUDE INGMSGU2

© Copyright IBM Corporation 2019

5-74

## INGMSG01

The **Automation Tables** field defines one or more Automation Tables that are loaded and used for message automation. These Automation Tables replace INGMSG00, the Automation Table that is used during initialization. The default is INGMSG01. This field can remain blank. However, if you want to have SA z/OS load additional ATs besides INGMSG01, then you must list INGMSG01. The ATs are loaded (refreshed) by the agent in the same sequence in which they are listed here.

INGMSG01 is also used to suppress the following messages:

- Some command echoes
- Command echoes for remote domains
- REXX SAY and TRACE
- Broadcast messages in an EMCS environment
- SA z/OS messages originating from the SSI
- Messages from other SYSDIDs
- Old messages queued by the SSI

INGMSG01 prevents multiple autotasks from automating off the same broadcast message.

## INGMSGU1 and INGMSGU2

- INGMSGU1 is included before INGMSG02
  - Used to override or prevent from flowing to INGMSG02
- INGMSGU2 is included after INGMSG02 and can contain other user entries
- Use of INGMSGU1 and INGMSGU2 must be rare
  - If required, define user Automation Table
- Additional Automation Tables can be used
  - Defined in SYSTEM INFO policy for each system
  - Loaded at agent initialization

### *INGMSGU1 and INGMSGU2*

User-customized AT segments belong in these members. Use INGMSGU1 if the entries must precede SA z/OS INGMSG02 entries. Otherwise, use INGMSGU2.

## INGMSGSA

- Is an Automation Table that contains automation for important messages and must not be changed
- Is loaded with INGMSG01

```
* AOFDA      AUTOTBL STATUS
' AOFDA
BNH361I THE AUTOMATION TABLE CONSISTS OF THE FOLLOWING LIST OF MEMBERS:
AOFDAPPt COMPLETED INSERT FOR TABLE #1: DSITBL01 AT 02/11/19 14:57:05 (FIRST)
DSI#0002 COMPLETED INSERT FOR TABLE #2: INGMSGSA AT 03/22/19 08:43:58
DSI#0003 COMPLETED INSERT FOR TABLE #3: INGMSG01 AT 03/22/19 08:43:58

' AOFDA
BNH363I THE AUTOMATION TABLE CONTAINS THE FOLLOWING DISABLED STATEMENTS:
TABLE: INGMSGSA INCLUDE: ____n/a____ GROUP : INGJES3
TABLE: INGMSGSA INCLUDE: ____n/a____ GROUP : INGPOPS
TABLE: INGMSGSA INCLUDE: ____n/a____ GROUP : INGTAPE
----- -----
TABLE: INGMSG01 INCLUDE: ____n/a____ GROUP : INGCICS
TABLE: INGMSG01 INCLUDE: ____n/a____ GROUP : INGIMAGE
TABLE: INGMSG01 INCLUDE: ____n/a____ GROUP : INGIMS
TABLE: INGMSG01 INCLUDE: ____n/a____ GROUP : INGJES3
```

INGMSGSA

SA z/OS provides this automation table containing all statements that are required for SA z/OS to work properly

# Lesson 6. Dynamic AT and MRT build

## Lesson 6: Dynamic AT and MRT build

- You can build an AT and MRT dynamically based on policy definitions and predefined message automation
- The AT is loaded as INGMSG02 into NetView STORAGE as INGMSG02
- The SA z/OS administrator can modify AT entries under the MESSAGES/USER DATA policy item  
Note: The administrator must have NetView AT programming skill
- User automation is supported and placed in a separate AT

### What this lesson is about

This lesson describes how to dynamically build the automation table and message revision table.

### Objectives

After completing this lesson, you should be able to dynamically build the automation table and message revision table.

### References

*SC34-2717 System Automation for z/OS Version 4 Release 1 Defining Automation Policy*

Dynamically building the Automation Table is known as *Easy Message Management*. This process reduces the amount of Automation Table customization required by the system administrator. The Automation Table files are created when the automation policy is built.

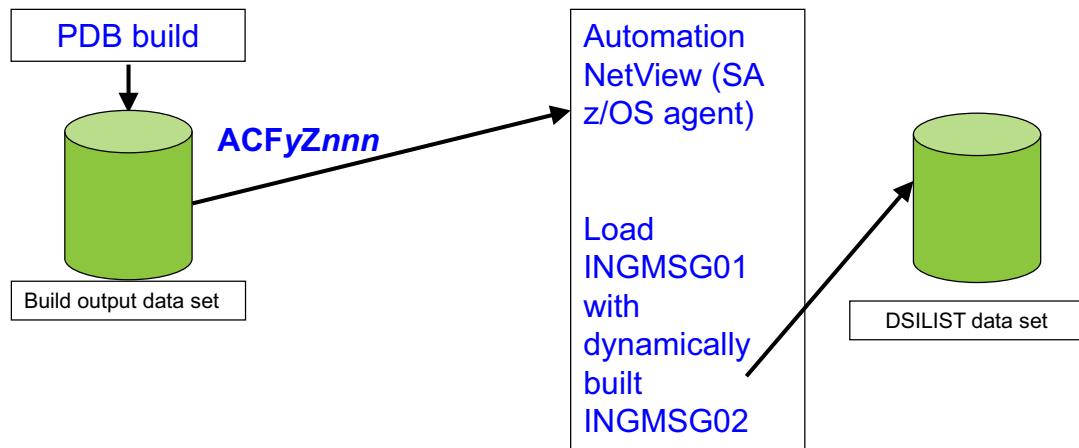
Building an Automation Table has the following benefits:

- No need to define all user entries in INGMSGU1 or INGMSGU2.
- SA z/OS detects whether an AT must be updated because of a change in the policy. If so, SA z/OS rebuilds the AT.
- The SA z/OS administrator does not have to program AT entries for any message that requires a command or reply or indicates a state change.

For example, deleting an Application from the policy also deletes the AT entries. The system administrator does not have to search for the AT entries.

- The SA z/OS agent uses the output from the PDB build to dynamically load the generated AT under the name of INGMSG02 when the SA z/OS agent initializes.
- AT entries for user Applications are also created by SA z/OS when the configuration data is built.

## INGMSG02 dynamic loading



- Issue **BR INGMSG01** to view the AT contents
- The AOFMATLISTING AAO controls whether an AT listing file is created. If generating AT listings, you allocate the DSILIST data set as a PDSE with a large amount of space
- The AOFSMARTAT AAO controls when the AT and MRT are refreshed

### INGMSG02 dynamic loading

Dynamic Automation Table building uses policy in the PDB to dynamically create and load an AT. The dynamic AT is included as part of INGMSG01 and written to the DSILIST data set. The working AT members (ACFMZxxxx, ACFXZxxxx, ACFEZ9999) are placed in the build output data set. They are dynamically loaded by the SA z/OS agent as INGMSG02 during agent initialization (included in INGMSG01). INGMSG02 is only in storage during Automation Table activation or refresh. To see the Automation Table that was loaded, you can issue a NetView **BROWSE INGMSG01** command.

## INGMSG02 content

- Predefined message automation
  - Entries that are required for automation to work regardless of which resources are running
- Many of the well-known product component messages are defined in INGMSG02
- Messages that are defined in the MESSAGES/USER DATA policy item, as follows:
  - C (CMD): Issue command in response to a message
  - R (REP): Issue a reply to a WTOR message
  - A (AT): Set status or modify AT entry:
    - AS: Define status messages for UP, ACTIVE, TERMINATING, TERMINATED, HALTED, ABENDING, ABENDED, BREAKING, BROKEN
    - AO: AT entries, as built by default, can be overridden by the automation administrator
  - M (MRT): Modify message revision table entry
  - K (COD): Issue command based on text with a message

### INGMSG02 content

INGMSG02 is the AT that is used by SA z/OS message processing. INGMSG02 uses LABEL statements to enable and disable code segments dynamically. With this method, only the product component messages that are defined for automation in the PDB are actively processed. For instance, if you are not running or automating Information Management System (IMS), IMS message processing is disabled. INGMSG02 provides automation for many products and components, such as TSO, NetView SSI, NetView Application, automation manager, TCP/IP, RMF, Tivoli Workload Scheduler, MQ, DB2, IMS, CICS, OMEGAMON, WebSphere Application Server, and GDPS.

You can use CODEs and user actions for messages to generate AT entries dynamically. If the CODE or user action is for a fully expressed message that is well-known to SA z/OS, then it generates an entry. Otherwise, the SA z/OS administrator must provide the command logic and AT entries to process CODE and USER actions for messages. You can determine if a message is well-known to SA z/OS by searching for it under the MVS Components +SA\_PREDEFINED\_MSGS policy. SA z/OS automates over 600 predefined messages.

## MESSAGES/USER DATA: AT statements

| AOFGDYNM  | Message Processing            | Line 00000001 Col 001 075 |
|---|-------------------------------|---------------------------|
| Command ==>   |                               | Scroll ==> CSR            |
| Entry Type : Application  | PolicyDB Name : OPSDEMO       |                           |
| Entry Name : RV01   | Enterprise Name : OPSDEMO     |                           |
| Line Commands: S/C (Cmd), R (Rep), K (Cod), U (Usr), A,M (AT, MRT, MPF) |                               |                           |
| I, D (insert or delete lines), CN (Class Name)                          |                               |                           |
| Message ID field length . 16 (1 - 32)                                   |                               |                           |
| Cmd Message ID  | Description                   | C R K U A M F             |
| IEF404I   |                               | P                         |
| VAPL2TEST   | Test msg for example only     | *                         |
| VAPL21000A  | Specifications for VAPL21000A | 3                         |
| VAPL21010I  |                               | *                         |
| WTORS   | Specifications for WTORS      | 2 P P P                   |
| ***** Bottom of data *****  |                               |                           |

NetView Automation Table statements:

```
IF MSGID = 'VAPL21000A' THEN
  EXEC(CMD('ISSUEACT') ROUTE(ONE %AOFOPGSSOPER%));
IF MSGID = 'VAPL21010I' THEN
  EXEC(CMD('ACTIVMSG UP=YES') ROUTE(ONE %AOFOPGSSOPER%));
```

© Copyright IBM Corporation 2019

5-80

### MESSAGES/USER DATA: AT statements

The ISSUEACT and ACTIVMSG commands are examples of *generic routines* that read and interpret the PDB and make decisions based on the policy data. ISSUEACT merges the function of the ISSUECMD and ISSUEREP commands to retrieve the command or reply policy for a message. You can find examples of generic routines on [page 415](#) and additional information in the *IBM System Automation for z/OS: Programmer's Reference* manual.

## MESSAGES/USER DATA: AT and MRT actions

```
COMMANDS VIEW HELP
AOFGMADV Message Automation Overview Scroll ===> CSR
Command ===> ■
AC AT condition      MS MRT action selection      MF MPFLSTxx specifications
AS AT status          MO MRT override           IG Ignore for AT, MRT, MPF
AO AT override

Message ID : VAPL21010I

Generated AT entry :
IF MSGID = 'VAPL21010I' THEN
  EXEC(CMD('ACTIVMSG UP=YES')) ROUTE(ONE %AOFOPGSSOPER%);

Generated MRT entry :
UPON ( MSGID = 'VAPL21010I' )
REVISE('Y' AUTOMATE)

Generated MPF entry :
VAPL21010I
```

Dynamic AT and MRT entries for VAPL21010I message

- AO: Override (modify) AT statement
- AC: Define automation for mixed-case messages
- MO: Override (modify) MRT statement
- AS: Define Automation Status and, optionally, Health Status

### MESSAGES/USER DATA: AT and MRT actions

In this case, the message VAPL21010I is defined as the UP message for RV01. The VAPL21010I AT entry is automatically created by the build process.

## MESSAGES/USER DATA: AT and MRT overrides

- **AO** action: Opens ISPF edit window to modify the AT statement to be built

```
ISREDDE2   SAZOS.PDB.OPSDCOPY(INGC1021) - 01.01          Columns 00001 00072
Command ===> _____                                         Scroll ===> CSR
***** * ***** * ***** * ***** * ***** * ***** * ***** * Top of Data * ***** * ***** * ***** *
000100 IF MSGID = 'VAPL21010I' THEN
000200 EXEC(CMD('ACTIVMSG UP=YES')) ROUTE(ONE %AOFOPGSSOPER%));
***** * ***** * ***** * ***** * ***** * ***** * ***** * Bottom of Data * ***** * ***** * ***** *
```

- **MO** action: Opens ISPF edit window to modify the MRT statement to be built

```
ISREDDE2   SAZOS.PDB.OPSDCOPY(INGC1021) - 01.01          Columns 00001 00072
Command ===> _____                                         Scroll ===> CSR
***** * ***** * ***** * ***** * ***** * ***** * ***** * Top of Data * ***** * ***** * ***** *
000100 UPON ( MSGID = 'VAPL21010I' )
000200 REVISE('Y' AUTOMATE)
***** * ***** * ***** * ***** * ***** * ***** * ***** * Bottom of Data * ***** * ***** * ***** *
```

### MESSAGES/USER DATA: AT and MRT overrides

Using the **AO** action requires NetView Automation Table programming skills. Test all code fragments before implementing them with EMM. Using the **MO** action requires NetView message revision table programming skills. The Automation Table is case-sensitive. Code everything in uppercase characters except mixed-case message texts, for example. Syntax checking can be enabled for overrides and is done in the NetView address space. Messages related to the syntax check are returned in the ISPF customization dialog.



**Note:** All Automation Table entries can be modified, including those for predefined messages. While modifying predefined message automation is rare, be careful not to break automation when you modify it.

## MESSAGES override syntax check

Press F3 in the ISPF edit window to invoke the syntax check.

The screenshot shows an ISPF edit window with the following details:

- Title bar: AOFGMADV - Message Automation Overview
- Buttons: Confirm Cancel
- Section header: AOFGM054 - AT Syntax Check Result
- Text area:

```
CNM521E INVALID ACTION NAME "EXEK"
DSI417I #0000002 : IF MSGID = 'VAPL21010I' THEN EXEK(CMD('ACTIVMSG UP=YES')
ROUTE(ONE * AUTGSS AUTSYS AUTOBASE AUTO1));
```
- Message: DW0525I TEST OF NETVIEW AUTOMATION FILE WAS UNSUCCESSFUL
- Instructions at the bottom:

```
Press ENTER to keep override as is.
Press END or CANCEL to continue editing.
```

### Message override syntax check

After you press F3 in the ISPF editor, a syntax check is done in the NetView address space. A result is returned to the ISPF customization panel.

# Status message reports

```
AOFGREPK                               Report Selection Menu
Option ==> __

  1 Report      Create Policy Database Report
  2 Unlinked    Create List of entries not linked in Policy Database
  3 ViewUnlinked View List of unlinked entries

  4 Resources   Create List of generated Resources
  5 ViewResources View List of generated Resources

  6 MemberList   Create List of unused members in configuration data set
  7 ViewMemberList View List of unused members

  8 StatusMsgs   Create Status Messages Report
  9 ViewStatusMsgs View Status Messages Report

  A Symbols     Create List of Symbols (AOCCLONEx) in Policy Database
  B ViewSymbols  View List of Symbols
```

© Copyright IBM Corporation 2019

5-84

## Status message reports

StatusMsgs (option **8**) creates a report of all messages that are defined to indicate status changes for an Application. ViewStatusMsgs (option **9**) displays a report of all messages that indicate status changes created by option 8.

## Status message report example

| ISREDDE2 SAZOS.PDB.OPSDCOPY.REP(RPTSMMSG) - 01.00 |           |  | CHARS 'RV01' found |
|---|-----------|--|--------------------|
| Command ==>                                       |           |  | Scroll ==> CSR     |
| 000445 TPX992I                                    | !UP       | !TPX is available                      | +SA_PRED           |
| 000446 VAPL21010I                                 | !UP       |  | !RV01              |
| 000447 VAPL22010I                                 | !UP       |  | !RV02              |
| 000448 VAPL23010I                                 | !UP       |  | !RV03              |
| 000449 VAPL24010I                                 | !UP       |  | !RV04              |
| 000450 VAPL25010I                                 | !UP       | !UP message                            | !RV05              |
| 000451 VAPL26010I                                 | !UP       |  | !RV06              |
| 000452 VAPL27010I                                 | !UP       |  | !RV07              |
| 000453 VAPL28010I                                 | !UP       |  | !RV08              |
| 000454 VAPL29010I                                 | !UP       |  | !RV09              |
| 000455 VPS000N                                    | !UP       | !VPS is available (V1R8)               | +SA_PRED           |
| 000456 VPS000N                                    | !UP       | !VPS is available (V2R1)               | +SA_PRED           |
| 000457 VPS003N                                    | !ACTIVE   | !VPS is starting (V1R8)                | +SA_PRED           |
| 000458 VPS122E                                    | !ABENDED  | !VPS terminates due to VTAM (V2R1)     | +SA_PRED           |
| 000459 VPS0300E                                   | !ABENDED  | !VPS ACB open failed (V2R1)            | +SA_PRED           |
| 000460 VPS0301E                                   | !ABENDED  | !VPS was unable to inform VTAM (V2R1)  | +SA_PRED           |
| 000461 VPS122E                                    | !ABENDING | !VPS terminates due to VTAM (V1R8)     | +SA_PRED           |
| 000462 VPS300E                                    | !ABENDING | !VPS ACB open failed (V1R8)            | +SA_PRED           |
| 000463 VPS800E                                    | !ABENDING | !VPS Dispatcher subtask abended        | +SA_PRED           |
| 000464 VPS801E                                    | !ABENDING | !VPS Communication subtask abended     | +SA_PRED           |
| 000465 VPS802E                                    | !HALTED   | !VPS LOG subtask abended (V1R8)        | +SA_PRED           |
| 000466 VPS803E                                    | !ABENDING | !VPS Driver subtask abended            | +SA_PRED           |
| 000467 VPS804E                                    | !ABENDING | !VPS MRD subtask abended               | +SA_PRED           |
| 000468 VPS805E                                    | !ABENDING | !VPS Timer subtask abended             | +SA_PRED           |
| 000469 VPS806E                                    | !ABENDING | !VPS TCPIP main system subtask abended | +SA_PRED           |
| 000470 VPS812E                                    | !ABENDING | !VPS Server connect subtask abended    | +SA_PRED           |

| Message ID | Status | Description |
|------------|--------|-------------|
|------------|--------|-------------|

© Copyright IBM Corporation 2019

5-85

### Status message report example

This slide shows an example of the status messages (StatusMsgs) report. For example, VAP2010I is the UP message for application RV01. This report displays the messages predefined by SA z/OS as well as those defined in the PDB. The StatusMsgs report contains five columns:

- Message ID
- Status
- Description
- Entry Name
- Typ (entry type)

The width of the columns is dependent on the data being displayed. All messages are listed that were defined in the PDB. Predefined messages, such as IEF404I, are identified as +SA\_PREDEFINED\_MSGS. You can use PF10 and PF11 to scroll right and left to see all of the data.

# Lesson 7. Advanced customization topics

## Lesson 7: Advanced customization topics

- Many user exits can be coded
- Many AAOs can be used to control processing
- Generic routines are also provided

### What this lesson is about

This lesson describes the purpose of several advanced customization features including user exits and advanced automation options.

### Objectives

After completing this lesson, you should be able to describe the purpose of several advanced customization features such as user exit, generic routines and advanced automation options.

### References

*SC34-2717 System Automation for z/OS Version 4  
Release 1 Defining Automation Policy*

## SA z/OS exits

- **Initialization exits:** Invoked after SA z/OS starts its various tasks, but before the primary Automation Table is loaded
- **Environmental Setup Exits:** defined in SYSTEM INFO policy
- **Static exits:** Invoked at fixed points during SA z/OS processing
- **Flag exits:** Can be used to branch to other products to determine processing
- **Command exits:** Invoked during command processing
- **Customization dialog exits:** Invoked during phases of the customization dialog processing

SDF exit AOFEXX05 that allows the installation to replace user variables that are defined in the SDF tree and panel definitions

Based on the system for which the tree and panels are generated

Exits are documented in the *IBM System Automation for z/OS 4.1: Customizing and Programming manual*

© Copyright IBM Corporation 2019

5-87

### SA z/OS exits

On this slide, you see several types of SA z/OS exits. In most cases, the exits are REXX routines that are called by SA z/OS. Samples are provided in the SINGSAM library. Be careful of how often exits are invoked; it can cause a performance problem. More information about all of the user exits is in the *IBM System Automation for z/OS 4.1: Customizing and Programming manual*.

### Initialization exits

Initialization exits that are called at the start of SA z/OS initialization, before message AOF603D is issued.

Descriptions of the SA z/OS initialization exits are listed in the following table.

| Exit Name | Description  |
|-----------|--|
| AOFEXDEF  | Called at the start of SA z/OS initialization, before message AOF603D is issued. |
| AOFEXI01  | Invoked before the reply to AOF603D ENTER AUTOMATION OPTIONS is issued.          |

| Exit Name | Description  |
|-----------|--|
| AOFEXI02  | Invoked after the operator replies to the AOF603D reply.                                     |
| AOFEXI03  | Invoked before SA z/OS loads NetView AT.   |
| AOFEXI04  | Invoked after SA z/OS loads NetView AT.  |
| AOFEXI05  | Invoked before either an ACF load or refresh takes place.                                    |
| AOFEXI06  | Invoked after an ACF load or refresh has completed and before the AOF540I message is issued. |
| AOFEXINT  | Invoked when SA z/OS initialization is complete, before message AOF540I is issued.           |

## Environmental exits

Environmental exits have these characteristics:

- Defined by using the SYSTEM INFO policy item of the SYStem policy object.
- Invoked after SA z/OS starts its various tasks, but before the primary Automation Table is loaded.
- Invoked after SA z/OS initializes its control structures.
- Used to initiate your own automation.
- Executed under automation operator AUTINIT1.
- These exits are not driven if you run RESYNC.
- Unlike the other static exits, you must specify the name of the routine or routines to invoke in the automation control file.

## Static exits

Static exits have these characteristics:

- Always invoked if they are in the DSICLD concatenation.
- Positive return codes from these exits are typically ignored, although you must always code your exit to end with a return code of zero.

The main purpose of static exits is for you to take your own actions at specific points during SA z/OS processing.

| Exit Name | Description   |
|-----------|---|
| AOFEXSTA  | Called from AOCUPDT every time the automation status of an application is updated   |
| AOFEXX02  | Decide whether or not an SDF update should be performed for the specified resource. |
| AOFEXX04  | Called from CHKTHRES to check error thresholds.                                     |
| AOFEXX05  | replace user variables in the SDF panel definition at initialization time.          |

The SDF exit AOFEXX05 is started by System Automation for z/OS at initialization when setting up the SDF panels or by means of the RESYNC SDFDEFS command. The exit is used by the installation to replace user variables in the SDF panel definition. A user variable must follow the same convention as a z/OS system symbol; it must start with an ampersand (&) and finish with a dot (.). An example is &MVDOMAIN. Refer to the sample exit for details of the parameters that are passed to the exit and the return codes.

## Flag exits

Flag exits are called when SA z/OS must evaluate an automation flag. They cause normal SA z/OS processing to use data from an external source, such as a scheduling function, to determine whether automation is on or off for a given resource at that particular instant.

Flag exits can be defined for these items:

- Any flag (Automation, Initstart, Start, Recovery, Shutdown, or Restart).
- Any resource.
- Any minor resource. See the description of the policy item MINOR RESOURCES in *IBM System Automation for z/OS: Defining Automation Policy* for more information about minor resources.

You can specify multiple exits for each flag. A flag exit is invoked only if SA z/OS must have an *opinion* on the current flag setting. Flag exits and flags work on a *veto* basis. A flag is ON when all flags and flag exits agree that it is on. Flags are set to YES, NO, or EXIT. Exits are called only when the flag setting is EXIT.

## Command exits

Command Exits can be called during the processing of certain commands, such as INGREQ, INGSCHED, DISPINFO, DISPSTAT, INGLIST, INGVOTE, or INGSET, SETSTATE, INGTHRES,

INGALERT, INGRGDPS, INGGROUP, INGMOVE, INGCICS, INGIMS, INGOPC, INGEVENT, INGSUSPD, INGAMS, INGRUN, INGLKUP.

## ***Customization dialog exits***

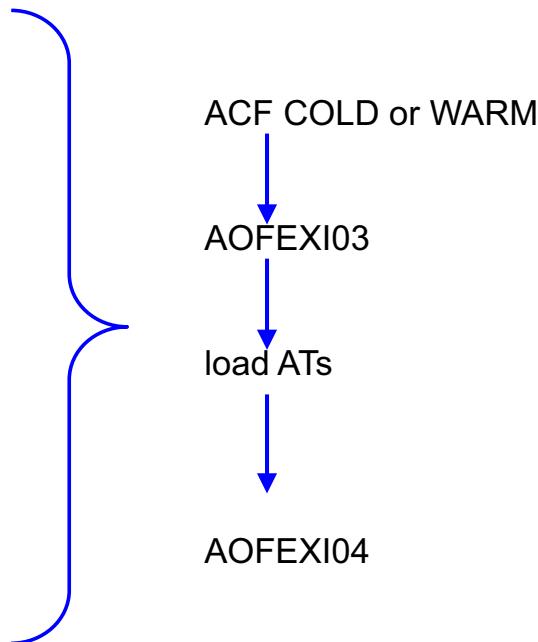
Customization dialog exits are invoked during the customization dialog processing for stages, for example:

- BUILD
- DELETE
- COPY
- CONVERT
- REPORT
- RENAME
- IMPORT

## AT load exits: Example

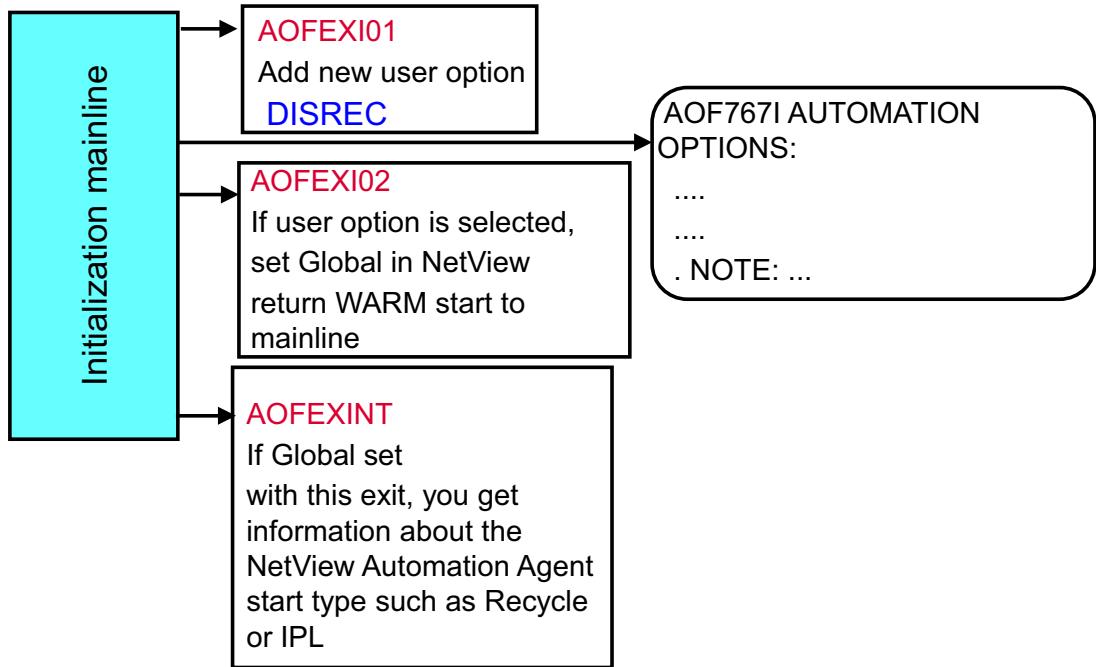
User exits during AT load:

- AOFEXI03
  - Save existing AT listings
  - Save AT statistics (AUTOCNT)
  - Save SA z/OS statistics (INGRPT)
  - Load Automation Tables
- AOFEXI04
  - Reset INGRPT statistics



This diagram depicts the sequence of events during the loading of the Automation Table and the various SA z/OS exits that are involved.

## Initialization exits example



© Copyright IBM Corporation 2019

5-89

### Initialization exits example

This diagram shows how an installation can use the exits to add installation-written initialization to the overall SA z/OS initialization process. In the diagram, the operator sees an additional option on the AOF767I message. The operator can provide additional installation-specific keywords as a response to the AOF603D WTOR message. The AOFEXI01 exit has control before the AOF767I and AOF603D messages are issued. It receives the multi-line AOF767I message that SA z/OS agent initialization composed, in the default SAFE passed to the routine. Exit processing can add, delete, or edit lines in the AOF767I multi-line message before it is shown to the operator. These lines can document additional options (such as DISREC, shown on the slide) and provide notes about their use to the operator. The AOFEXI02 exit is invoked after the operator replies to the AOF603D WTOR. The exit is passed the operator response, which can be modified to remove, add, or change values before further processing by SA z/OS initialization. One example might be to provide a specialized *disaster recovery* start option called DISREC. This requirement is common where an installation might want to initialize only certain base subsystems (TSO, VTAM) and delay initialization of others until system updates are completed by using TSO. Another use might be to prevent operators from overriding default WARM options.

## Example command exit: AOFEXC01

- Installation can filter or modify INGREQ requests
- Exit starts after INGREQ parameter verification and before issuing the request
- Nonzero return code causes INGREQ panel to be redisplayed with message AOF227I or exit-supplied message

AOF227I UNABLE TO PROCESS INGREQ - reason

### *Example command exit: AOFEXC01*

AOFEXC01 is a command exit that is used when INGREQ is called. With AOFEXC01, an installation can filter or modify INGREQ requests. For example, you can use AOFEXC01 to restrict operator access to critical resources.

## Advanced automation options (AAOs)

- Common global variables are useful to control the functions of SA z/OS and define command parameter default settings
  - Can be binary: Enabled or disabled
  - Can contain user-defined defaults

© Copyright IBM Corporation 2019

5-91

### *Advanced automation options (AAOs)*

Advanced Automation Options (AAOs) are NetView common global variables that can be set to control how SA z/OS functions. They are either enabled or disabled. There are two categories of AAOs:

- *Behavioral* for controlling the behavior of SA z/OS automation, for example, AOFINITREPLY.
- *Installation defaults for commands parameters*, for example, INGREQ\_REMOVE.

The AAOs that are used to control SA z/OS functions are in the AOFSTYLE member in SINGSAMP. Be careful when you use a RESTYLE COMMON command when you have AAOs defined in the NetView style sheet. You must issue AOFEXDEF separately every time you use a RESTYLE COMMON. All AAOs are documented in the *IBM System Automation for z/OS: Customizing and Programming* manual.

## AAO examples

| Variable                | Use   |
|-------------------------|---|
| <b>AOFCNMASK</b>        | Determines names of EMCS consoles that are acquired   |
| <b>AOFINITREPLY</b>     | Timeout value for replying to AOF603D message   |
| <b>AOFMATLISTING</b>    | Controls creation of Automation Table listing   |
| <b>AOFSHUTDELAY</b>     | Delay, in minutes, for SA z/OS to wait for a termination message before continuing the shutdown process |
| <b>AOFSETSTATESCOPE</b> | Default SCOPE value for SETSTATE command  |
| <b>INGREQ_REMOVE</b>    | Default AUTOREMOVE value for INGREQ command   |
| <b>AOFSMARTMAT</b>      | Defines if AT and MRT must be loaded or refreshed when ACF is   |

© Copyright IBM Corporation 2019

5-92

### AAO examples

This slide shows a subset of the Advanced Automation Options. The full set of AAOs are documented in the *IBM System Automation for z/OS: Customizing and Programming* manual. You can set AAOs in the AOFEXDEF EXEC or the NetView style sheet.

In this example, AOFMATLISTING AAO is set by using **AOFEXDEF**. Add these two lines to the AOFEXDEF exit:

```
AOFMATLISTING = '0'  
'GLOBALV PUTC AOFMATLISTING'
```

In this example, the style sheet is used to set the AOFMATLISTING AAO:

- \* AT listing
- \* 1 - AT listing is created for all ATs loaded by SA (recommended)
- \* 0 - Disable creation of AT listings

```
COMMON.AOFMATLISTING = 1
```

## AAO example: AOFINITREPLY

### AOFINITREPLY value

- 00:02 AOF603D is issued with a timeout of two minutes (default)
- hh:mm:ss AOF603D is issued with a timeout of hh hours, mm minutes, and ss seconds
- 0 AOF603D is not issued; automation continues with SA z/OS default settings

#### Example:

*nn* AOF603D ENTER AUTOMATION OPTIONS OR 'R' (RE-DISPLAY) - DOMAIN AOFDA

### AAO example: AOFINITREPLY

You can set the **AOFINITREPLY** Advanced Automation Option (AAO) to a value of zero (**0**) to not issue the AOF603D message and use the default actions indicated in the AOF767I message. Operators have no opportunity to modify the SA z/OS agent initialization. You can set the AOFINITREPLY global variable in AOFEXDEF or CNMSTGEN. The default value is two minutes. Any NetView time format, including seconds, is accepted.

- hh:mm:ss

The initial reply AOF603D is issued and automatically responded after hh:mm:ss.

## Example generic routines

- AOCQRY: Verifies automation is allowed for a resource
- ISSUEACT: Issues replies or commands defined in the MESSAGES/USER DATA policy
  - Merges ISSUECMD and ISSUEREP
- ACFREP: Procedure can issue replies defined in the automation policy
- ACFCMD: Procedure can issue commands defined in the automation policy
- AOFCPMSG: You can process the captured messages.
  - Also saves important messages
- CDEMATCH: Retrieves the MESSAGES/USER DATA CODEs policy

© Copyright IBM Corporation 2019

5-94

### Example generic routines

In this list are several commonly used generic routines. All supported routines are in the *IBM System Automation for z/OS: Programmer's Reference* manual. In the previous lesson, you saw how to create Automation Table entries for a message (VAPL21000A) that issued the ISSUEACT command:

```
IF
  MSGID = 'VAPL21000A'
  THEN
    EXEC(CMD('ISSUEACT')) ROUTE(ONE %AOFOPGSSOPER%) ;
```

# Student exercises

## Unit 5

### 1. Additional customization:

- Define application classes
- Define application RV01
- Define service period, event, and trigger
- Use a flat file to create Applications RV02, RV03, and RV04



© Copyright IBM Corporation 2019

*Student exercises*

Perform the exercises for this unit.

## Summary

Now that you have completed this unit, you should be able to perform the following tasks:

- Define Applications by using SA z/OS symbolics
- Define Application classes and linked instances
- Implement planned suspend capability
- Define Automation setup for Notify Operators, Automation Timers, MVS Automation, JES spool management, and Automation Operators
- Define Service Periods, Triggers, Events, and Pacing Gates
- Define monitor resources
- Describe the SA z/OS dynamic Automation Table build feature

© Copyright IBM Corporation 2019

### Summary

Now that you have completed this unit, you should be able to perform the following tasks:

- Define Applications by using SA z/OS symbolics
- Define Application classes and linked instances
- Define Automation setup for Notify Operators, Automation Timers, MVS Automation, JES spool management, and Automation Operators
- Define Service Periods, Triggers, Events, and Pacing Gates
- Define monitor resources
- Describe the SA z/OS dynamic Automation Table build feature





## 6 Initialization and runmodes

---



### Unit 6 Initialization and runmodes



© Copyright IBM Corporation 2019  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

---

**What this unit is about**

In this unit, you learn about the initialization and synchronization of the automation manager and automation agent. You have several options for initializing each component. You also learn about refreshing the automation configuration file and managing the primary automation manager. The concept of runmodes, as it relates to system initialization, is also covered. In this unit, you learn about the initialization and synchronization of the automation manager and automation agent. You have several options for initializing each component. You also learn about refreshing the automation configuration file and managing the primary automation manager. The concept of runmodes, as it relates to system initialization, is also covered.

**How you check your progress**

You can check your progress in the lab exercises.

**References**

*SC34-2717 System Automation for z/OS Version 4*

*Release 1 Defining Automation Policy*

[https://www.ibm.com/support/knowledgecenter/SSWRCJ\\_4.1.0/com.ibm.safos.doc\\_4.1/manuals.html](https://www.ibm.com/support/knowledgecenter/SSWRCJ_4.1.0/com.ibm.safos.doc_4.1/manuals.html)

# Objectives

---

After completing this unit, you should be able to perform the following tasks:

- Describe the automation environment
- Describe automation manager initialization
- Describe automation agent initialization
- List and explain the manager and agent start options
- Define and implement runmodes

## *Objectives*

After completing this unit, you should be able to perform the following tasks:

- Describe the automation environment
- Describe automation manager initialization
- Describe automation agent initialization
- List and explain the manager and agent start options
- Define and implement runmodes

# Lesson 1. Structure and availability

## Lesson 1: Structure and availability

- The automation manager environment
- The automation agent environment
- Sysplex environment
- Automation manager availability

© Copyright IBM Corporation 2019

6-3

### What this lesson is about

This lesson describes the Initialization options.

### Objectives

After completing this lesson, you should be able to use Initialization options.

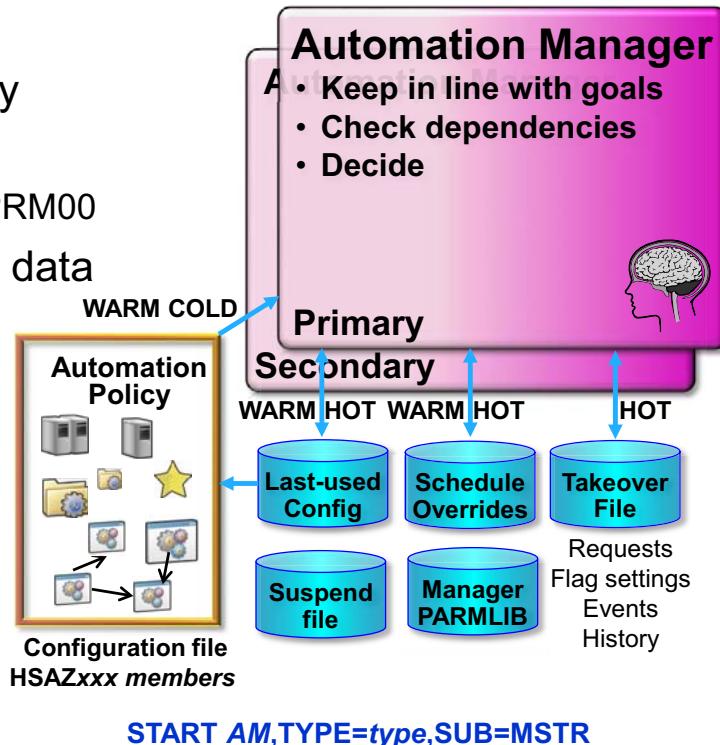
### References

*SC34-2717 System Automation for z/OS Version 4  
Release 1 Defining Automation Policy*

## The automation manager environment

Shared data sets:

- Configuration file = Policy
- Manager PARMLIB
  - Initialization member HSAPRM00
- Takeover File: persistent data
- Schedule Overrides
- Name of last-used configuration file
- Suspend file



© Copyright IBM Corporation 2019

6-4

### The automation manager environment

The automation manager is a z/OS started task. It can be started either from the command line, with the z/OS start command, or automatically at IPL, with an entry in the COMMNDxx member. The default procedure name is AUTOMGR.

This slide shows several of the data sets that are required by the automation manager:

- HSAPLIB: The DD name of the automation manager parameter library. This data set contains at least one initialization member which is named HSAPRM with a two-character suffix. The default name of the initialization member is HSAPRM00. In this member, there are statements that define how an automation manager initializes and operates. An example is the name of the configuration files data set. The default initialization member in HSAPLIB can be overridden by a command-line start command, or within the start procedure JCL. The example below shows the selection of member HSAPRM01.

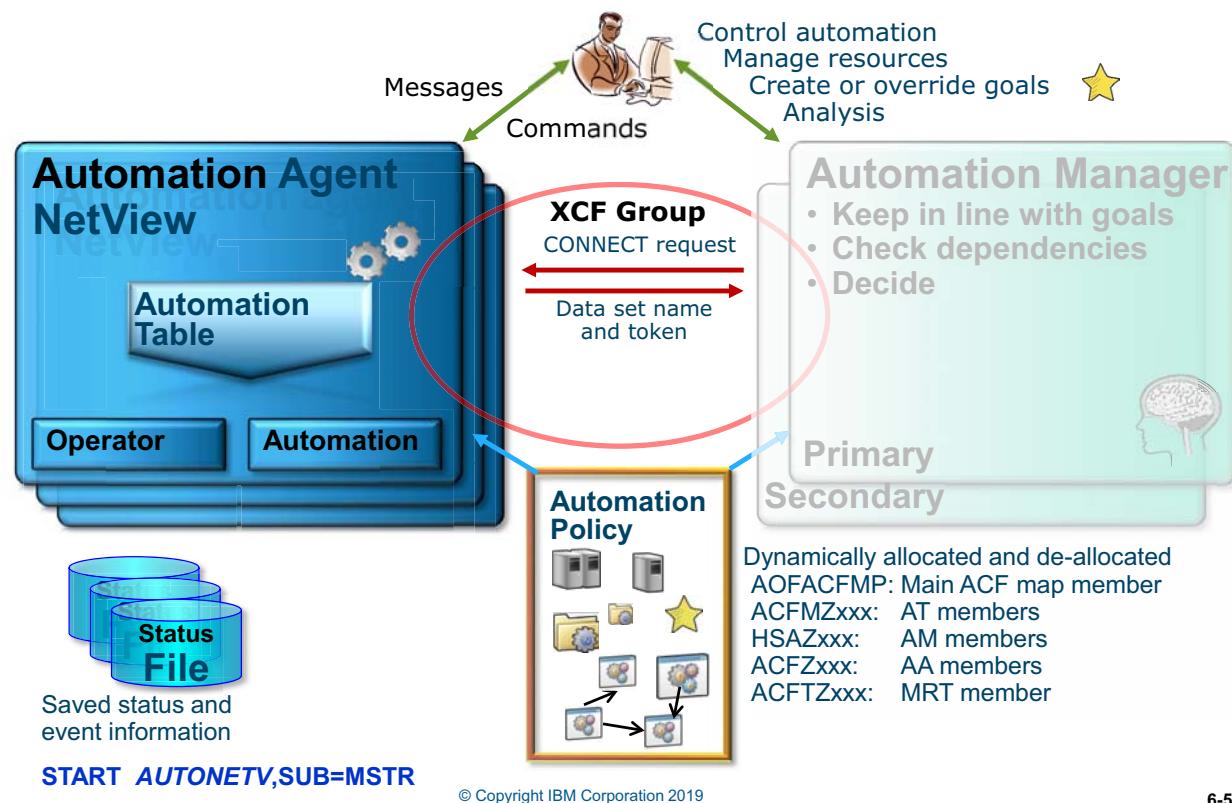
START AUTOMGR, M=01

- HSAOVR: The DD name of the schedule override file. HSAOVR points to a VSAM data set which contains overrides (schedule and resource) to defined service periods. The information in this data set is used for WARM or HOT starts of the automation manager.

- Takeover file: The name of this data set is specified in HSAPRMxx. The takeover file is a VSAM data set that contains persistent data that is used by the manager during HOT starts of the automation environment. Persistent data includes information such as votes, events, automation manager flag changes, history, and *application group* changes.
- HSACFGIN: This DD name points to a data set that contains the name of the last-used automation manager configuration file. The last-used name is updated whenever the automation manager is started COLD, and when the configuration files are refreshed during WARM and HOT starts.

The box at the base of the slide represents the automation manager configuration (AMC) file. The build option in the customization dialog generates the automation configuration file members from information in the System Automation for z/OS policy database. Many customers use generation data groups (GDGs) for the automation configuration file. Using GDGs provides for a backup copy of the files. During customization, the value of the SCOPE parameter in the customization dialog specifies whether to generate the NetView automation table for a single systems, a sysplex, or an enterprise.

# The automation agent environment



6-5

The automation agent environment

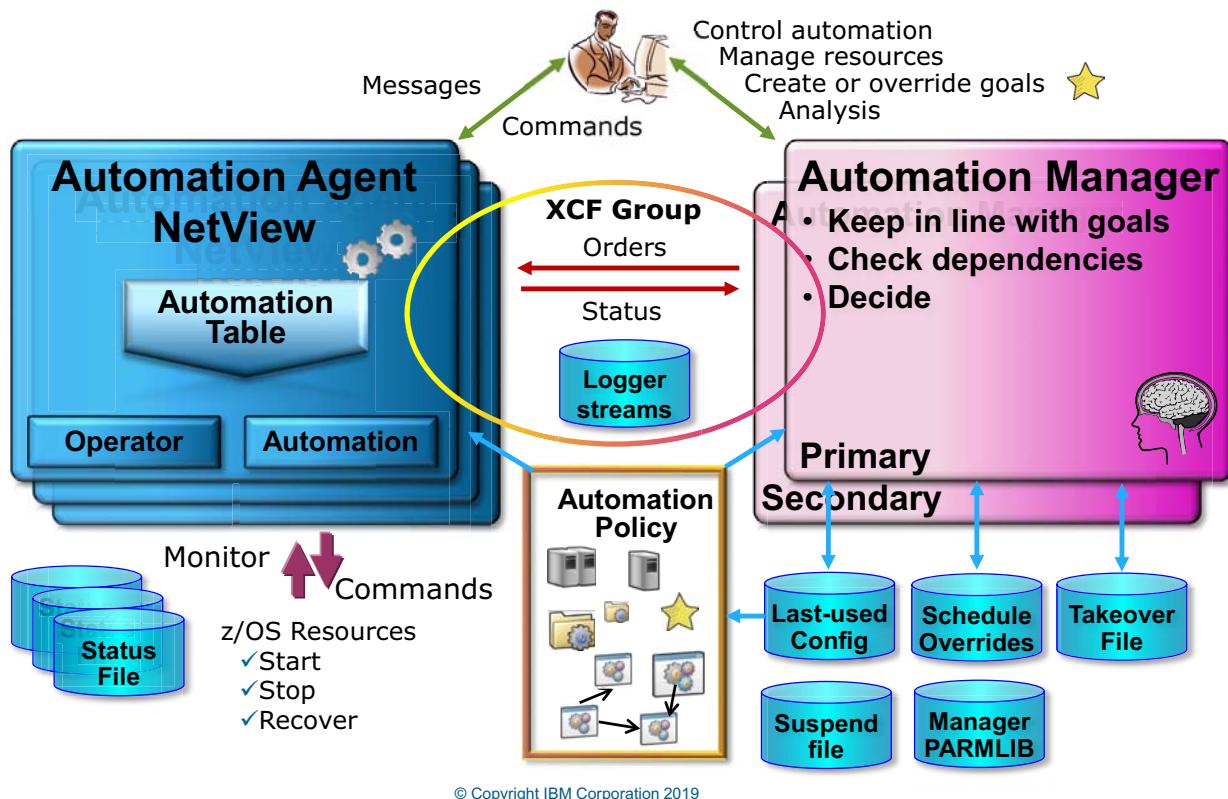
The automation agent runs in NetView. NetView is started with the SUB=MSTR parameter. The name of the supplied NetView start procedure is INGENVSA. When an agent initializes, it connects with its primary automation manager and receives the name of the manager configuration file and a token. The agent uses this information to locate its automation control files in the configuration data set, and to ensure that it is consistent with the manager files. The automation agent and manager communicate through z/OS cross-system coupling facility (XCF).

The following data sets are used by the agent:

- The configuration file holds the automation configuration file that is read by the agent when it initializes. Typically, this file is separate and dynamically allocated by the agent while it is required, and then deallocated.
- The automation status file is a VSAM data set that contains information saved by the agent, including status and error events.

The NetView automation table and optionally the NetView message revision table are loaded after the configuration file is processed.

## Sysplex environment



Sysplex environment model

6

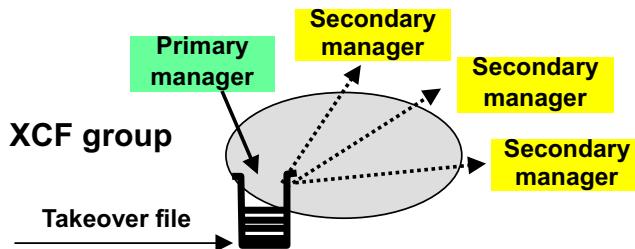
This slide shows a simple example of a sysplex configuration. One automation manager is running in each system in the sysplex. The diagram shows each agent accessing its own automation control files, automation table, and message revision table in the automation configuration file. Each agent must have its own automation status file.

The two managers share all the files as shown, or each automation manager can have its own HSAPLIB file. When a second automation manager starts in the sysplex group, it discovers that a manager is running, and it initializes as a secondary manager. In this case, it does not matter what type of initialization (WARM, HOT, or COLD) was requested. The secondary automation manager performs no processing (no status is received from agents, no current information is maintained). It allocates the same files. However, it does not read from any files except the HSAPLIB file. It waits to take over when requested or needed.

The secondary automation manager is shown here in a different system. Additional automation manager address spaces can be started in the same system as the primary manager. The situation is essentially unchanged.

# Automation manager availability

- Primary automation manager makes automation decisions
- Secondary automation managers wait to become primary
- A takeover means that automation decisions are delayed but not lost
  - Elapsed time depends on the system load
- Event notification is a z/OS service
  - XCF SYSGONE or XCF MEMBER LEAVE
- Typically, no interaction occurs between the primary automation manager and the secondary automation manager
  - Secondary automation managers do not affect the workings of primary automation managers
  - Data for restart is saved on a shared disk



© Copyright IBM Corporation 2019

6-7

## Automation manager availability

If the primary automation manager fails, a secondary automation manager takes over. The secondary automation manager performs a HOT start by reading the current persistent data from the HSATKOVF file and the schedule overrides from the HSAOVF file. No information is lost and the takeover is transparent to the agents, except for delays in agent-manager communication. Automation managers detect other managers because each joins the System Automation z/OS XCF Group at initialization. The default name of the XCF group is INGXSG. The XCF group name is defined in INGXINIT or HSAPRMxx. All automation managers in the same sysplex group join the same XCF group. XCF Group processing is a z/OS service that signals members of a group when specific events occur. The following two events are important:

- MEMBER LEAVE: This event occurs when a member is no longer available. For example, an automation manager address space terminates normally or abnormally.
- SYSGONE: This event is generated when a system leaves the sysplex, either normally abnormally. These signals are used by automation manager processing to determine whether an automatic takeover is necessary.

# Lesson 2. Initialization options

## Lesson 2: Initialization options

- Automation manager initialization options
  - HOT
  - WARM
  - COLD
- Automation agent initialization options
  - STOP
  - PAUSE
  - NOSTART
  - [RUNMODE=n] – Optional
  - CONTINUE

### What this lesson is about

This lesson describes how the managers and agents start and initialize. Start options are discussed, and the differences between them.

### Objectives

After completing this lesson, you should be able to use Initialization options.

### References

*SC34-2717 System Automation for z/OS Version 4  
Release 1 Defining Automation Policy*

## Manager initialization

- Checks for the primary automation manager
- If starting as primary, the start type is determined from one of the following methods:
  - The START command: S AM,TYPE=HOT,SUB=MSTR
  - The JCL procedure: EXEC PGM=HSAPINIT,PARM='START=HOT'
  - The HSAPRMxx parameter: START=HOT
- Defaults to a WARM start if no specific start-type information is available
- Takeovers are always HOT starts
  - No data is lost during a HOT start

### Manager initialization

When an automation manager starts, it first tries to determine its position. It checks for a primary automation manager. If a primary manager is running, the address space initializes as a secondary automation manager. In this case, it only joins the XCF group.

The primary automation manager can initialize in one of three ways: HOT, WARM, or COLD. How it initializes is determined by the following factors:

- The parameters on the START command
- The parameters within the JCL
- The value in the START= parameter in the HSAPRMxx member

HOT starts use existing data, such as data stored in the takeover file. There is no loss of data with a HOT start. A takeover is effectively a HOT start by the new primary automation manager.

Sometimes, the desired type cannot be used. For example, if a HOT start is requested and there is no valid information in the takeover file, then a WARM start occurs. The automation manager can be enabled for automatic restart manager (ARM).

## Manager start types

|   | HOT                 | WARM                 | COLD  |
|---|---------------------|----------------------|---|
| Configuration data set name                       | Last used DSN       | Last used DSN        | DSN comes from parmlib or from an operator PROMPT |
| Schedule overrides                                | Taken from last run | Taken from last run  | <b>Contents deleted</b>                           |
| Persistent data which includes votes and runmodes | Taken from last run | <b>Contents lost</b> | <b>Contents lost</b>                              |

© Copyright IBM Corporation 2019

6-10

### Manager start types

This slide shows the impact of the different start types. During a COLD start, the name of the configuration file is written to the data set specified in the HSACFGIN DD statement. After a WARM or HOT start, the agent manager reads this data set to locate the name of the appropriate configuration file (last used data set).

When schedule overrides are created, they are stored in the VSAM file that is identified in the HSAOVR DD statement. On WARM or HOT starts, this file is read and the schedule overrides from the previous automation manager are retained. On COLD starts, the VSAM file is reformatted and the old data is lost.

Persistent data (votes, automation flags, group settings) is stored in the VSAM takeover file and reinstated when HOT starts are done. With hot starts, active *runmodes* can also be carried over. On WARM or COLD starts, the persistent data file is reformatted and the old data is lost; runmodes are not carried over. When starting the automation manager with **PROMPT=YES**, a write to operator with reply (WTOR) is issued, requesting the name of a valid configuration data set name.

## Agent initialization sequence

- Contacts the primary automation manager
  - Receives the data set name and token
  - Dynamically allocates the automation control file unless it is preallocated
- Reads the automation configuration file main member and compares tokens from the primary automation manager
- Saves policy data and timestamps in CGlobals
- Issues AOF767I and AOF603D messages to an operator and processes the response
- Loads ACF information from the data set
- Loads the NetView automation table  
Syntax problem will stop initialization
- Loads the NetView message revision table (optional)

### *Agent initialization sequence*

At initialization, the agent receives the name of the configuration data set from the primary automation manager. The name of the automation control that the agent uses is in the automation configuration file map. Configuration file member AOFACFMP contains the name of the main automation control file member for the agent. The main control file contains the member names of the automation control files for the resources in the agent's NetView domain. The agent saves the policy data and timestamps in the cache.

System Automation z/OS then issues a write to operator (WTO) message with initialization options followed by a WTOR. The operator replies with one of the options, or System Automation z/OS continues initialization after a two-minute delay (the default setting).

After the WTOR is processed, all automation control file information from the automation configuration file on disk is read and analyzed. Each fragment is dynamically allocated and data is read and analyzed. The fragment is then released. All data is written into NetView CGLOBAL variables. The agent loads the NetView automation table, and optionally the NetView MRT referenced in the automation control file. After the automation configuration file is loaded, the agent registers itself with other agents in the same SAplex. The agent can send system automation commands to other agents in the SAplex. The agent also registers with the automation manager

and exchanges information with the automation manager. At this point the agent is initialized, and System Automation for z/OS is fully operational.

The RESYNC command causes System Automation z/OS to rerun selected parts of its initial status determination routines or to reinitialize parts of its environment setup. The command ACF ATLOAD, reloads the NetView automation table.

## Agent initialization options

```
AOF767I AUTOMATION OPTIONS: 000
. STOP      - CANCEL AUTOMATION
. PAUSE     - SUSPEND AUTOMATION
. NOSTART   - DO NOT AUTOMATE SUBSYSTEM STARTUP
. RUNMODE=x - SET RUNMODE (CURRENT *ALL)
. ENTER     - CONTINUE
*0003 AOF603D ENTER AUTOMATION OPTIONS OR 'R' (RE-DISPLAY) - DOMAIN AOFDA
```

- Messages at automation agent initialization
  - AOF767I: Lists valid responses to message AOF603D
  - AOF603D: WTOR - reply with agent initialization option
    - The typical response is a null reply - ENTER

This slide shows the form of the AOF767I message with the valid responses to the AOF603D WTOR. The AOF603D message contains the name of the domain that issued the message. This information can be important. The message is broadcast to all consoles and can drive automation in all NetView domains. Using the domain ID, simple automation table filtering is possible. In most cases, a null reply is used. Any invalid response causes the WTOR to be issued again.

If you do not enter a reply to the message, the system proceeds with the default response (null) at the end of the administrator-defined delay. A NetView common global variable controls how long initialization waits for the operator to reply to the AOF603D WTOR. The default value is two minutes. If you are not sure how to respond to the AOF603D message, reply with PAUSE. The agent reissues the AOF767I message with an AOF606D WTOR and waits. The AOF606D message is similar to the AOF603D message. Agent initialization is complete when you see message AOF540I.

To ensure a proper initialization, check that message AOF869I indicates no errors. You can also issue the INGAMS command to check that the agents are in status READY.

If there are errors with the automation table, you see message AOF715A and a request to enter the name of a valid automation table. If you know that you do not want to run the System Automation z/OS agent, enter STOP. Agent initialization does not take place. Recycle NetView if you want to run the agent. If you do not want System Automation z/OS to start any subsystems, even though the policy indicates, enter NOSTART as part of the response. No start is like having the automation manager hold flag set to YES, which might be useful when testing and can prevent initial problems.

# Lesson 3. Runmodes

## Lesson 3: Runmodes

- Named automation elements Runmode and Runtoken
- Loosely termed desired-availability tags
- Runmode can be set:
  - At initialization time as a reply to message AOF603D
  - With INGRUN command
- An active runmode causes the following actions:
  - Generation of stop request with SOURCE=INGRUN (DEFAULT) and priority “should be down” against sysname/SYG/sysname resource
  - STOP votes are propagated to all resources of that system and along active relationships to other systems
  - When a resource is (forced) qualified for a runmode, the stop vote is still visible but shown as not winning
- The runmode, \*ALL indicates no active runmodes; all resources in the SYG are unaffected

© Copyright IBM Corporation 2019

6-13

### What this lesson is about

This lesson describes the use of runmodes.

### Objectives

After completing this lesson, you should be able to describe and implement runmodes.

### References

*SC34-2717 System Automation for z/OS Version 4 Release 1 Defining Automation Policy*

Runmodes are a flexible way to control the availability of resources without the need to place explicit START or STOP requests against them or manipulate their automation flag. Runmode requests can be used to switch between different setup scenarios such as day shift versus night shift or weekday versus weekend. You can also use runmodes to IPL a system in a progressive fashion. You define and activate runmodes at the system level.

A runmode consists of named automation elements runmode and runtoken that are defined in the automation policy database. A runmode can be loosely termed a *desired-availability tag*. A runmode has one or more runtokens (character strings that are associated with runmodes) that are associated with it. These character string identifiers are called runtokens and are used to link runmodes to resources on a system.

By assigning the runtoken to a resource you request that a resource qualifies for the runmodes where that runtoken is listed. The same runtoken can be listed in one or more runmodes and one or more applications.

The goal of runmodes is to make qualified resources available and unqualified resources unavailable. A resource is considered qualified for one of the three following reasons:

- The qualification is forced for the resource using the INGRUN command.
- The resource is a member of a group that is forced to qualify.
- At least one of its runtokens matches a runtoken defined for the current runmode.

When a runmode is activated on a system either by a reply to message AOF603D or by means of the INGRUN command, an implicit stop request is generated against the system group (SYG) for that system. STOP votes are then propagated to all resources of that system.

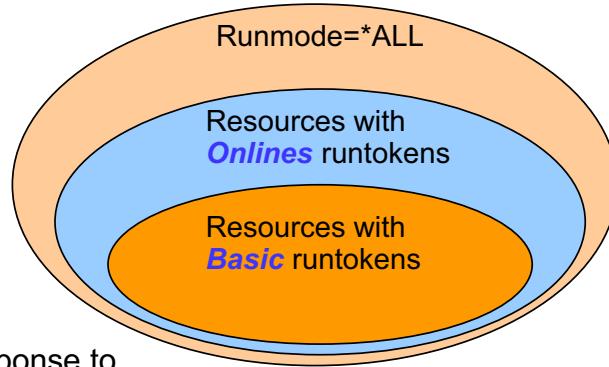
When a resource is (forced) qualified for a runmode, the stop votes are still visible but shown as not winning or as being propagated if it is the highest priority stop vote. The stop vote is propagated along active relationships to other systems if the resource has a relationship to resources in other systems. So, even though a runmode is set for a system, it can affect other systems that have no or different runmodes. The stop vote carries the runtokens.

The runmode, \*ALL indicates no active runmodes; all resources in the SYG are unaffected.

Resources are then made available (started) if they are linked to a runtoken that is associated with the active runmode. If a resource has no runtoken and one of the runtokens that is specified in the runmode definition is \*NULL, the resource is started.

## Runmode example

- A runmode can be established in two ways:
  - At initialization time as a reply to message AOF603D
  - With the **INGRUN** command
- Examples of runmode definitions:
  - Basic: B1 B2
  - Onlines: Cics1 Cics2 DB2a DB2b
  - Monitor: Omeg1
- Examples of resource runtokens:
  - VTAM: B1 Cics1 Omeg1
  - CICSPROD: Cics1
  - DB2PROD: DB2a
- Examples of setting the runmode in response to AOF603D:
  - **RUNMODE=ONLINES**  
This example qualifies resources that have an ONLINES runtoken.
  - **RUNMODE=\*ALL** -  
This example qualifies all resources



© Copyright IBM Corporation 2019

6-14

### Characteristics of runmodes

This slide shows the runmode information for a sample scenario that is intended to help clarify the concept of runmodes. There are two ways that you can establish a runmode in the running system after it is defined in the automation policy. You can respond to the AOF603D initialization message with the runmode name or you can run the INGRUN command to specify the runmode name in the automation NetView.

This example shows three explicit runmode definitions in the System Automation policy. A runmode is **basic** with runtokens of B1 and B2, a second runmode is **onlines** with runtokens of Cics1, Cics2, DB2a, and DB2b, and a third runmode is **monitor** with a runtoken called Omeg1. There is also a fourth runmode that is provided by System Automation with the name of \*ALL. The \*ALL runmode specifies that all resources qualify regardless of their runtoken values just as though runmodes were not being used.

The slide also shows that three of the System Automation policy-defined applications have runtokens defined in the application information portion of their application policy. The first is the application called VTAM. Because VTAM is a basic system component and it needs to be qualified when the basic runmode is set, VTAM has a runtoken of B1 in its list of runtokens. This situation means that VTAM is available for runmode basic. However, VTAM also needs to be available for

the online applications that are defined in the System Automation policy. Therefore, VTAM also has one of the runtokens that qualifies for the onlines runmode. This situation means that VTAM qualifies when the runmode onlines is set. VTAM also has a runtoken listed that qualifies for the monitor runmode so VTAM can be made available for OMEGAMON when the monitor runmode is set.

The second and third applications shown on the slide are called CICSPROD and DB2PROD. They each have runtokens listed in their application information that qualify them for the onlines runmode. If the runmode is set to onlines, they are made available. However, because they do not have a runtoken listed that qualifies them for the basic and monitor runmodes, they are made unavailable if the basic or monitor runmode is set.

The example also shows two possible responses to the AOF603D initialization message. The first is for a runmode named onlines and the second is for a runmode named \*ALL. If the response to AOF603D is renominations, then, according to the runtokens specified for VTAM, CICSPROD, and DB2PROD, all three of these applications are made available. However, any applications without these runtokens are made unavailable. If the response to AOF603D is the special keyword \*ALL, then all resources qualify as though runmodes are not being used.

The diagram of concentric ovals shows that by using runmodes and runtokens you have a varying scope of control for starting and stopping resources. You can specify a runmode name of basic for the AOF603D response, represented by the inner most oval, and then after verifying that the basic components have all started correctly, you can change the runmode to onlines by using the INGRUN command in order to start the CICS and DB2 subsystems, represented by the middle oval. Then, at some point when you want all of your system automation applications available, you can set the runmode to \*ALL, represented by the outermost oval.

To shut down your system in stages, you can then revert back to runmode onlines. This action shuts down applications that are only in the outermost oval. Then, you can set the runmode to basic to stop CICS and DB2. Finally, you can stop the complete system.

# Defining runmodes

- Use ISPF customization panel Entry Type Selection option 99 User E-T pairs
- Define a new User E-T pair or select an existing entry

| AOFGENAM<br>Command ==>    |                           | Entry Name Selection   | Row 1 to 1 of 1<br>Scroll ==> CSR |
|----------------------------|---------------------------|--|-----------------------------------|
| Entry Type : User E-T Pair |                           | PolicyDB Name : OPSDEMOCOPY<br>Enterprise Name : OPSDEMOCOPY |                                   |
| Action<br><u>S</u>         | Entry Name<br>RUNMODEDEMO | Short Description<br>Demo runmodes                           |                                   |

↓

| AOFGEPOL<br>Command ==>                                |  | Policy Selection  | Row 1 to 5 of 5<br>Scroll ==> CSR |
|--|--|---|-----------------------------------|
| Entry Type : User E-T Pair<br>Entry Name : RUNMODEDEMO |  | PolicyDB Name : OPSDEMOCOPY<br>Enterprise Name : OPSDEMOCOPY            |                                   |
| Action<br><u>S</u>                                     | Policy Name<br>DESCRIPTION<br>E-T DATA | Policy Description<br>Enter description<br>Define user entry-type pairs |                                   |
|  | WHERE USED<br>COPY                     | List systems linked to this entry<br>Copy data from an existing entry   |                                   |

Select E-T DATA
 

© Copyright IBM Corporation 2019

6-15

## Defining runmodes

From the Entry Type Selection panel, select option 99 EUT to define runmodes. The names used for defining entries in the EUT policy are user-defined. After you define an entry, you can select the E-T DATA policy item to define a runmode.

## Defining runmodes (continued)

Select an existing User E-T Pairs entry or create a new one.

| AOFGUET  |        | UET Entry-Type Selection   |                               | Row 1 to 1 of 1<br>SCROLL==> CSR |
|----------|--------|----------------------------|-------------------------------|----------------------------------|
| Command  | ====>  | Entry Type : User E-T Pair | PolicyDB Name : OPSDEMOCOPY   |                                  |
|          |        | Entry Name : RUNMODEDEMO   | Enterprise Name : OPSDEMOCOPY |                                  |
| Action   | Entry  |                            | Type                          |                                  |
| <u>S</u> | INGRUN |                            | MODE                          |                                  |

| AOFGDYNNU  |         | UET Keyword-Data Specification |                               | Line 00000001 Col 001 075<br>Scroll ==> CSR |
|------------|---------|--------------------------------|-------------------------------|---|
| Command    | ====>   | Entry: INGRUN                  | Type: MODE                    |   |
|            |         | Mixed case . . . NO (YES NO)   | Keyword length. . . 20 (1-64) |   |
| Cmd        | Keyword | Data                           |                               |   |
| <u>NO</u>  | RVAPPL  | 'RVAPL0'                       |                               |   |
| <u>YES</u> | RVAPPL  | 'RVAPL1'                       |                               |   |

Runtokens

- The names **INGRUN** for entry name and **MODE** for type are required, reserved words
- There are two runmodes defined: RVAPL0 and RVAPL1

The User E-T Pairs entry name must be called INGRUN and the Type must be MODE. After you create an entry, you can define the runmodes you need. Keyword column on the UET Keyword-Data Specification panel is for the runmode name. The data column is for the list of runtokens associated with that runmode.

## Defining runmodes (continued)

### Select Where Used

| AOFGXWHU                   |                 | Where Used      |      | Row 1 to 2 of 2 |
|----------------------------|-----------------|-----------------|------|-----------------|
| Command ==>                |                 |                 |      | SCROLL==> CSR   |
| Entry Type                 | : User E-T Pair | PolicyDB Name   | :    | OPSDEMOCOPY     |
| Entry Name                 | : RUNMODEDEMO   | Enterprise Name | :    | OPSDEMOCOPY     |
| Action                     | Status          | Name            | Type |                 |
|                            | SELECTED        | MVSA            | SYS  |                 |
|                            | SELECTED        | MVSB            | SYS  |                 |
| ***** Bottom of data ***** |                 |                 |      |                 |

© Copyright IBM Corporation 2019

6-17

### *Defining runmodes (continued)*

You also specify the systems where the runmodes are used.

# Assigning an APL or MTR to a runmode

## Define runtokens in the Application Info or Monitor Info

Runtokens can be inherited from a class

| AOFGDYNA                         | Application Information       | Line 00000026<br>Scroll ==> CSR        |
|----------------------------------|-------------------------------|--|
| Command ==>                      |                               |  |
| Entry Type : Application         | PolicyDB Name : OPSDEMOCOPY   |  |
| Entry Name : RVCLASS             | Enterprise Name : OPSDEMOCOPY |  |
| External Shutdown . . . . .      | (FINAL ALWAYS NEVER)          |  |
| Shutdown Pass Interval . . . . . | 00:01:00<br>(hh:mm:ss)        |  |
| Cleanup Delay . . . . .          |                               | (hh:mm:ss)                             |
| Command Prefix . . . . .         |                               |  |
| Message Prefix . . . . .         |                               |  |
| Sysname . . . . .                |                               |  |
| Monitor Routine . . . . .        |                               | (name NONE)                            |
| Monitor Interval . . . . .       |                               | (hh:mm NONE)                           |
| Inform List . . . . .            | SDF                           | (SDF EIF E2E IOM ITM SMF TTT USR NONE) |
| ARM Element Name . . . . .       |                               |  |
| WLM Resource Name 1 . . . . .    |                               |  |
| WLM Resource Name 2 . . . . .    |                               |  |
| WLM Resource Name 3 . . . . .    |                               |  |
| Owner . . . . .                  |                               |  |
| Info Link . . . . .              |                               |  |
| Runtokens . . . . .              |                               |  |
| RVAPL0 RVAPL1                    |                               |  |
| ***** Bottom of data *****       |                               |  |

© Copyright IBM Corporation 2019

6-18

### Assigning an APL or MTR to a runmode

You can assign Applications and Monitor resources to one or more runmodes by specifying one or more runtokens in the information policy item. The specified runtokens must match the runtokens that are listed when the runmodes are defined.

# Use the INGRUN command in NetView to activate a runmode

Use the question mark (?) In the Runmode field to see a list of possible runmodes

```
INGKYRMO          SA z/OS - Command Dialogs
Domain Id . : AOFDA ----- INGRUN ----- Date . . . : 10/28/18
Operator Id : INGC102                                     Time . . . : 08:52:22

Request    => _____ Request type (SET, ADD or DEL)
Target      => _____ System name, domain ID or sysplex name

--- Parameters for SET request -----
System     => _____ System name
Runmode    => _____ Runmode name (mode or ?)
Persistent => _____ Keep request across IPL (YES/NO)
Type       => _____ Type of processing (NORM/IMMED/FORCE)
Priority   => _____ Priority of request (FORCE/HIGH/LOW)
Override   => _____ (ALL/NO/TRG/FLG/DPY/STS/SUS)
Verify     => _____ Check affected resources (YES/NO/WTOR)
Comment    => _____
```

---

```
AOKSEL5          SA z/OS - Command Dialogs           Line 1  of 3
Domain Id . : AOFDA ----- INGRUN ----- Date . . . : 03/22/19
Operator Id : INGC102                                     Time . . . : 17:06:34

Select one item, then press ENTER.

Sel. Item
-----
| *ALL
| NO_RVAPPL
| YES_RVAPPL
```

(CURRENTLY ACTIVE)

© Copyright IBM Corporation 2019

6-19

*INGRUN command*

After defining and assigning runmodes, you can rebuild and refresh the automation control file. The runmodes are then available for activation by using the INGRUN command. This slide shows the INGRUN command panel and the list of available runmodes by using a question mark (?) in the **Runmode** field.

# Student exercises

## Unit 6

### 1. Runmodes



© Copyright IBM Corporation 2019

*Student exercises*

Perform the exercises for this unit.

## Summary

---

Now that you have completed this unit, you should be able to perform the following tasks:

- Describe the automation environment
- Describe automation manager initialization
- Describe automation agent initialization
- List and explain the manager and agent start options
- Define and implement runmodes

---

© Copyright IBM Corporation 2019

### *Summary*

Now that you have completed this unit, you should be able to perform the following tasks:

- Describe the automation environment
- Describe automation manager initialization
- Describe automation agent initialization
- List and explain the manager and agent start options
- Define and implement runmodes





## 7 Gateways and Status Display Facility



### Unit 7 Gateways and Status Display Facility



© Copyright IBM Corporation 2019  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

**What this unit is about**

In this unit, you learn how to define Gateway sessions and autotasks and how to use and customize the Status Display Facility (SDF) component of SA z/OS. SDF displays the status of your SA z/OS resources from 3270 panels. You can customize the SDF panels to aggregate status or to group resources based on business function, for example.

**How you check your progress**

You can check your progress in the lab exercises.

**References**

*SC34-2717 System Automation for z/OS Version 4*

*Release 1 Defining Automation Policy*

[https://www.ibm.com/support/knowledgecenter/SSWRCJ\\_4.1.0/com.ibm.safos.doc\\_4.1/manuals.html](https://www.ibm.com/support/knowledgecenter/SSWRCJ_4.1.0/com.ibm.safos.doc_4.1/manuals.html)

# Objectives

After completing this unit, you should be able to perform the following tasks:

- Define Gateway sessions and autotasks
- Describe how the Status Display Facility (SDF) collects status information
- Define SDF tree structures
- Create SDF panel definitions
- Customize SDF definitions in DSIPARM

## Objectives

After completing this unit, you should be able to perform the following tasks:

- Define Gateway sessions and autotasks
- Describe how the Status Display Facility (SDF) collects status information
- Define SDF tree structures
- Create SDF panel definitions
- Customize SDF definitions in DSIPARM

# Lesson 1. Gateways

## Lesson 1: Gateways

- An overview of gateways
- Definition of gateways
- Definition of GATEWAY autotasks

© Copyright IBM Corporation 2019

7-3

### What this lesson is about

This lesson provides an overview of gateways and how to implement them.

### Objectives

After completing this lesson, you should be able to describe gateways and to define gateways and GATEWAY autotasks.

### References

*SC34-2716 System Automation for z/OS Version 4 Release 1 Planning and Installation*

*SC34-2717 System Automation for z/OS Version 4 Release 1 Defining Automation Policy*

## An overview of gateways

- Definition of a gateway

- A link between two Tivoli NetView for z/OS domains that run System Automation for z/OS automation agents
- NetView RMTCMD or XCF inside SAplex
- Communication occurs through *inbound* and *outbound* sessions
- Default taskname: GAT&domain.
- Sessions start when System Automation for z/OS initializes
- Are typically kept active. The time interval is set in the Gateway Monitor Time field in the SYSTEM INFO policy (AOFGATE timer)

- Purpose of a gateway

- Base for Status Display Facility focal point operation
- Base for non-sysplex SPOC
  - Used to determine active systems
  - Enables commands and messages to flow

### An overview of gateways

A *gateway link*, more commonly called a *gateway*, is a link between two NetView domains. The link consists of an inbound session and an outbound session between the two domains. A gateway link enables commands, status, and messages to flow between the domains. Such a link forms the basis for the Status Display Facility (SDF) *focal point*. Gateways can be set up between domains even when neither domain is a focal point.

Automation uses z/OS cross-system coupling facility (XCF) for communication between agents within the same sysplex. Gateway sessions between agents in different sysplexes use the NetView RMTCMD function. RMTCMD is a NetView function that uses TCP/IP or VTAM/SNA for communication between domains. Examples are shown later in this unit.

Default taskname is GAT&domain. Sessions start when System Automation for z/OS initializes and are typically kept active. The time interval is set in the Gateway Monitor Time field in the SYSTEM INFO policy resulting in an AOFGATE timer).

Purpose of a gateway:

- Base for Status Display Facility focal point operation
- Base for non-sysplex SPOC
- Used to determine active systems
- Enables commands and messages to flow

## Definition of gateways

- **Explicit** gateway definitions (entry type NTW in the PDB) are not necessary for systems within the **local SAplex** and **extended SAplex**
- Nevertheless **SDF** relies on certain data in the SA internal data model to
  - Ensure that SDF status forwarding to FP works also in the ‘extended’ SAplex
  - Ensure that all data is deleted on the SDF FP in case of system shutdown or connection loss
  - Ensure that all connections can be seen in DISPGW with the correct INBOUND and OUTBOUND status
- Therefore gateway definitions are created automatically (if not already defined in the PDB) during **configuration build**
  - On each **target system**: for the SDF FP and backup FP system (policy SDF FOCALPT in entry type NTW)
  - On the **primary and backup FP system**: for all target systems
    - To make this possible the SA NetView domain ID needs to be specified on SYSTEM INFO policy of the SDF focal point

### Definition of gateways

Explicit gateway definitions (entry type NTW in the PDB) are not necessary for systems within the local SAplex and extended SAplex. Nevertheless SDF relies on certain data in the SA internal data model to:

- Ensure that SDF status forwarding to FP works also in the ‘extended’ SAplex
- Ensure that all data is deleted on the SDF FP in case of system shutdown or connection loss
- Ensure that all connections can be seen in DISPGW with the correct INBOUND and OUTBOUND status

Therefore gateway definitions are created automatically (if not already defined in the PDB) during configuration build:

- On each target system: for the SDF FP and backup FP system (policy SDF FOCALPT in entry type NTW)
- On the primary and backup FP system: for all target systems

To make this possible the SA NetView domain ID needs to be specified on SYSTEM INFO policy.

# GATEWAY Policy item

- GATEWAY policy item is part of the Network Policy (NTW)
- Network Policy Selection panel

| AOFGEPOL  | Policy Selection                                    | Row 2 to 10 of 10<br>Scroll ==> CSR        |
|---|---|--|
| Command ==>                                       |   |  |
| Entry Type : Network<br>Entry Name : BASE_NETWORK | PolicyDB Name : NEW_PDB<br>Enterprise Name : SAPLEX |  |
| Action  | Policy Name   | Policy Description                         |
|   | SDF FOCALPOINT                                      | Define Status Display Facility focal point |
|   | FULL SESSIONS                                       | Define TAF sessions (Applications)         |
|   | GATEWAY   | Define gateways                            |

- GATEWAY Definitions Panel

| AOPFINE4   | GATEWAY Definitions | Row 1 to 23 of 23<br>SCROLL==> CSR |
|--|---------------------|------------------------------------|
| Command ==>  |                     |                                    |
| Specify each NetView domain that commands or responses are forwarded to. |                     |                                    |
| Domain   | Description         |                                    |
| ING01  | Gateway to SYS1     |                                    |
| ING02  | Gateway to SYS2     |                                    |
| ING03  | Gateway to SYS3     |                                    |

© Copyright IBM Corporation 2019

7-6

*GATEWAY Policy item*

Explicit gateway definitions (entry type NTW in the PDB) are not necessary for systems within the local SAPlex and extended SAPlex.

The GATEWAY policy item is part of the Network Policy (NTW).

The screenshots show:

- Network Policy Selection panel
- GATEWAY Definitions Panel

# GATEWAY autotasks

- Define all gateway autotasks in NetView DSIPARM data set member DSIOPF
- Define the outbound gateway autotask (GATOPER) in the Automation Operators policy object

```
AOFGDYNO      Automation Operator Definitions      Line 00000001 Col 001 075
Command ===> _____ Scroll ===> CSR

Entry Type : Automation Operators  PolicyDB Name   : NEW_PDB
Entry Name  : GATEWAY_AUTOOPS    Enterprise Name : SAPLEX

Line Commands: S (Select), M (Move), B (Before), A (After), R (Repeat)
               I (Insert lines), D (Delete lines)

Cmd AutoFunc Primary Operator  Backup Operator  Messages for this Operator
____ GATOPER_ GAT&DOMAIN. _____
```

## Gateway autotasks

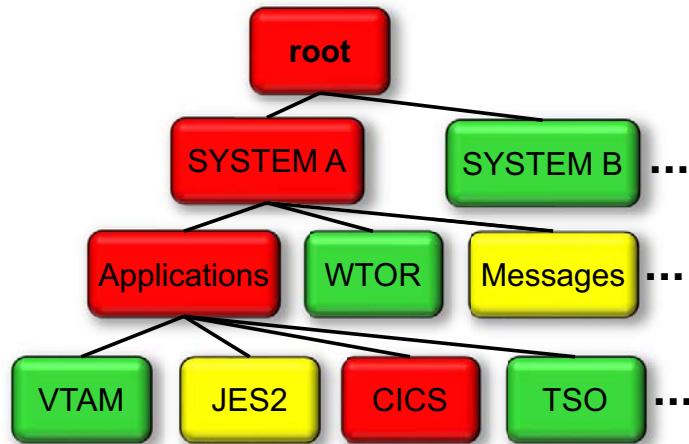
The following customization is required:

- Define all gateway autotasks in NetView DSIPARM data set member DSIOPF
- Define the outbound gateway autotask in the GATOPER policy for the Auto Operators policy object

# Lesson 2. Status Display Facility overview

## Lesson 2: Status Display Facility overview

- Introduction to SDF
- SDF displays
- SDF features



### What this lesson is about

This lesson provides an overview of the Status Display Facility.

### Objectives

After completing this lesson, you should be able to describe the purpose of the Status Display Facility.

### References

*SC34-2647 System Automation for z/OS Version 4 Release 1 User's Guide*

# Introduction to SDF

SDF is a presentation display within agent NetView

- Operates independently of Automation Agent and other SA z/OS monitoring
- Maintains and displays agent status for these items:
  - Applications and Application Groups
  - Configuration refreshes
  - Gateway sessions
  - Monitor resources
  - Spool
  - WTORs
  - Captured and exceptional messages
  - Batch job and IBM Workload Scheduler related events
  - Tape
  - TSO users
  - Processor Operations LPAR resources and Processor resources
- Uses a tree structure so that status is aggregated
- Provides dynamic display of resources and their status
- Inform List policy must include SDF to enable status updates

## Introduction to SDF

The Status Display Facility (SDF) is a standard function of SA z/OS that you use to manage status information and present it to NetView operators. Internally, it is a separate task in NetView. SDF supports the presentation of most types of status. SDF displays the Agent Status of resources, such as Applications. The SDF status is not the same as the current health or even the Desired Status of a resource.



**Hint:** You can disable SDF by coding the following statement in the NetView style sheet:

`TASK.AOFTDDF.MOD=*NONE*`

The SDF process operates solely within the SA z/OS automation agent, although some of the information that is presented is known only to the automation manager. For example, status on Application Groups is from the automation manager. SA z/OS automation routines report status changes to SDF for the resources controlled by SA z/OS. SDF is not part of any automation activities.

# SDF displays

- 3270 screen, full attributes
  - Default screen size is 24x80. Higher sizes supported too
- Three main types of panels as follows:
  1. Root panel showing a system or enterprise  
First panel shown, links to the status component panels
  2. Panels for *status components* below the *root component*
  3. Detailed status display built from a *Status Descriptor*
- Multiple simultaneous users
- Program function (PF) key capabilities
- Root and Status component panels are technically the same:
  - Dynamically updated
  - Can be customized by users
  - Aggregation and hierarchy capabilities

## SDF displays

An SDF display is a 3270 full screen display. It shows status values by using full color and other 3270 attributes, such as reverse video, underlining, and blinking. In each installation, you can customize SDF panels.

Default screen size is 24x80. Higher sizes like 62x160 are supported too using the parameter SCREENSZ

- Root component

The root component is typically an element on the first screen, showing a system or enterprise, that is displayed when SDF is started. There is only one root component.

- Status component

Resources that are monitored by SDF are called *status components*. The status component panel displays all monitored resources in a system. Each monitored resource is shown in the color of its current status. For example, JES2 is shown in green if it is running.

- Detail Status Display

A detail status display is built from information in a status descriptor. This panel is displayed when you press the Tab key to the appropriate resource on the status component panel and

pressing the detail PF key. Each status component can have one or more status descriptors, or detail records, associated with it.

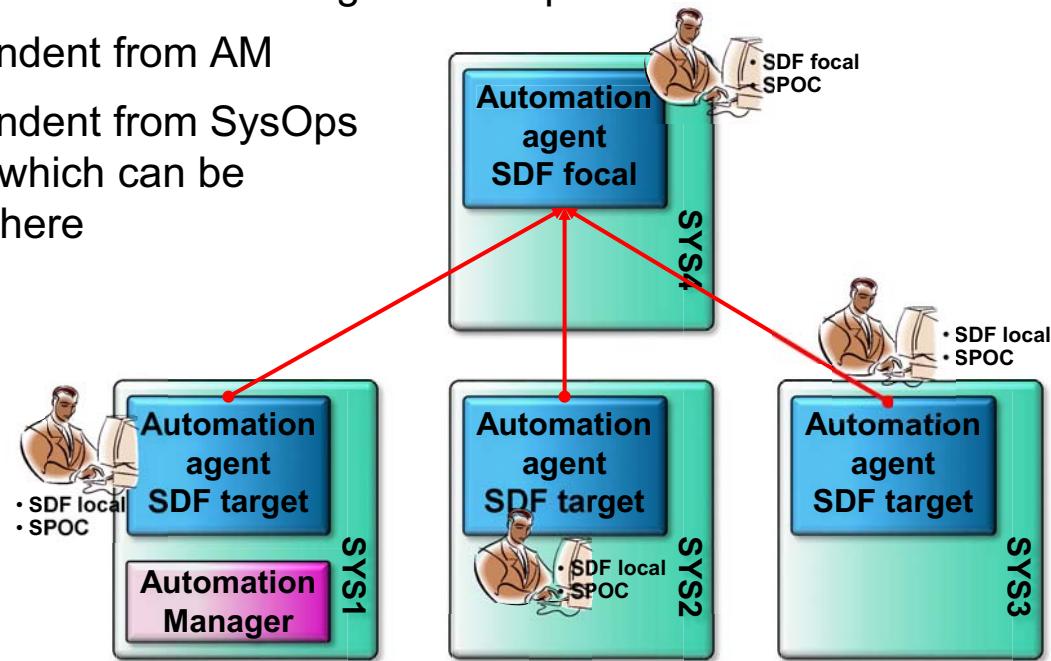
You can navigate between the panels by pressing PF8 to drill down for further details, for example. Several NetView users can view SDF displays simultaneously. These users can view the display of their choice. Within any display, users can press PF keys to invoke commands.

You can aggregate the status of resources so that summary displays can be created. Use PF keys to drill down from an aggregate resource to a more detailed view of its constituents, and to return to the aggregate view.

A status descriptor is a detailed record of information about a resource status. In its raw form, a status descriptor is a multi-line SA z/OS message containing information such as the Root component and status component to which the status descriptor applies. SDF uses information in a status descriptor to generate a detail status display.

## Status Display Facility focal point and Single Point of Control (SPOC)

- SDF is an agent function and needs to be implemented on all systems with one being the focal point
- Independent from AM
- Independent from SysOps SPOC which can be everywhere



© Copyright IBM Corporation 2019

7-11

### Status Display Facility focal point

This slide shows a simple System Automation for z/OS implementation across four systems. SYS4 is the *SDF focal point*. SYS1, SYS2, and SYS3 are the *SDF target systems*.

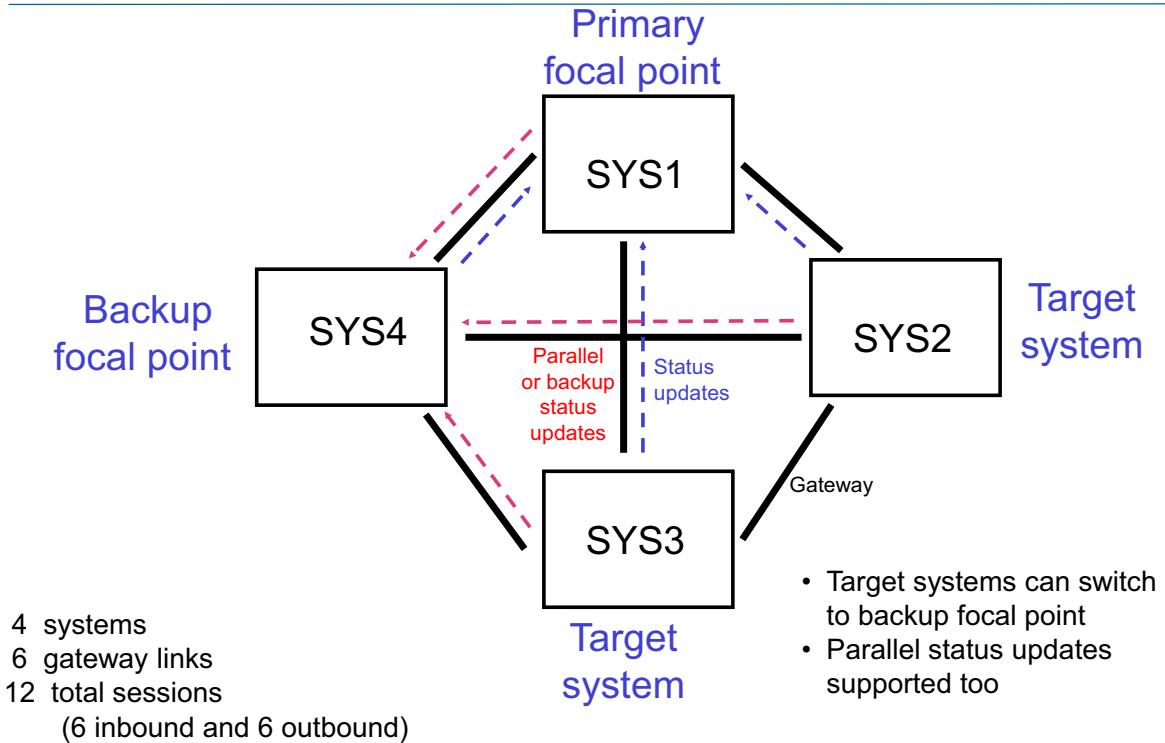
Each system is running a automation agent and Status Display Facility (SDF). If an agent in a target system detects a problem, SDF is updated in the agent's SDF, and the status is propagated over the outbound gateway session to the SDF focal point.

The SDF focal point in SYS4 is customized to display the status of resources across all four systems on an Enterprise Status panel. An operator in SYS4 can then use SDF as a status focal point.

An operator on any system can also access SysOps automation functions to control resources across all four systems if they are in the same SAplex or have gateway sessions to each other like shown in an example later in this unit which shows shows four domains that are interconnected. This type of management is called *Single Point of Control (SPOC)*.

In this example, the SDF focal point is in SYS4. The location of the primary automation manager is independent from the SDF focal point. The primary automation manager and the the SDF focal point can be on different systems. For example the primary automation manager can run in SYS1.

## Gateways, focal point, and SPOC



© Copyright IBM Corporation 2019

7-12

### Gateways, focal point, and SPOC

This slide illustrates three concepts:

- SDF focal points
- SPOC
- Gateway links between the domains to support the focal point and SPOC

In this example, gateway links, represented by the solid lines, connect all four domains. There are 6 gateway links, 12 sessions. This type of configuration enables any domain to be used as the single point of control. SYS1 is the primary focal point and SYS4 is the backup focal point.

The dashed lines represent status, which flows to the focal point from domains, SYS4, SYS2, and SYS3. If all of these systems are in the same sysplex, the gateway sessions use XCF for communication.

The target systems can switch to backup focal point SYS4. Parallel status updates are supported too.

## Customized root panel for an enterprise (SINGNPRM SDF sample INGPTOP)

MVSA

SA z/OS 4.1 Education

05/22/19 19:54:25

Environment >Configuration Refresh

SAplex  
sysplex  
VM guest      SYSPLEX1  
PLEX12  
mvsch0d      >MVSA      >MVS

STANDALONE SYSTEMS >MVSC

Hardware >Processors

>Ensembles

====> \_\_\_\_\_  
1=Help 2=Detail 3=Return      6=Roll 8=Zoom

© Copyright IBM Corporation 2019

7-13

Customized root component (SINGNPRM SDF sample INGPTOP)

The "SDF" command will initially show the panel defined in member *INGPTOP*, and from there customers can use the standard well known SDF navigation techniques and PF Keys to view data for specific systems and underlying resources.

This slide shows the customized root panel *INGPTOP* that is provided as a sample in the SINGNPRM data set. The systems have been adapted and the panel title has been changed. This is an enterprise-wide panel that can be used as the top panel defined for an SDF focal point. Also system symbols are used to display the sysplex name and the z/VM guest the system runs in.

The panel was designed to show the SA for z/OS controlled systems MVSA and MVS in SAplex SYSPLEX1. System MVSC is displayed in the standalone systems area.

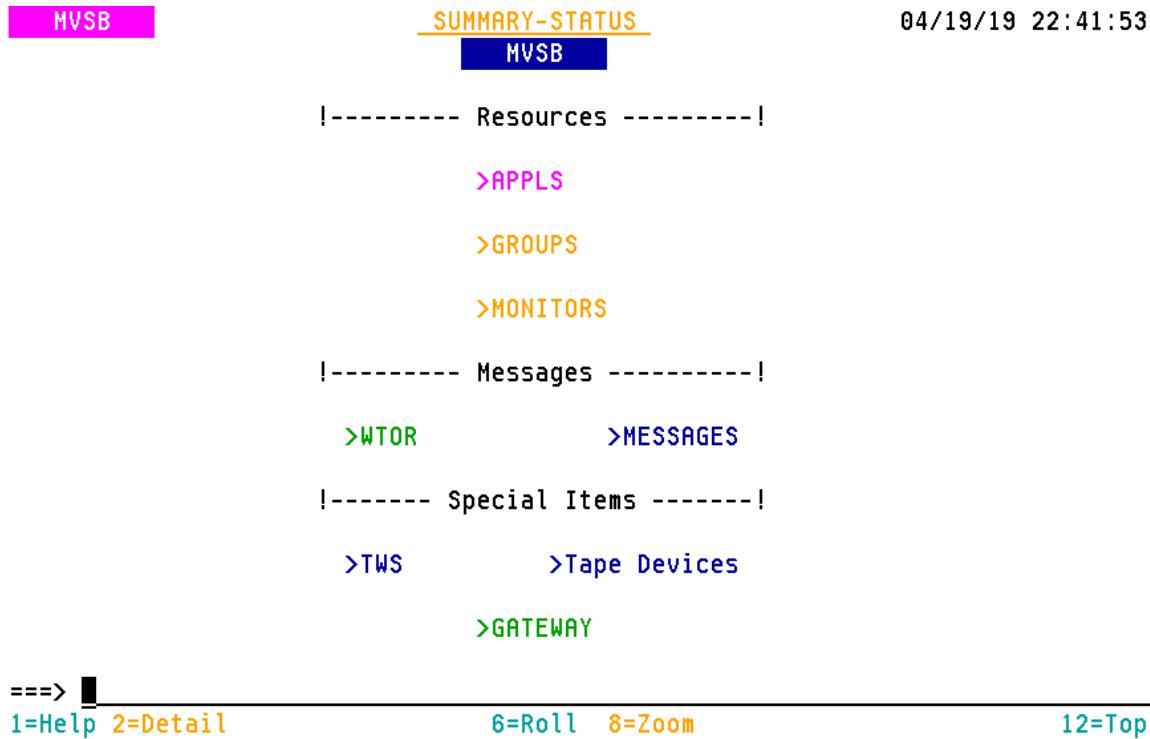
Configuration Refresh Monitoring displays the status of the configuration refreshes.

You place the cursor on a system name and use the F8 key to go to the Summary status for a system panel.

This panel uses SDF sample INGPTOP provided in data set SINGNPRM.

The hardware area has not been removed but is not implemented here.

## Root panel for a system SDF sample INGPMAIN. Top level on SDF targets



© Copyright IBM Corporation 2019

7-14

### Summary status for a system

This slide shows the Summary status for system MVSB using SDF sample INGPMAIN. This panel is the top level panel on a SDF target system.

You place the cursor on a Status item and use the F8 key to go to the Status component panel.

This panel uses SDF sample INGPMAIN provided in data set SINGNPRM.

# Subsystem (APL) status for system MVS B

## SDF sample INGPAPL

MVS B

MVS B: SUBSYSTEM-STATUS

04/19/19 22:42:38

1/32(32)

|          |                 |
|----------|-----------------|
| RV04     | SDSF            |
| RV06     | TCP/IP          |
| APPC     | TELNET          |
| ASCH     | TN_PORT         |
| AUTONETV | TSO             |
| AUTOSSI  | VLF             |
| DLF      | VTAM            |
| FTP      | ZFS             |
| FTP_PORT | RV02            |
| JES2     | <u>BLSJPRMI</u> |
| LLA      | <u>IRRDPTAB</u> |
| OMVS     | <u>SYSVIPLC</u> |
| RACF     | <u>AUTOMGR2</u> |
| RESOLVER | <u>HZSPROC</u>  |
| RMF      | RV05            |
| RMFGAT   | AUTOMGR         |

==> █

1=Help 2=Detail  
13=EXPLAIN

6=Roll

9=Bottom 10=Previous 11=Next 12=Top  
17=SETSTATE 18=INGVOTE 19=INGREQ 23=INGLIST 24=INGINFO

© Copyright IBM Corporation 2019

7-15

### Subsystem status

The Subsystem status shows the subsystem (APL resources) status for the selected system and their status represented by their color. The subsystems are sorted by priority. To select a subsystem from the status component panel, you place the cursor on the subsystem and press F2. The detailed status component panel is then displayed.

This panel uses SDF sample INGPAPL provided in data set SINGNPRM.

## Application group (APG) status for system MVSB SDF sample INGPAPG

| MVSB   | MVSB: APPLGROUP-STATUS | 04/19/19 22:44:11 |
|--|------------------------|-------------------|
|  |                        | 1/7(7)            |
| <b>HZSPROC</b><br><b>RVXSERVER</b><br><b>BASE_SYS</b><br><b>FTP_DAEMON</b><br><b>TELNET_3270</b><br><b>AM_X</b><br><b>RVXMOVE</b>  |                        |                   |
| ====> <a href="#">1=Help</a> <a href="#">2=Detail</a> <a href="#">6=Roll</a> <a href="#">9=Bottom</a> <a href="#">10=Previous</a> <a href="#">11=Next</a> <a href="#">12=Top</a><br><a href="#">18=INGVOTE</a> <a href="#">19=INGREQ</a> <a href="#">22=INGGROUP</a> <a href="#">23=INGLIST</a> <a href="#">24=INGINFO</a> |                        |                   |
| © Copyright IBM Corporation 2019   |                        | 7-16              |

### Application group status

The application group status shows the application group (APG resources) status for the selected system and their status represented by their color. The application groups are sorted by priority. To select an application group from the application group panel, you place the cursor on the application group and press F2. The detailed status component panel is then displayed.

This panel uses SDF sample INGPAPG provided in data set SINGNPRM.

# Monitor (MTR) status for system MVSB

## SDF sample INGPMTR

MVSB

MVSB: MONITOR-STATUS

04/19/19 22:44:59

1/3 (3)

MTRNETV  
JES2MON  
JES2SPOOL

==>   
1=Help 2=Detail

6=Roll 9=Bottom 10=Previous 11=Next 12=Top  
18=INGVOTE 19=INGREQ 22=DISPMTR 23=INGLIST 24=INGINFO

© Copyright IBM Corporation 2019

7-17

### Monitor status

The Monitor status shows the Monitor (MTR resources) status for the selected system and their status represented by their color. The Monitor resources are sorted by priority. To select an Monitor resource from the Monitor status panel, you place the cursor on the Monitor resource and press F2. The detailed status component panel is then displayed. See next page.

This panel uses SDF sample INGPMTR provided in data set SINGNPRM.

## Detailed status component for a Monitor resource

```
1 of 3          ---- Detail Status Display ----      10/25/18 11:26:31
Component . . . : MTRNETV           System . . . . . : MVSA
Color . . . . . : YELLOW            Priority . . . . . : 450
Date . . . . . : 10/25/18           Time . . . . . . : 11:21:50
Reporter . . . . : AUTWRK03        Node . . . . . . : AOFDA

Info . . . . . : MTRNETV
Reference value : MTRNETV
User data . . . . : USER Data  
240 bytes

A0F550I STATUS OF MTRNETV MONITOR : OBSERVED=AVAILABLE HEALTH=WARNING -
MTRNETV: NETVIEW TOTL 254 62.66 100.00 0.00 0      35331      N/A

====>
1=HELP 3=RETURN 4=DELETE 6=ROLL 7=UP 8=DOWN 11=BOTTOM 12=TOP
```

© Copyright IBM Corporation 2019

7-18

### Detailed status component

The detailed status component shows the details associated with a resource, which in this case is the MTRNETV monitor resource.

The detailed status component panel can not be customized.

### User data

The AOCUPDT command was updated to include the USER parameter, which specifies the user data that is associated with the resource.

The data is stored in a 240-byte area initialized with blanks. The parameter accepts two values:

1. The offset which defaults to 1 specifies the location in the area where the subsequent data is stored.
2. The user data enclosed by delimiters. The delimiter can be any printable character greater than x'40' that does not occur in the data.

SDF truncates the data when the offset plus the length of the data exceed the maximum size.

## Gateway status for system MVSB to system MVSA SDF sample INGPGTW

MVSB

MVSA: GATEWAY-STATUS

04/19/19 22:47:59

| Item   | Message text  | 1/2(2) |
|--------|---|--------|
| AOFDBI | AOF569I 22:12:39 : STATUS OF AOFDA INBOUND GATEWAY FROM DOMAIN AOFDB IS |        |
| AOFDBO | AOF568I 22:12:39 : STATUS OF AOFDA OUTBOUND GATEWAY TO DOMAIN AOFDB IS  |        |

==> █

1=Help 2=Detail

6=Roll

9=Bottom 10=Previous 11=Next 12=Top

24=DISPGW

© Copyright IBM Corporation 2019

7-19

### Gateway status

The Gateway status shows the Gateway status for the selected system and their status represented by their color as well as the latest, related message. The Monitor resources are sorted by priority. To select a Gateway status from the Gateway status panel, you place the cursor on the Gateway line and press F2. The detailed status component panel is then displayed.

This panel uses SDF sample INGPGTW provided in data set SINGNPRM.

## WTOR status for system MVSA on system MVSB

| MVSB     | MVSA: WTOR-STATUS  | 04/19/19 22:46:55 |
|----------|--|-------------------|
| Jobname  | Reply ID / Message text  | 1/3(3)            |
| RV01     | 0004 VAPL21000A REPLY WARM OR COLD                               |                   |
| RV02     | 0005 VAPL22000A REPLY WARM OR COLD                               |                   |
| AUTONETV | 0006 DS1802A AOFDA REPLY WITH VALID NCCF SYSTEM OPERATOR COMMAND |                   |

====> █

1=Help 2=Detail      6=Roll      9=Bottom 10=Previous 11=Next 12=Top

© Copyright IBM Corporation 2019

7-20

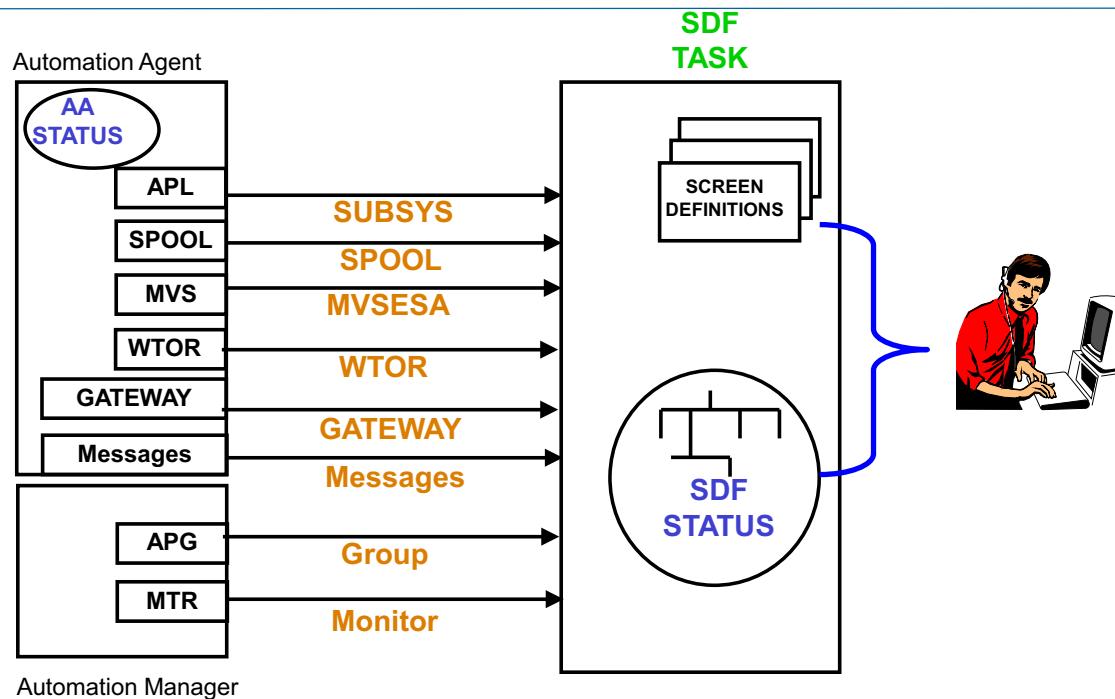
### WTOR status

The WTOR status shows the WTORS for the selected system and their priority represented by their color as well as the reply ID and message. The WTORS are sorted by priority. To select a WTOR from the WTOR status panel, you place the cursor on the WTOR line and press F2. The detailed status component panel is then displayed.

This panel uses SDF sample INGPWTOR provided in data set SINGNPRM.

# Lesson 3. How SDF works

## Lesson 3: How SDF works



Status updates require an Inform List policy that includes a value of **SDF**

### What this lesson is about

This lesson describes how SDF works in the run-time environment.

### Objectives

After completing this lesson, you should be able to describe how SDF works in the runtime environment.

### References

*SC34-2647 System Automation for z/OS Version 4 Release 1 User's Guide*

In this figure, you see a simplified explanation of how SDF works. During SA z/OS agent automation routines, you can see the status of many resources. At important points, the routines issue calls to SDF to report the updated status.

Statuses are reported for these items:

- The Agent status of each subsystem that is managed by the automation agent in that NetView.
- WTORs: Each WTOR is represented as a resource if it is *outstanding*.
- Gateways: Each session that forms part of a Gateway link.
- Application Groups
- Monitor resources
- MVS automation components, which also report status when they pass thresholds.
- Spool utilization, which is represented as a resource.
- Product-specific automation (such as Tivoli Workload Scheduler)

SDF receives these status updates and maintains a pool of resources and their status (known as *status descriptors*). The installation can define SDF screens (or panels) to show any selection of these status descriptors. The installation might also define a hierarchy, or tree structure, of how these status descriptors are collected into aggregate resources. These aggregate resources can also be used as part of the description of the SDF panels. The information flow is all one way, from the automation agent into the status descriptor records.

Several users can view panels simultaneously. Whenever a status changes in a panel that a user is viewing, the panel is updated automatically and instantly to reflect that status change for all users.

The SDF status is based on information received; not as an interface to the actual Agent Status. The use of SDF is optional and does not impact automation. SDF can be stopped and started without interfering with automation.

The only requirement for the interface is that the tree structure must have definitions for the fixed status types that are reported to it. You can also add your own reporting to what is performed by SA z/OS, and you can link it to the hierarchy definition and to the panel definitions.



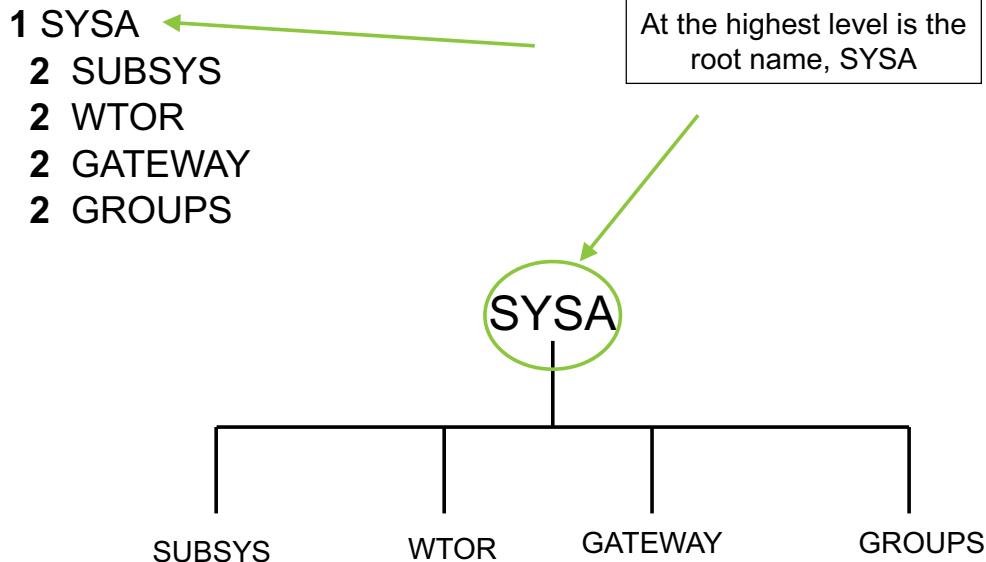
**Hint:** You can use the RESYNC command to resynchronize the SDF status information. Use RESYNC SDF on the local domain and RESYNC FP on the Focal Point domain. Use RESYNC SDFDEFS to refresh (reload) the SDF panel definitions.

## SDF tree structure

- SDF uses a tree structure to manage the status of resources
  - Typically, there is one tree per target system
  - Focal point system SDF requires ALL trees
  - The default setting for the root of the tree is the MVS system name
- Each status update is assigned to lowest part of the tree structure
- The user can customize the structure; for example, define levels of aggregation

The concept of a tree structure is fundamental to SDF operation. As each update is received by SDF, the update is assigned to some part of a tree structure, which is defined by the installation. If it cannot assign the update to a tree structure, the update is discarded and cannot be used in any status display. Each SA z/OS domain that is reporting typically has one tree structure, although additional trees can be defined at installation.

## SDF tree structure example



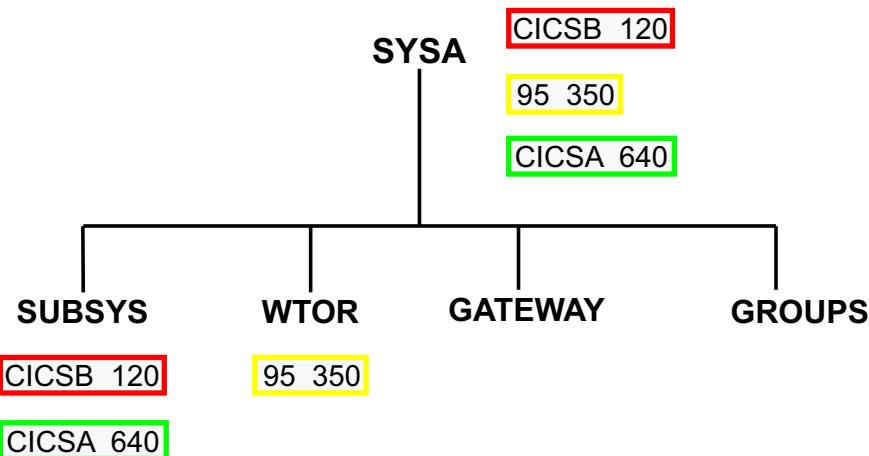
© Copyright IBM Corporation 2019

7-23

### SDF tree structure example

This slide shows an example of a simple SDF tree definition. Each tree structure has a root (shown at the top, level 1). For the SA z/OS standard statuses, the root name is typically a name that matches the system or domain, the SDF root name. All statuses updated by SA z/OS use the same root name. In this example, it is SYSA. Each furthest part of the tree structure, or *leaf*, is one of the standard status types. In this definition, there are no intermediate aggregate values. The only aggregate is the root name itself. The definition on the left side of the slide is the way that the types are defined to SDF. The resource types are defined at level 2.

# Status management in SDF



**SA z/OS:** WTOR 95 DSI802A

Animated

© Copyright IBM Corporation 2019

7-24

## Status management in SDF

This example illustrates what happens in SDF when status updates are received, based on the example from the previous slide. Initially, no statuses, or *status descriptor records*, are available. Then, an update is received from the automation agent that indicates that the subsystem status for CICSA is UP.

Status within SDF is maintained as a priority value. UP as a status is not recognized in SDF because it is independent of the automation agent and other SA z/OS components. SA z/OS reports it as a status update of priority 640. This update is assigned to the SUBSYS leaf. SUBSYS has one status descriptor record. Because of the hierarchy structure, the aggregate level SYSA also has one status descriptor record.

The status descriptor record contains information (fields) about the update. For example, the text field has a CICSA IS UP message, and the origin field is the domain where the record originated. (The syntax is simplified in this slide.) Only in the DATA field is there any reference to the UP status.

If another update is received, this time to indicate that CICSB is BROKEN, the automation agent translates BROKEN to a priority of 120 and issues an SDF update. SDF adds this update to the SUBSYS leaf. For status descriptor records at the same level, SDF maintains the records in priority

sequence. The CICSB record is first, followed by the CICSA record. Low numbers are highest priority. The status descriptor records collected at the SYSA aggregate level are also sorted that way.

If a WTOR is detected, SA z/OS issues a status update for the WTOR and assigns a priority of 350. It is assigned to the WTOR leaf, as shown on the slide, and is also collected at the SYSA level. Because of the sorting, it becomes the second status descriptor record at the SYSA level because of its priority.

SDF reporting continues in this way. There is also a DELETE process. For example, when the WTOR is responded to, a delete request is sent to SDF and the status descriptor record is deleted. If the status for CICSB is updated to CTLDOWN, the automation agent issues a request to update the status. As part of most update processes, existing records for the same resource are deleted.

# Lesson 4. Customizing SDF panels

## Lesson 4: Customizing SDF panels

Overview of defining and customizing SDF panels

- Defining panels (PANEL)
- Status component definition statements
  - STATUSFIELD: location on panel
  - STATUSTEXT: text displayed
- Text fields and data definition statements
  - TEXTFIELD: location on panel and attributes
  - TEXTTEXT: text displayed
- PF key definitions (PFKnn)
- End panel statement (ENDPANEL)

### What this lesson is about

This lesson describes how to customize SDF panels.

### Objectives

After completing this lesson, you should be able to customize SDF panels.

### References

*SC34-2560 System Automation for z/OS Version 4 Release 1 Programmers Reference*

## SDF features

- The dynamic panel generation function generates panels for all specified systems
- Panels can be cashed and can be refreshed
- Definition members support %INCLUDE statements
  - Requires parentheses around the specified member or members
- With exit AOFEXX05, the installation can replace user variables that are defined in the SDF tree and panel definitions
  - Based on the system for which the tree and panels are generated
- The Status Display Facility has its own AOFS messages
  - You can refer to the *IBM System Automation for z/OS Version 4 Release 1 Messages and Codes* manual for the message numbers
  - Message help is also available from NetView by entering HM AOFSxxx

### SDF features

- The Status Display Facility dynamic generation function generates panels for all specified systems.
- Panels can be cashed and can be refreshed.
- The exit AOFEXX05 runs at initialization when setting up the SDF panels or by using the RESYNC SDFDEFS command. The exit is used by the installation to replace user variables in the SDF panel definition. A user variable must follow the same convention as a z/OS system symbol; it must start with an ampersand (&) and finish with a dot(.). &MVDOMAIN. is an example. Refer to the sample exit for details of the parameters that are passed to the exit and the return codes.
- The Status Display Facility now has its own message set using AOFS prefixes.

## SDF panel definition statements

| Statement               | Description   |
|-------------------------|---|
| <b>PANEL (P)</b>        | Defines the start of a panel definition and basic navigation                |
| <b>STATUSFIELD (SF)</b> | Defines a panel field whose color is determined by the status of a resource |
| <b>STATUSTEXT (ST)</b>  | Text inserted into the STATUSFIELD  |
| <b>TEXTFIELD (TF)</b>   | Defines a panel field containing text                                       |
| <b>TEXTTEXT (TT)</b>    | Text inserted into the TEXTFIELD  |
| <b>PFKnn</b>            | Action taken when PF key is pressed   |
| <b>ENDPANEL (EP)</b>    | End of a panel definition   |

You use most, if not all, of these statements to define an SDF panel

### *SDF panel definition statements*

Defining a 3270 panel that a user can view involves defining which status descriptors are displayed on a panel and their locations. You create a definition of the panel by specifying the different fields and status descriptors on the panel.

This table shows several definition statements that you can use to create a panel. The next slide show several additional statements. These two slides are not intended to be a complete list. Several of the more important statements are explained through examples.

## SDF panel definition statements (continued)

| Statement         | Description   |
|-------------------|---|
| <b>BODY</b>       | Defines the section in the panel that can be used by SDF to display the various status components listed in the order of their priority |
| <b>BODYHEADER</b> | Defines a descriptive header and the scrolling information of the body section  |
| <b>BODYTEXT</b>   | Defines the text displayed in the panel header line   |
| <b>CELL</b>       | Defines the various information units to be displayed for a status component and their placement in the body section                    |
| <b>INPUTFIELD</b> | Defines the location of the input field   |
| <b>DATETIME</b>   | Defines the location and attributes of the current date and time  |

### *SDF panel definition statements (continued)*

The *BODY* statement is used to define the section in the panel that can be used by SDF to display the various status components listed in the order of their priority, as well as the layout of the table. Up to three sort fields can be specified. Each sort field consists of the type of information followed by the sort order.

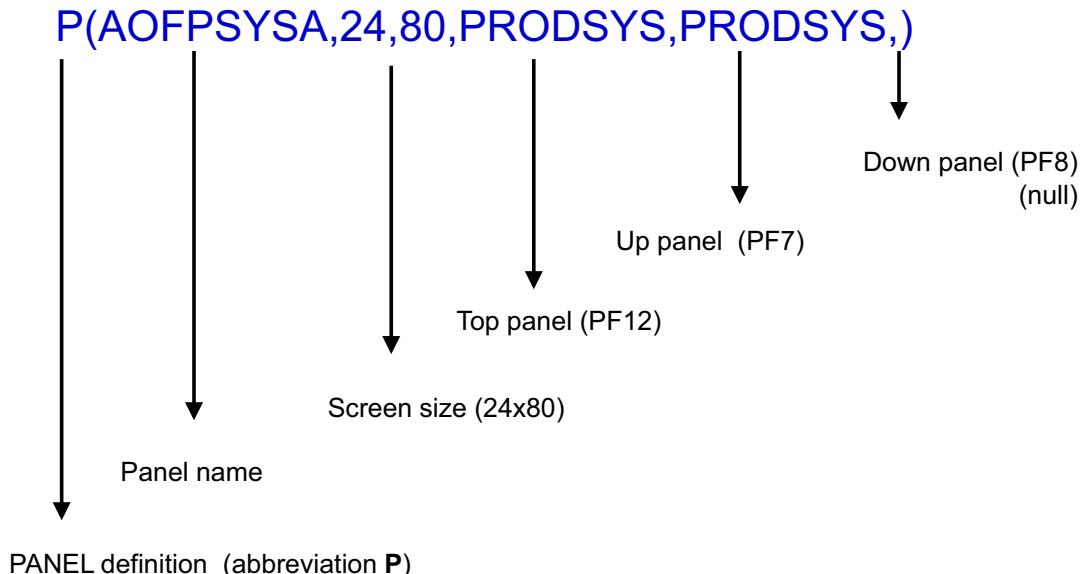
New is that you can specify the column number where the body section begins and ends. Default is the entire screen.

The *BODYHEADER* statement is used to define a descriptive header and the scrolling information of the body section. The length of the header is implicitly defined by the body width.

The *BODYTEXT* statement defines the text displayed in the panel header line.

The *CELL* statement is used to define the various information units to be displayed for a status component and their placement in the body section. New is that you can specify the first and last character of the data to be moved to the panel field.

## PANEL definition statement



Left panel, right panel, and ignore are also options

© Copyright IBM Corporation 2019

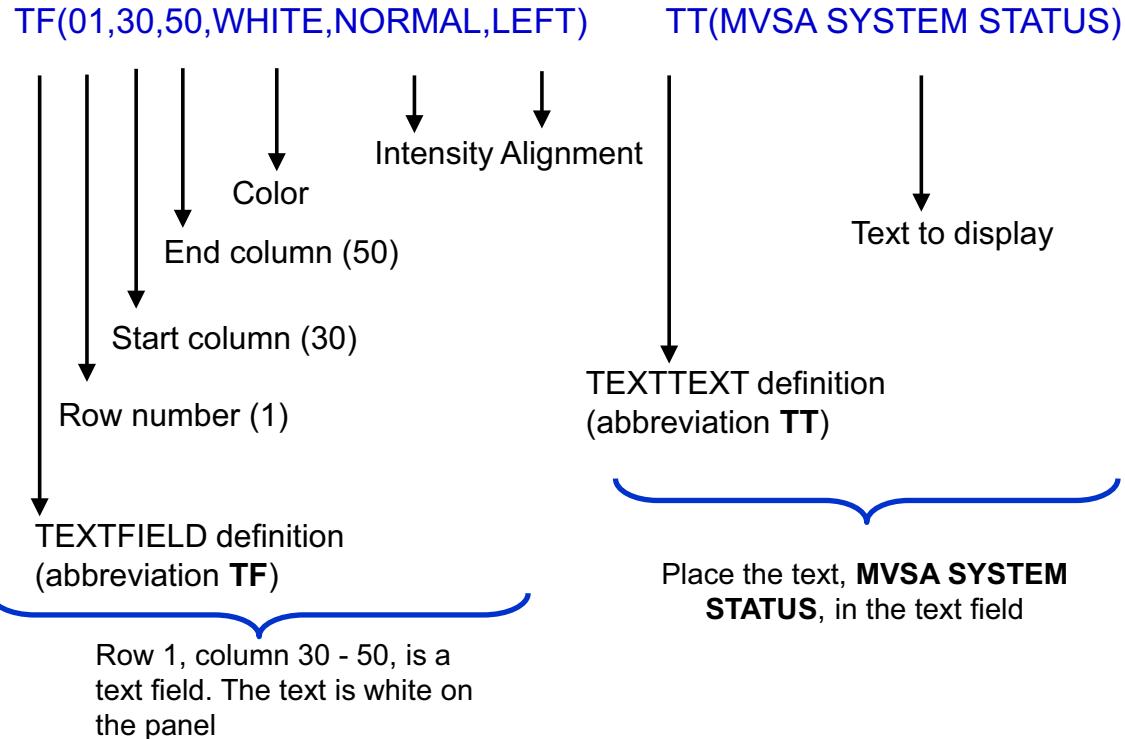
7-29

### PANEL definition statement

In this example, the **PANEL** statement defines the panel AOFPSYSA. The screen size is 24x80. Other screen sizes are supported: 32x80, 42x80, 27x132, and 62x160. The screen size can be determined at runtime. If the user presses either PF7 or PF12, the PRODSYS panel is displayed. Panel PRODSYS is defined the same way. No panel is defined for PF8. If users press that key, they see a message that there is no down panel. There are also options for left\_panel, right\_panel, and ignore.

For left\_panel, you see the panel when the left panel PF Key is pressed or the LEFT command is issued. The panel name can be an asterisk (\*) for vertical scrolling through the displayed information of a body section. The right\_panel works the same way. Using IGNORE causes SDF to ignore the UP/DOWN command when used in a BODY section.

## Defining text fields: TF and TT



© Copyright IBM Corporation 2019

7-30

### Defining text fields: TF and TT

If you want headings or explanations for your panel, you specify the type and location by using the TEXTFIELD and TEXTTEXT statements. A TEXTFIELD (TF) statement identifies where on the screen a text field is. The statement also indicates what color the text is, and the characteristics; for example, normal or reverse video. The TEXTTEXT statement must follow a TEXTFIELD statement to indicate the text that is displayed in the text field. It can be uppercase, lowercase, or mixed case.

This example defines a Text Field (TF) on the panel in row 1 that starts in column 30 and ends in column 50. The text attributes are white with normal intensity. The text to display in the field is defined by the TT definition, in this case, **MVSA SYSTEM STATUS**. You cannot specify row 1, column 1. The 3270 processing always prefixes each field with a 1-byte character for the attributes of the field. Typically, you cannot select column 1.

## Defining text fields: Example

TF(01,30,50,WHITE,NORMAL)  
TT(MVSA SYSTEM STATUS)

| MVSA  | MVSA SYSTEM STATUS |           |           |       |         |
|---|--------------------|-----------|-----------|-------|---------|
| Subsystem   | Subsystem          | Subsystem | Groups    | Reply | Jobname |
| AM2   | APPC               |           | AM_X      |       |         |
| AM  | ASCH               |           | COREAPPLS |       |         |
| RESOLVER  | TCPIP              |           | LOOKASIDE |       |         |
| VLF   | TELNET             |           | BASE_SYS  |       |         |
| AUTONETV  | CMDRCVR            |           | BASE_USS  |       |         |
| LLA   | VTAM               |           | NETWORK   |       |         |
| AUTOMGR   | JES2               |           |           |       |         |
| DLF   | AUTOSSI            |           | MVS Comps |       |         |
| RMF   | RV01               |           |           |       |         |
| RMFGAT  | RV02               |           |           |       |         |
| TSO   | RV03               |           |           |       |         |
| OMVS  | RV04               |           |           |       |         |
| <br>Spool ----> JES2  |                    |           |           |       |         |
| <br>Gateway -->   |                    |           |           |       |         |
| <br>====> =   |                    |           |           |       |         |
| 1=HELP 2=DETAIL 3=RETURN 4=DELETE 6=ROLL 7=PREV SCR 10=LEFT 11=RIGHT 12=TOP |                    |           |           |       |         |

© Copyright IBM Corporation 2019

7-31

### Defining text fields: Example

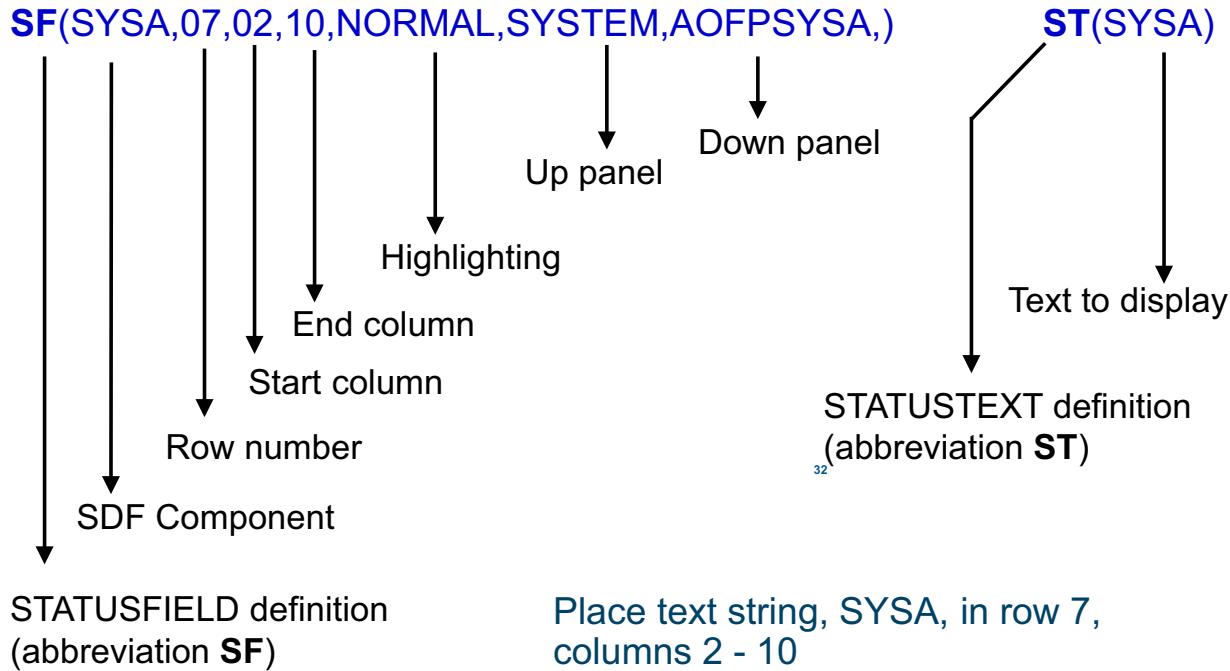
On this sample MVSA System Status panel, you see the result of the TT and TF examples on the previous slide. In this case, the text MVSA SYSTEM STATUS is displayed in black because a white background is used for the panel (MVSA).

Another example of TT and TF is the definition of MVSA in row 1, columns 2 through 10, as follows:

TF(01,02,10,BLUE,NORMAL)

TT(MVSA)

## Defining fixed text status fields: SF and ST



### Defining fixed text status fields: SF and ST

The content of status fields is displayed in a color that is appropriate to the status of some resource. The text displayed can be fixed, or it can relate to the resource, or *status descriptor*. In this example, fixed text is displayed as the name of a system. However, its color is determined by the status of a resource, the aggregate SYSA.

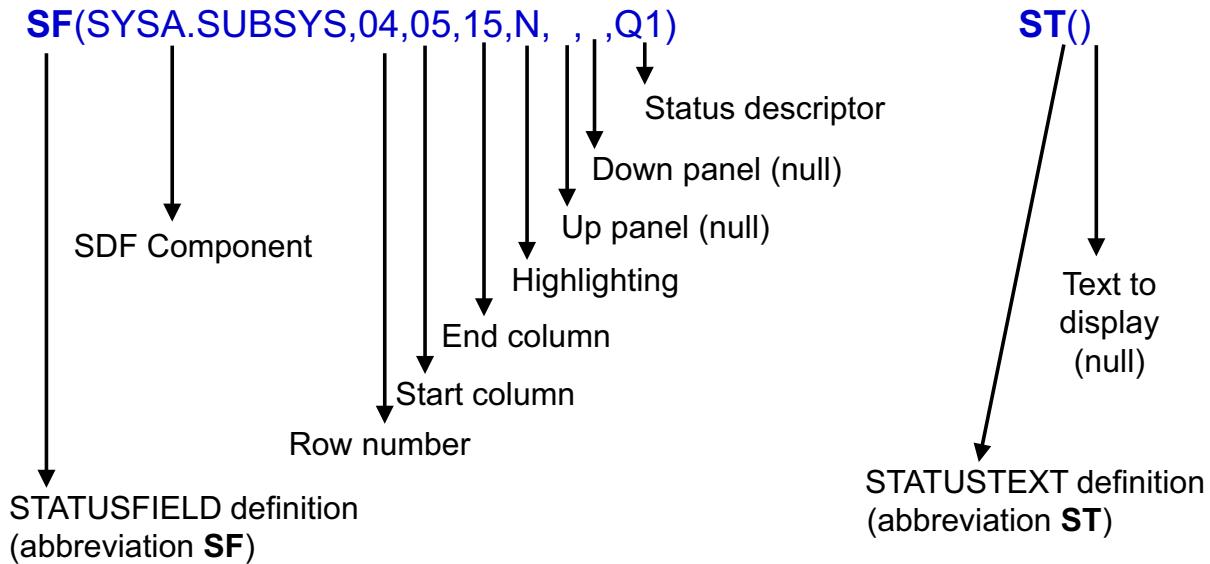
The STATUSFIELD indicates the location of the field. The first positional value is the resource that is used to determine how it is displayed; for example, color. In this case, the resource is an aggregate, and the status is the highest priority status at that resource. If it is CICSB, in its BROKEN status, the field displays in reverse video red.

The UP and DOWN panel definitions indicate the panel that is displayed if the cursor is positioned on this field and PF7 or PF8 is pressed. In this example, pressing PF8 displays panel AOFPSYSA, which displays the status of all resources in system SYSA. A function is defined to see more detailed information. Using a subsequent STATUSTEXT definition here indicates that the text SYSA is always defined. For more description, you can add, for example, *London Online Server*.

This example also illustrates the use of an aggregate. A status field text can equally reference a resource such as SYSA.CICSA. In that case, its status text might read *C/CS Order Entry*.

*Application.* The color and display attributes that are used are those of the highest status record for that aggregate.

## Defining dynamic status fields: SF and ST



Status descriptors; for example, Q1 is used instead of fixed text, making the SDF panel fields dynamic

### Defining dynamic status fields: SF and ST

In the second STATUSFIELD example, the status descriptor name is one of the status types (or leaves) in the tree structure. It is SUBSYS, or subsystem. Which of the many subsystems are used to create the field? The STATUSTEXT specification indicates no text (it can be omitted).

The answer is in the final operand, the status descriptor. This operand is specified as Q1, and it means that first record at the SUBSYS level is used. Because each SUBSYS record has many fields, the Q indicates that the reference value field of the record is used. This field displays the job name of the first SUBSYS record and shows the job name of the most critical subsystem being monitored on SYSA. The color and other attributes of the display field are relevant to the first SUBSYS entry.

By using the numbered form of status value, you can show lists of resources in their priority sequence. This approach is more practical if you want to display resources without having to hard-code the panel fields for each resource.

## Status descriptor use

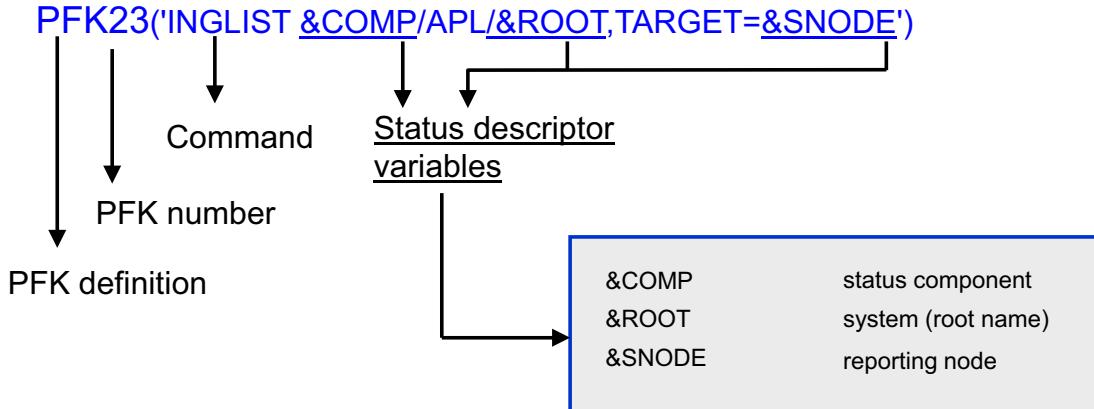
- Status descriptor
  - Color from status of *n*th item at a tree level
  - Text from some attribute of *n*th item
- Possible attributes:
  - A: Alternate status component
  - B: Major status component
  - C: Component name
  - D: Date
  - M: Message text
  - O: User data
  - P: Priority
  - Q: Reference value
  - R: Root component
  - S: reporting operator ID
  - U: Number of duplicate records
  - V: Jobname
  - T: Time
  - W: full name of the status component
  - X: reporting domain ID

### *Status descriptor use*

Many fields are available for building your SDF panel. These fields are described in the *IBM System Automation for z/OS Programmer's Reference* manual. To see how the fields are used, look at the Detail Record of an SDF display (**PF2**).

The status descriptor number may be prefixed with a letter denoting the type of information to be displayed. If no prefix is supplied, the MVS job name is displayed if the resource is a subsystem or WTOR. Valid prefixes are on the chart.

## Defining PF keys



- Use variables to substitute data when cursor is on a status field
- You should assign the `SDFCONF` command to the PF4 key:  
`PFK4=SDFCONF &ROOT, &COMPAPPL, &RV, &SID, &SNODE, &DATE, &TIME, &DA`

Variables for the `PFKn` and `DPFKn` parameters are documented in the *IBM System Automation for z/OS Programmer's Reference* manual

© Copyright IBM Corporation 2019

7-35

### Defining PF keys

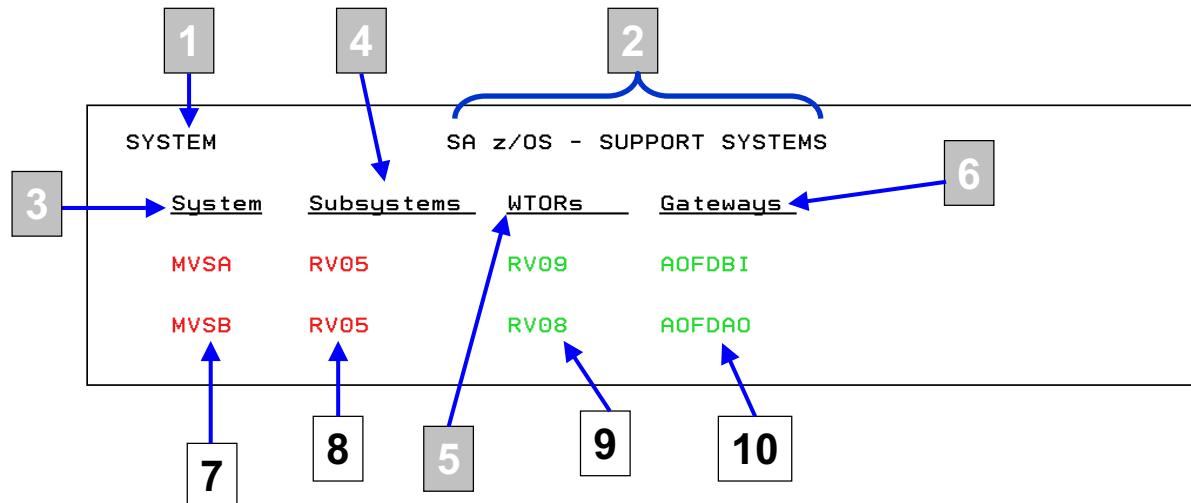
You can define a PF key so that it performs an action on the field on which the cursor is positioned. In other words, you define **PFKn** to perform an action when the **PFnn** key is pressed. This example shows that pressing PF23 runs an INGLIST command. The command operands are the name of the subsystem, system information, and the node reporting the problem.

If the cursor is not positioned on a field, you see an error message. The message indicates that SDF cannot resolve the meaning of the status descriptor variables (`&COMP`, `&ROOT`, and `&SNODE`).

You can also run a hard-coded command, such as INGLIST with no parameters. In this case, the cursor can be positioned anywhere when you press the PF key. It always runs INGLIST.

You might want to define PF keys to invoke commands; for example, for Subsystems: DISPINFO, INGINFO, INGREQ, EXPLAIN, or INGLIST; or for Gateways: DISPGW.

## Example of SDF panel and field definitions



© Copyright IBM Corporation 2019

7-36

### Example of SDF panel and field definitions

This example shows an enterprise SDF panel with ten fields defined as follows:

#### 3. Panel Name = **SYSTEM**

Text Field:

```
TF(01,02,10,WHITE,NORMAL)
TT(SYSTEM)
```

#### 4. Panel Title = **SA z/OS - SUPPORT SYSTEMS**

Text Field:

```
TF(01,23,58,WHITE,NORMAL)
TT(SA z/OS - SUPPORT SYSTEMS)
```

#### 5. Column 1 Heading = **System**

All resources displayed in this column are systems.

Text Field:

```
TF(03,05,10,T,U)
TT(System)
```

**6. Column 2 Heading = Subsystems**

All resources displayed in this column are Subsystems (Applications).

This is a *Text Field*:

TF (03,14,24,T,U)

TT (Subsystems)

**7. Column 3 Heading = WTORs**

All resources displayed in this column are Application WTORs.

This is a *Text Field*:

TF (03,27,34,T,U)

TT (WTORs)

**8. Column 4 Heading = Gateways**

All resources displayed in this column are Gateway sessions.

This is a *Text Field*:

TF (03,37,45,T,U)

TT (Gateways)

**9. Column 1 Data = MVSB**

*Status Field* (contains the fixed character string MVSB and is an aggregation of the status of the resources for system MVSB):

SF (TIVED2,05,05,10,N,,MVSBCOMP)

ST (MVSB)

**10. Column 2 Data = RV05**

*Status Field* (contains the Application with the highest priority in its status descriptor).

SF (TIVED2.APPLIC,05,14,24,N,,MVSBCOMP,Q1)

**11. Column 3 Data = RV08**

*Status Field* (contains the job name of the Application with an outstanding WTOR with the highest priority in its status descriptor).

SF (TIVED2.APPLIC,05,14,24,N,,MVSBCOMP,Q1)

**12. Column 4 Data = AOFDAO**

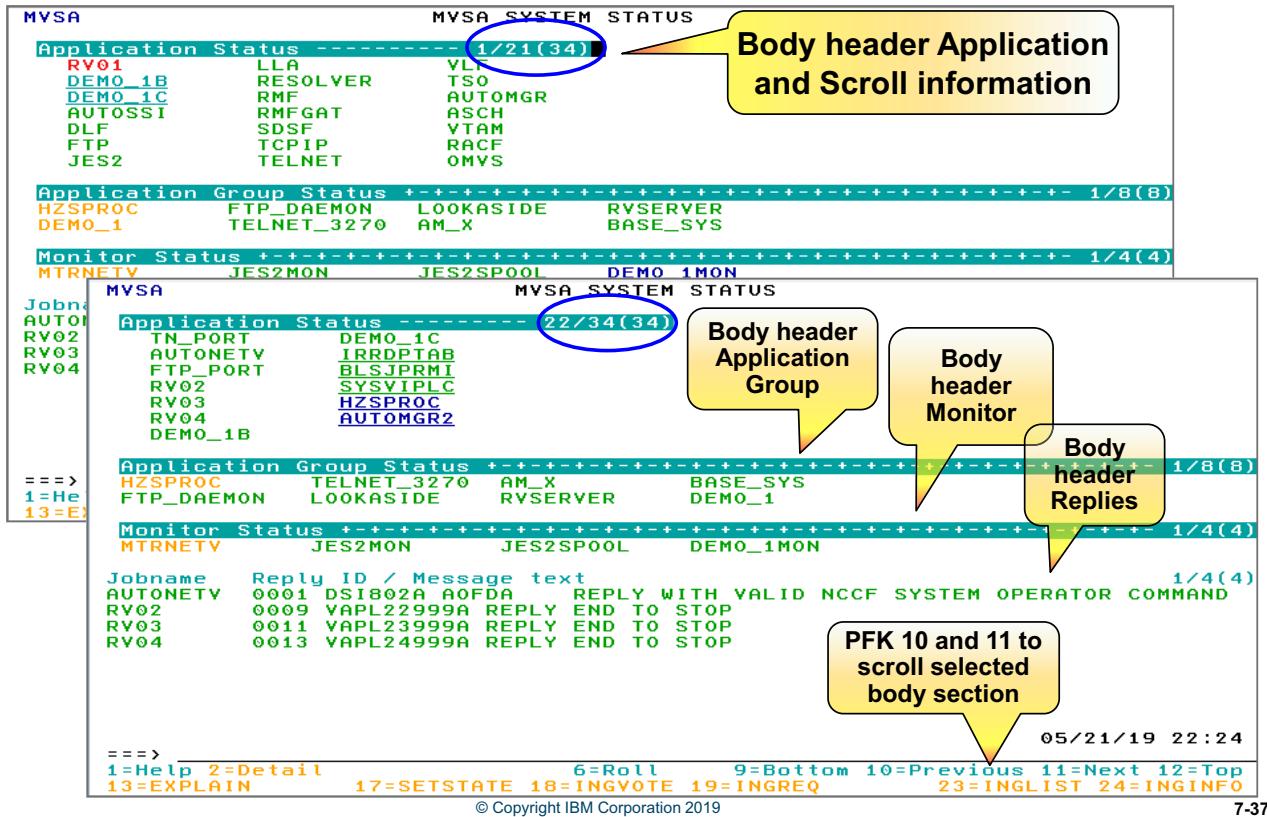
*Status Field* (contains the Gateway session with the highest priority in its status descriptor):

SF (TIVED2.GATEWAY,05,37,45,N,,MVSBCOMP,1)

These ten fields are a subset of the complete panel definition. You can also define PF key definitions and other fields.

# SDF multiple body sections example

## BODYHEADER and BODYTITLE statements



## *SDF multiple body sections example*

This slide shows an example of a panel definition that uses multiple body sections.

1. The first body section displays scroll-enabled application entries.
  2. The second body section displays scroll-enabled groups.
  3. The third body section displays scroll-enabled monitors.
  4. The fourth body section displays scroll-enabled replies.

The scroll information is shown to the right of the body title.

The screenshot at the bottom has scrolled the first body section which displays applications

The next slide shows the panel definitions that are used to create this panel.

# Body statements

## Scrollable body

```

P(MVSATEST,32,80,SYSTEM,SYSTEM,,*,*)
TF(01,02,10,BLUE,NORMAL)
TT(MVSA)
TF(01,30,50,WHITE,NORMAL)
TT(MVSA SYSTEM STATUS)
/* Subsystems
BODY(MVSA.SUBSYS,04,10,*,1,02,40)                                     */
BH(03,,,T,R,-)
BT(Application Status)                                                 Start line,
CELL(04,12,N,C)                                                       End line
/* APGs
BODY(MVSA.GROUPS,13,14,*,1,02,80)                                         */
BH(12,L,S,T,R,+-)
BT(Application Group Status)                                              number of columns (*) entire BODY,
CELL(02,12,N,C)                                                       distance between columns is 1
/* MTRs
BODY(MVSA.MONITOR,17,17,*,1,02,80)                                         */
BH(16,L,S,T,R,+-)
BT(Monitor Status)                                                 Start column, end column
CELL(02,12,N,C)                                                       End line with displacement from the bottom line
/* WTORS
BODY(MVSA.WTOR,20,*-5,1)                                                 */
BODYHEADER(19,,,T)                                                 Start line, alignment, scroll information, color
BT(Jobname Reply ID / Message text)
CELL(02,11,N)
CELL(12,*,N,M)
/* PF Keys
PFK13('EXPLAIN #COMP,TARGET=#SNODE')
PFK17('SETSTATE #COMP,TARGET=#SNODE')
...

```

Asterisk (\*) indicates that you can scroll the body section by using PF keys

### Body statements

The definition for this panel does not contain any status fields.

The **BODYHEADER** or BH statement is used to define a descriptive header and the scrolling information of the body section. The length of the header is implicitly defined by the body width.

The **BODYTEXT** or BT statement defines the text displayed in the panel header line.

The **BODY** statement is used to display the various status components listed in the order of their priority:

1. MVSA is the root component name as defined in the root node of the tree structure.
2. SUBSYS is the major component name as defined in the AOFTREE member.
3. The line number where the body section begins is: 4.
4. The line number where the body section ends is: 10.
5. \* is . the number of columns to be generated. SA z/OS attempts to fill the entire BODY width with columns.
6. The distance between columns is 1.

7. The column number where the body section begins is: 2.
8. The line column where the body section ends is: 40.
9. sync\_scroll is not specified.
10. XO for sorting is not specified.

# Lesson 5. Customizing SDF parameters

## Lesson 5: SDF Definition Process

- Define the hierarchy of monitored resources
- Define SDF status panels
- Customize the SDF initialization parameters in NetView DSIPARM member AOFINIT
- Define SDF resource status, color, highlight and priority values using the customization dialog (optional, next lesson)

### What this lesson is about

This lesson describes how to customize SDF parameters.

### Objectives

After completing this lesson, you should be able to customize SDF parameters.

### References

*SC34-2644 System Automation for z/OS Version 4 Release 1 Customizing and Programming*

SDF runs as a separate task in NetView, and it has some initialization definitions that you must understand. These definitions define parameters such as the tree structure to use, the panel definitions, and general parameters, for example.

These definitions are typically read as the SDF task (named AOFTDDF) initializes. You can also use the SDFPANEL and SDFTREE commands to update the SDF panels or tree structure while SA z/OS is running.

## DSIPARM customization and members

- AOFINIT: SDF initialization statements
- AOFTREE: Define SDF tree structures
- AOFPNLS: Anchor point for defining panels and points to other members with panel definitions by using %INCLUDE
- Data REXX functionality and system symbols supported
- Static and dynamic SDF panels
- NetView style sheet Advanced Automation Options CGLOBALS

```
COMMON.AOF_AAO_SDFROOT.0 = 1
COMMON.AOF_AAO_SDFROOT.1 = MVSA MVSB MVSC
COMMON.AOF_AAO_SDFCSAPLEX.0 = 1
COMMON.AOF_AAO_SDFCSAPLEX.1 = SYSPLEX1 SYSPLEX2 PLEX01
```

&SDFROOT variable to  
generate dynamic panels

&SDFCSApex variable for  
Config Refresh Monitoring

enterprise  
SAplexes

### DSIPARM customization and members

The DSIPARM data set contains the following SDF-related members:

- AOFINIT: SDF initialization statements.
- AOFTREE: Define SDF tree structures.
- AOFPNLS: Anchor point for defining panels and points to other members with panel definitions by using %INCLUDE.
- Static and dynamic SDF panels.

### NetView style sheet Advanced Automation Option (CGLOBALS) for SDF

#### AOF\_AAO\_SDFROOT

- Defines the value of the &SDFROOT variable that is used as the root name for the SDF panels.
- When used, the variable AOF\_AAO\_SDFROOT.0 must exist and defines the number of adjacent globals.
- The value of an adjacent global can be the name of a single system or a list of system names separated by a blank character. A list can be used at the SDF focal point to have SA z/OS

generate the necessary panel definitions for all systems in the list. The system name can be appended by one or more member names separated by slashes, the first name refers to the panel definitions, and the second name refers to the tree definitions, for example:

```
SYS1 SYS2/MYPNLS SYS3&SLASH./MYTREE SYS4/MYPNLS/MYTREE.
```



**Note:** Special Note for Clients with MVS &SYSNAME greater than 4 characters length:  
the SDF Root and COMMON variable representing the Root values must be updated to be 4 or less characters in length.

Also update the SDF Root value in the policy SYSTEM area, under item SYSTEM INFO, so only 4 characters are used, and ensure that value matches an element in the SDFROOT cnmstyle definition.

#### AOF\_AAO\_SDFCxxx.\*

- Names the System Automation subplexes of the entire enterprise.
- Config Refresh Monitoring uses the reserved AAO AOF\_AAO\_SDFCsaplex.n.
- The sample above defines three SAplexes

#### AOF\_AAO\_SDFCxxx.\* (not shown as it is for special replication only)

- Defines the value of the user-defined &SDFCxxx variable that is used replicate sub-trees. Like the variable &SDFROOT, which enforce SDF to replicate a complete tree definition as many times as it finds different values in the corresponding common globals you may also use variables for defining component names to enforce SDF to replicate sub-trees. Each variable should be defined in the CNMSTYLE sheet when the variable must be resolved at SDF initialization time and must begin with &SDFC and end with a period. The characters in between must follow the NetView rules of defining common globals.

## AOFTREE definition

- Member in DSIPARM: Name must be AOFTREE
- Use %INCLUDE combined with tree structure definitions in AOFTREE: %INCLUDE(name)
- Use system symbols
- &SDFROOT defaults to local system or AAO  
COMMON.AOF\_AAO\_SDFROOT
- Tree structures are loaded during initialization
  - Load additional tree structures using the SDFTREE command

root component

Simple example

```
1 &SDFROOT.  
2 APPLS  
3 SUBSYS  
2 APG  
3 GROUPS  
2 MONITOR  
2 WTOR  
2 CPMSSGS  
2 GATEWAY
```

### AOFTREE definition

The AOFTREE definition has a fixed name in the DSIPARM data set, and must be present. It contains the definition of all the tree structures that are used by SDF. To define the structures, you can use a %INCLUDE format. It has a different syntax than the %INCLUDE that is used throughout basic NetView. When you BROWSE a member such as AOFTREE, you do not see the expanded full content. AOFTREE definitions are typically small. You can edit everything into one member.

SA z/OS supplies a sample SDF tree structure in the SA z/OS sample library. This tree structure is referenced by a %INCLUDE statement in member AOFTREE in the NetView DSIPARM data set. You can customize this sample tree structure to meet your requirements. This order of dependency does not have to be the same as that used for system startup or shutdown using SA z/OS. System symbols are supported for the tree structure.

In this tree structure, the &SDFROOT variable contains the name of the root component.

Tree structures loaded after SDF is started must be contained in separate members. Each member must be named after the root component for which the tree structure is defined.

## AOFPNLS definition

- Member in DSIPARM; name must be AOFPNLS
  - Use %INCLUDE combined with panel definitions
  - Use system symbols or your own variables
  - Can contain one, many, or no panel definitions
- Dynamic panels
  - Use &SDFROOT in some SDF statements, especially panel name and any place where the root name is used, like TT or SF
  - A panel is generated for each system name in AOF\_AAO\_SDFROOT.n

|                     |         |
|---------------------|---------|
| %INCLUDE (INGPTOP)  | STATIC  |
| %INCLUDE (INGPMAIN) | DYNAMIC |
| %INCLUDE (INGPAPL)  | DYNAMIC |
| %INCLUDE (INGPAPG)  | DYNAMIC |

### Note: Uses nonstandard %INCLUDE format

© Copyright IBM Corporation 2019

7-42

#### AOFPNLS definition

The AOFPNLS definition also has a fixed name. AOFPNLS contains one or more panel definitions. However, it is not required at initialization time. AOFPNLS uses a nonstandard %INCLUDE facility that is unique to SDF. The panel names are defined by the PANEL statements in the definitions:

- Use %INCLUDE combined with panel definitions
- Use system symbols or your own variables.  
This can help reduce both customization work and errors.
- Can contain one, many, or no panel definitions

#### Dynamic panels

Dynamic panels use the &SDFROOT variable in some SDF statements, especially panel name and any place where the root name is used, like TT or SF statements

You can specify the option STATIC or DYNAMIC for the SDF %INCLUDE statement. If you specify DYNAMIC, this generates the panel definitions for all of the system names that you specify in AOF\_AAO\_SDFROOT.n common global variable.

The SDF %INCLUDE statement requires parentheses around the specified member or members.

## AOFINIT definition

- Member in DSIPARM  
Name must be AOFINIT
- Basic statements for SDF initialization
- Sample member in SINGNPRM
- Possible changes:
  - Set initial panel to SYSTEM only for focal point system using Data REXX
  - Increase number of users
  - Modify default colors
  - Modify default PF keys
  - Parameter MAXTREEDSPSZ for data space size of tree

```
SCREENSZ=10000,32,80
%> SELECT
%> WHEN Cursys() = 'MVSG' Then Do
INITSCRN=SYSTEM
%>End
%>Otherwise Do
INITSCRN=&SYSNAME.MAIN
%>End
%>End
MAXOPS=10
PROPUP=YES
PROPDOWN=NO
TEMPERR=3
PFK1=AOCHELP SDF
.
PRIORITY=1,199,RED
PRIORITY=200,299,PINK
PRIORITY=300,399,YELLOW
PRIORITY=400,499,TURQUOISE
PRIORITY=500,599,GREEN
PRIORITY=600,699,BLUE
DCOLOR=WHITE
EMPTYCOLOR=BLUE
```

### AOFINIT definition

The AOFINIT definition is the basic set of statements that define overall SDF operation. These statements are contained in a DSIPARM member that must be named AOFINIT. A sample AOFINIT is supplied in the SINGNPRM data set that is distributed by IBM. You can typically use AOFINIT as is. At most, you might want to make a few alterations.

The statements are described in the following list:

- **INITSCRN** defines the panel that is shown if a user enters SDF with no operands. This defines the default panel to display and typically some higher-level or summary panel. The IBM default setting is **SYSTEM**.
- **MAXOPS** defines the maximum number of users who can use SDF simultaneously. If this value is too low, some users receive errors when they run the SDF command.
- The **PRIORITY** definitions provide default values for colors, if none are used in the calls to update SDF. The priority values used by the SA z/OS automation agent are defined in the customization dialog (option 42, SCF, Status Details). These values are used only if you specify an empty color value in the customization dialog. If nothing is defined, then defaults are set by the agent during initialization. For standard SDF operations, you do not have to alter these definitions.

- **DCOLOR** indicates the default color that the status field has if the field definition refers to a root or component that is not defined in the AOFTREE definition, which indicates a specification error.
- **EMPTYCOLOR** indicates the color a status field displays on your screen if there is no status record matching your definition.

Parameter **MAXTREEDSPSZ** for data space size of tree. Refer to the MAXTREEDSPSZ parameter description in IBM System Automation for z/OS Programmers Reference for calculating and defining the size of the data space.

## Using a PANEL definition

- Define in AOFPNLS by PANEL definition
    - Panel definitions are preloaded into storage
    - Member is not read each time it is accessed
  - Search DSIPARM for a member named panel\_name
    - If found, check for PANEL statement and use if valid
    - Member is read each time the panel is accessed
- Note: NetView automatically loads commonly used members in storage

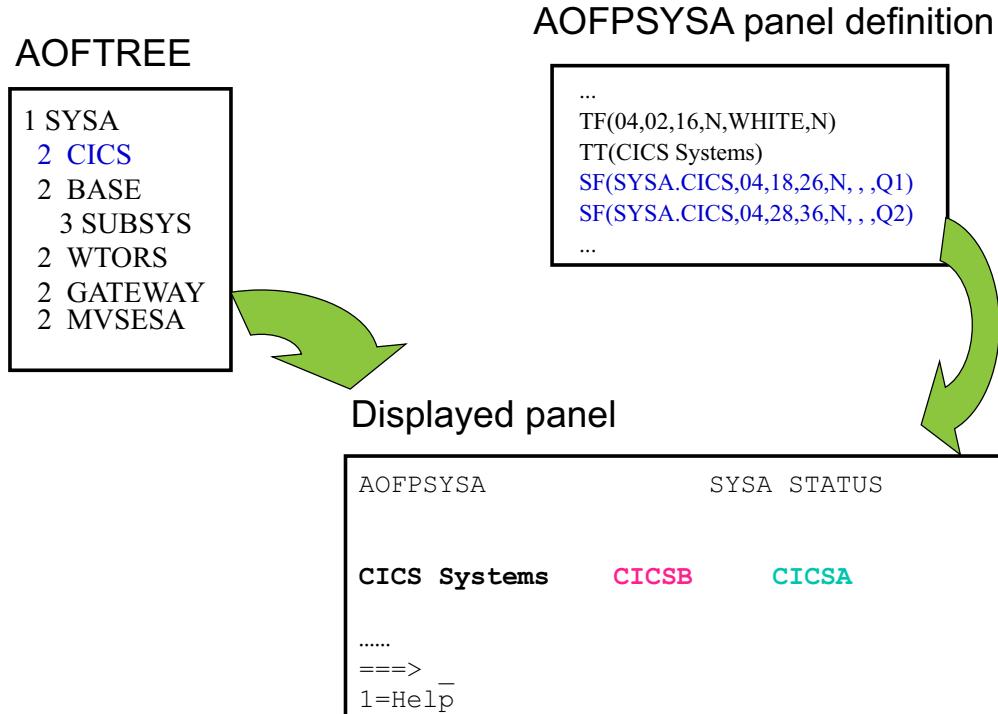
### Using a PANEL definition

When SDF panels are displayed, SDF searches for the panel in the static set of panels that are defined in the AOFPNLS definition. If the panel (identified by the PANEL statement) is found, SDF uses that definition. The definition used is the one that is read at SDF initialization time.

Otherwise, SDF searches the DSIPARM data sets for a member with the same name as the panel. If the panel is found, SDF reads the DSIPARM member and looks for a PANEL statement. If it matches, the definition is used. If it does not match, then a panel failure occurs.

The DSIPARM member is read each time the panel is initially accessed by the user. The load time is slower than for static panels. However, a panel might not be read often. With this approach, you are also using the current definition of that panel. During testing, this approach might be the best way to manage your panels.

## Extending the tree



### Extending the tree

This example creates a new aggregate at level 2 of the tree definition, and uses this definition to collect the CICS systems. When an update is received for CICSA or CICSB, it is collected at the CICS aggregate (and SYSA). Although CICSA and CICSB are subsystems, their status is not collected by SUBSYS at level 3. The aggregate BASE collects everything except CICSA and CICSB. CICSA and CICSB are in the CICS aggregate, which is where they are aggregated. A status descriptor can be aggregated into only one aggregate at one level. By using this definition, you can create panels that show the overall status for all CICS systems and all the other subsystems. The Q1 status descriptor always displays the resource with the most critical problem.

## Extending the tree: Config Refresh Monitoring

AOFTREE (sample INGTCFG)

```
1 INGCFG  
2 &SDFCsaplex.  
3 AGENT
```

Displayed panel

INGPCFG panel definition in member AOFPNLS

```
P(INGCFG,*,*,SYSTEM,SYSTEM,,*,*  
:  
BODY(INGCFG,05,*-6,6,03,79,,VA)  
BH(01,L,S,T,N,68,79)  
CELL(02,09,N,V)  
CELL(11,26,N,B)  
CELL(28,35,N,C)  
:
```

| MVSB    | Configuration Refresh |        |   |
|---------|-----------------------|--------|---|
| Sysplex | SApplex               | System | ! |
| PLEX12  | SYSPLEX1              | MVSA   |   |
| PLEX12  | SYSPLEX1              | MVSB   |   |

CNMSTYLE sheet definitions

```
COMMON.AOF_AAO_SDFCSAPLEX.0 = 1  
COMMON.AOF_AAO_SDFCSAPLEX.1 = SYSPLEX1 SYSPLEX2
```

### Extending the tree: Config Refresh Monitoring

This example implements Configuration Refresh Monitoring.

The tree and panel definitions are defined in the members INGTCFG and INGPCFG.

You also must add CNMSTYLE sheet definitions for AOF\_AAO\_SDFCSaplex.n variables.

Since only active systems are displayed on the INGPCFG panel, you can share the definition across all systems no matter whether the information is available or not.

The BODY statement of panel INGPCFG defines two logical columns. To keep the column header of each logical column identical, we cannot use the BH statement for displaying the header. Otherwise, we would see the number of systems on the header of the right column. Instead, we use TF and TT statements for defining each header. The associated BH statement is now located in the upper-right corner of the screen and just displays the number of systems that are active.

## Config Refresh Monitoring Detail Status

```
1 of 1          ---- Detail Status Display ----      04/19/19 22:33:48

Component . . . : MVSB           System . . . . . :
Color . . . . . : GREEN          Priority . . . . . : 650
Date . . . . . : 04/19/19        Time . . . . . . : 22:26:01
Reporter . . . . : AUTMSG        Node . . . . . . : A0FDB

Info . . . . . : PLEX12   SYSPLEX1       MVSB
Reference value : SYSPLEX1_MVSB
User data . . . . :

A0F031I Configuration Refresh on 'MVSB' is 'COMPLETE'.
```

The configuration status of the agents that are shown in SDF have the following status definitions:

| Status   | Priority | Color  | Highlight | Clear  |
|----------|----------|--------|-----------|--------|
| CFG_PROB | 100      | Red    | Reverse   | Y, RV* |
| CFG_REFR | 450      | Yellow | Normal    | Y, RV* |
| CFG_COMP | 650      | Green  | Normal    | Y, RV* |

© Copyright IBM Corporation 2019

7-47

### Config Refresh Monitoring Detail Status

This is the Config Refresh Monitoring Detail Status.

SAplex SYSPLEX1 in sysplex PLEX12 has been successfully refreshed in system MVSB.

## Refreshing SDF

- Load additional tree structures using the SDFTREE command
- Load panels using the SDFPANEL command
- If you change the SDF definitions, refresh SDF with RESYNC command:

- **RESYNC SDF:**

Reinitializes the status information for the local SDF and forwards it to the SDF focal point

- **RESYNC SDFDEFS:**

Generates the SDF panels using the advanced automation option (AAO) AOF\_AAO\_SDFROOT.n for the SDF root names that are to be applied

| Sel | System | Message   |
|-----|--------|---|
| —   | MYSA   | AOF001I REQUEST "RESYNC" WAS SUCCESSFUL FOR SDFDEFS |

- **RESYNC FP:**

Resends the status information for the local SDF to the SDF focal point

- Starting the SDF Task: START TASK=AOFTDDF
- Stopping the SDF Task: STOP TASK=AOFTDDF

### Refreshing SDF

- Delete tree structures or load additional tree structures from the NetView DSIPARM data set by using the SDFTREE command
- Load panels from the NetView DSIPARM data set by using the SDFPANEL command
- If you change the SDF definitions, refresh SDF with the RESYNC command:
  - **RESYNC SDF:**  
Reinitializes the status information for the local SDF and forwards it to the SDF focal point
  - **RESYNC SDFDEFS:**  
Generates the SDF panels using the advanced automation option (AAO) AOF\_AAO\_SDFROOT.n for the SDF root names that are to be applied
  - **RESYNC FP:**  
Resends the status information for the local SDF to the SDF focal point
- Starting the SDF Task:  
`START TASK=AOFTDDF`
- Stopping the SDF Task:  
`STOP TASK=AOFTDDF`



**Note:** When SDF is restarted, all existing SDF status descriptors are lost, as they are kept only in memory.

## Primary and backup focal points (APAR OA55386)

- SDF supports one primary focal point (PFP) and one backup focal point (BFP)
- Target systems can switch focal points automatically
- SDF also supports *parallel update* of both focal points
  - Use SDF FOCALPOINT policy of Network (NTW) entry type
  - All systems can share one definition
- NetView style sheet Advanced Automation Options

```
COMMON.AOF_AAO_SDFFPP_ROOT.0 = 1
COMMON.AOF_AAO_SDFFPP_ROOT.1 = ROOTFP

COMMON.AOF_AAO_SDFBFP_ROOT.0 = 1
COMMON.AOF_AAO_SDFBFP_ROOT.1 = ROOTBFP
```

Defines the static root names of the primary focal point

Defines the static root names of the backup focal point

© Copyright IBM Corporation 2019

7-49

### Primary and backup focal points

SDF supports one primary focal point (PFP) and one backup focal point (BFP). The target systems can switch focal points automatically.

SDF also supports **parallel update** of both focal points.

Use SDF FOCALPOINT policy of Network (NTW) entry type to set the Parallel Update field to YES. All systems can share the same definition unless the focal point system should not forward its local data to the other focal point system

### NetView style sheet Advanced Automation Options

#### AOF\_AAO\_SDFBFP\_ROOT.\*

Defines the static root names of the backup focal point.

#### AOF\_AAO\_SDFFPP\_ROOT.\*

Defines the static root names of the primary focal point.

The value of an adjacent global can be the name of a single root name or a list of root names that

are separated by blank characters. The globals are evaluated only when the primary focal point and the backup focal point are updated in parallel.

If you have defined SDF root names besides the local system name, you need to make them available for synchronization with the RESYNC FP command. Assuming your PFP is named SY1P and has an additional root name PFPONLY, and the BFP is named SY2B with a similar root name BFPONLY, the following style sheet definitions shows you how to specify them:

```
%>select
%> when cursys() = 'SY1P' then
%> do
/* PFP has defined PFP=SY1P and BFP=SY2B */
COMMON.AOF_AAO_SDFFPP_ROOT.0 = 1
COMMON.AOF_AAO_SDFFPP_ROOT.1 = PFPONLY
%> end
%> when cursys() = 'SY2B' then
%> do
COMMON.AOF_AAO_SDFBFP_ROOT.0 = 1
COMMON.AOF_AAO_SDFBFP_ROOT.1 = BFPONLY
%> end
%> otherwise
%> nop
%>end /* select */
```

# Lesson 6. SDF policy customization

## Lesson 6: SDF policy customization

### Define the SDF root name

```

COMMANDS HELP
AOFGSPDO System Information
Command ===> _____
Entry Type : System PolicyDB Name : NEW_PDB
Entry Name : MVSBN Enterprise Name : SAPLEX
Operating system : MVS
Image/System name. . . . . mvsb
The following specifications are for MVS systems only:
Primary JES. . . . . JES2 Primary JES2/JES3 subsystem name
System monitor time. . . 00:59 Time between monitor cycles (hh:mm or NONE)
Gateway monitor time . . 00:15 Time between monitor cycles (hh:mm or NONE)
Automation table(s). . . INGMSG01
SDF root name. . . . . Root of system's SDF tree
Exit name(s) _____ Environment setup user exit names
USS automation path. . .
/usr/lpp/ing/ussauto/lib
SA NetView domain. . . . aofdb System Automation UNIX installation path
Network NetView domain . . . . . NetView domain ID of SA z/OS
NetView domain ID of network automation

```

SYSTEM INFO policy item: The default SDF root name is the system name

Animated

© Copyright IBM Corporation 2019

7-50

### What this lesson is about

This lesson reviews the policy definitions for SDF.

### Objectives

After completing this lesson, you should be able to define System Automation policy for SDF.

### References

*SC34-2717 System Automation for z/OS Version 4  
Release 1 Defining Automation Policy*

Root names are typically set to the name of the system. However, you can change the root name. The SDF root name is defined in the SYSTEM INFO policy item, which is part of the SYStem policy object. The default definition for the SDF root name is the system name. If you modify the **SDF Root Name** field, it must match the root name in the AOFTREE specification. If it does not match, an empty SDF panel is displayed.

## Customize SDF color and priority

| AOFGSCR                     |          | Status Display Facility Details |       |         |         | Row 1 to 20 of 20 |  |
|-----------------------------|----------|---------------------------------|-------|---------|---------|-------------------|--|
| Command ===>                |          |                                 |       |         |         | SCROLL===> CSR    |  |
| Entry Type : Status Display |          | PolicyDB Name : NEW_PDB         |       |         |         |                   |  |
| Entry Name : MY_SDF         |          | Enterprise Name : SAPLEX        |       |         |         |                   |  |
| Status                      | Priority | Highlight                       | Color | Clear   | Srv Req |                   |  |
| STARTED2                    | 230      | UNDERSCORE                      | PINK  | (Y,RV*) |         |                   |  |

- Reasons to change color:

- Some colors are poor quality on 3270 emulators
- Reverse video is easy to see at a distance
- Most people do not like a blinking status

- Reasons to change priorities:

- Might want to see problem statuses in different order
- Link Status Display to system:

| AOFGXC4E            |          | Status Display for System |  |  |  | Row 1 to 1 of 1 |  |
|---------------------|----------|---------------------------|--|--|--|-----------------|--|
| Command ===>        |          |                           |  |  |  | SCROLL===> CSR  |  |
| Entry Type : System |          | PolicyDB Name : NEW_PDB   |  |  |  |                 |  |
| Entry Name : MVSA   |          | Enterprise Name : SAPLEX  |  |  |  |                 |  |
| Action              | Status   | Status Display            |  |  |  |                 |  |
|                     | SELECTED | MY_SDF                    |  |  |  |                 |  |

© Copyright IBM Corporation 2019

7-51

### Customize SDF color and priority

To change the colors or highlighting for the standard SA z/OS statuses, you can alter them by using the customization dialog option **42** (SCR, Status Details). For example, you might not want to have blinking fields. STARTED2 is a status that causes the resource to blink when it is displayed on the SDF panels. In this example, a new SCR entry was created called *MY\_SDF\_STATUS*. You can define the SDF DETAILS policy item to change the highlight from BLINK to UNDERSCORE.

You might want statuses such as CTLDOWN and DOWN to show as higher priority than UP. For example, CLEAR=(Y,RV\*) deletes any existing status. The status descriptors with reference values that are prefixed with the same reference value (followed by anything else) are also deleted.

## Setting the WTOR colors: Messages policy

| AOFGDYNM   |              | Message Processing                                  | Line 00000001 Col 001 075<br>Scroll ==> CSR |   |       |   |       |  |  |  |  |  |  |  |  |  |  |  |
|--|--------------|---|---|---|-------|---|-------|--|--|--|--|--|--|--|--|--|--|--|
| Entry Type   | Entry Name   | PolicyDB Name : NEW_PDB<br>Enterprise Name : SAPLEX |   |   |       |   |       |  |  |  |  |  |  |  |  |  |  |  |
| Line Commands: S/C (Cmd), R (Rep), K (Cod), U (Usr), A,M (AT, MRT, MPF)<br>I, D (insert or delete lines) |              |   |   |   |       |   |       |  |  |  |  |  |  |  |  |  |  |  |
| Message ID field length . . 16 (1 - 32)  |              |   |   |   |       |   |       |  |  |  |  |  |  |  |  |  |  |  |
| Cmd  | Message ID   | Description   | C   | R | K     | U | A M F |  |  |  |  |  |  |  |  |  |  |  |
|  | IEA793A      | No dump datasets available                          |   |   | 1     |   |       |  |  |  |  |  |  |  |  |  |  |  |
|  | IEC613A      | Unable to position a tape                           |   |   | 1     |   |       |  |  |  |  |  |  |  |  |  |  |  |
|  | IEE767A      | Buffer shortage on system log                       |   | 1 |       |   |       |  |  |  |  |  |  |  |  |  |  |  |
|  | IEE768I      | Buffer on system log relieved                       |   | 1 |       |   |       |  |  |  |  |  |  |  |  |  |  |  |
|  | IEE824E      | Catch WTORs after console failure                   |   | 1 |       |   |       |  |  |  |  |  |  |  |  |  |  |  |
|  | IEF238D      | Cannot complete device allocation                   | 4   |   | P     |   |       |  |  |  |  |  |  |  |  |  |  |  |
|  | IEF402I      | Job failed  |   | 8 | P     |   |       |  |  |  |  |  |  |  |  |  |  |  |
|  | IEF450I      | Job abended   |   | 8 | P     |   |       |  |  |  |  |  |  |  |  |  |  |  |
|  | LOGREC       | LOGREC commands                                     | 3   |   | P P P |   |       |  |  |  |  |  |  |  |  |  |  |  |
|  | MVSDUMP      | MVSDUMP Threshold Commands                          | 3   |   | P P P |   |       |  |  |  |  |  |  |  |  |  |  |  |
|  | MVSDUMPRESET | MVSDUMP Reset Commands                              |   | 1 | P P P |   |       |  |  |  |  |  |  |  |  |  |  |  |
|  | MVSDUMPTAKEN | MVSDUMPTAKEN commands                               |   | 1 | P P P |   |       |  |  |  |  |  |  |  |  |  |  |  |
|  | SMFDUMP      | SMFDUMP commands                                    |   | 1 | P P P |   |       |  |  |  |  |  |  |  |  |  |  |  |
|  | CYCLOC       | CYCLOC commands                                     | 1   |   | P P P |   |       |  |  |  |  |  |  |  |  |  |  |  |
|  | WTORS        | Classification of WTORs                             |   | 1 | P P P |   |       |  |  |  |  |  |  |  |  |  |  |  |

Some WTORs are normal. Use CODE values to define how WTORs are displayed in SDF:

- *Normal*: green
- *Unusual*: yellow
- *Important*: pink
- *Critical*: red

### Setting the WTOR colors: Messages policy

By default, all WTORs display in yellow on SDF panels because they are reported with a default priority, indicating that they are *unusual*. However, some WTORs are expected. Many Applications leave a WTOR outstanding while they run, as a means of entering internal commands to the Application. Such WTORs might be considered *normal*.

Use the MESSAGES/USER DATA policy item of an Application or MVS components to identify the types of WTORs. Specify CODEs definitions for the **WTORS** special message ID.

## Setting the WTOR colors: Code processing

| AOFGDYN9      Code Processing : WTORS |         |        |        | Line 00000001 Col 001 075 |
|---------------------------------------|---------|--------|--------|---------------------------|
| Command ==> _____                     |         |        |        | Scroll ==> CSR            |
| Cmd                                   | Code 1  | Code 2 | Code 3 | Value Returned            |
| ____                                  | DSI802A | *      |        | NORMAL PRI                |
| ____                                  | DSI803A | *      |        | NORMAL PRI                |
| ____                                  | *       | *      |        | IMPORTANT SEC             |
| ***** Bottom of data *****            |         |        |        |                           |

For AUTONETV WTORs:

- DSI802A: Return NORMAL PRI to indicate that it is expected and is a primary WTOR for the Application. The WTOR is displayed in SDF in green color
- DSI803A: Return NORMAL PRI to indicate that it is expected and is a primary WTOR for the Application. The WTOR is displayed in SDF in green color
- All other WTORs: Return IMPORTANT SEC to indicate that the WTOR is an error. The WTORs are displayed in SDF in pink color

### Setting the WTOR colors: Code processing

On the Code Processing panel, you can create a single entry for each WTOR that the Application issues.

- The **Code 1** field is the message ID of each WTOR.
- The **Code 2** field is the job name for the Application. In this example, job name is not used.
- In the Value Returned column, you enter the keyword NORMAL if this WTOR is normally produced and requires no special attention.

The Value Returned can be one of these results:

- **NORMAL** for a normal WTOR. It is displayed in green.
- **UNUSUAL** for an unusual WTOR. It is displayed in yellow.
- **IMPORTANT** requires attention. It is displayed in pink and has high priority.
- **CRITICAL** requires action. It is displayed in red and has priority of 50.

Another keyword is returned for the code match. If this keyword is set to PRI, then the WTOR is the one that is replied to if a reply is specified for shutdown. This option is useful if an Application can issue more than one WTOR at any time. If no specification is made, the WTOR most recently

issued is assumed to be the one to reply to. If all of the CODE fields defined match those from the command list, the value in the **Value Returned** field is returned to the command list to be used for decision-making.

These definitions are the ones that display the NetView WTOR on the SDF panels. In a previous example, DS1802A was defined as an expected WTOR. It was displayed in green on the SDF panel. If this definition is omitted, all WTORs are displayed in yellow by default.

You can also add the WTORS specification to the MESSAGES/USER DATA policy item for the MVS COMPONENT policy object. In this case, you can use the **Code 2** field to specify a job name mask so that the specification only relates to certain jobs.

# Student exercises

## Unit 7

### 1. Status Display Facility

- Review the current SDF panels
- Copy the supplied samples
- Dynamically implement the sample SDF members
- Customize the enterprise-level SDF panel
- Test your SDF customization

### 2. Centralized operations exercises

- Define gateways, focal points, and operators
- Add MVSB to the Saplex
- Update MVSB system policy
- Verify gateway connectivity
- Verify SDF panels
- Verify multisystem operation



Perform the exercises for this unit.

## Summary

---

Now that you have completed this unit, you should be able to perform the following tasks:

- Define Gateway sessions and autotasks
- Describe how the Status Display Facility (SDF) collects status information
- Define SDF tree structures
- Create SDF panel definitions
- Customize SDF definitions in DSIPARM

---

© Copyright IBM Corporation 2019

---

*Summary*

Now that you have completed this unit, you should be able to perform the following tasks:

- Define Gateway sessions and autotasks
- Describe how the Status Display Facility (SDF) collects status information
- Define SDF tree structures
- Create SDF panel definitions
- Customize SDF definitions in DSIPARM



## 8 Defining Application Groups

---



### Unit 8 Defining Application Groups



© Copyright IBM Corporation 2019  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

---

**What this unit is about**

This unit provides in-depth information about Application Groups. This information includes the types of Application Groups, ways to define them in the automation policy, and ways to manage them within your enterprise.

**How you check your progress**

You can check your progress in the lab exercises.

**References**

*SC34-2717 System Automation for z/OS Version 4*

*Release 1 Defining Automation Policy*

*[https://www.ibm.com/support/knowledgecenter/SSWRC\\_J\\_4.1.0/com.ibm.safos.doc\\_4.1/manuals.html](https://www.ibm.com/support/knowledgecenter/SSWRC_J_4.1.0/com.ibm.safos.doc_4.1/manuals.html)*

# Objectives

---

After completing this unit, you should be able to:

- Describe application group scenarios
- Use the Customization Dialog to define the following Application Groups:
  - Sysplex
  - System
  - Basic
  - Server
  - Move

## *Objectives*

After completing this unit, you should be able to:

- Describe application group scenarios
- Use the customization dialog to define the following Application Groups:
  - Sysplex
  - System
  - Basic
  - Server
  - Move

# Lesson 1. Overview of Application Groups

## Lesson 1: Overview of application groups

- Why application groups?
- Application group type and nature
- Characteristics of an application group
- System application group
- Sysplex application group
- Group behavior: Active versus passive
- Preference values
- Basic steps to define a group

### What this lesson is about

This lesson provides an overview of System Automation application groups.

### Objectives

After completing this lesson, you should be able to describe the various types of application groups and their behaviors and natures.

### References

*SC34-2717 System Automation for z/OS Version 4  
Release 1 Defining Automation Policy*

You can use SA z/OS to group resources, such as Applications, into Application Groups. The process can be applied iteratively, such that an Application Group can be a member of another Application Group.

There are two types of Application Groups:

- A *System Application Group* is a group of resources within the same system.
- A *Sysplex Application Group* is a group of resources on different systems contained within the sysplex. Typically, these are the Applications that can be moved across systems in the sysplex.

Individual Applications can be members of several Application Groups.

# Why application groups?

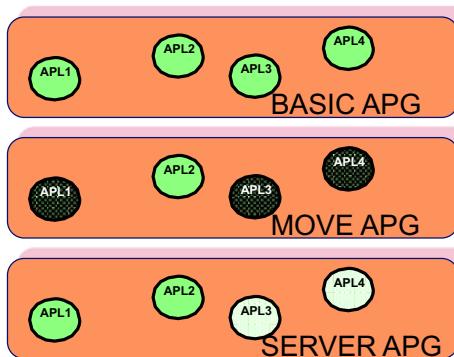
- Applications can not be linked to a system or SAplex directly, they must be linked to an application group (APG)
  - Simplifies linking, just link a group vs. many applications
  - Application group can be invisible to operators (linking only)
- Design common and system specific application groups
- Design common and SAplex specific application groups
- Can be built to match business functions, cluster and enterprise-wide
- Aggregated APG state derived from the statuses of its members and nature
- Active APGs are a manageable resource for goal driven automation
  - Can be source or target of dependencies which simplifies definition
  - Availability and suspend requests are propagated to members
- Server APG has variable availability and satisfactory targets to adjust number of members
- Move APG has an availability target of 1 and allows “moving” members
- Home and backup systems can be controlled by preference values, status, and recovery mode

## Why application groups?

Application groups have multiple benefits and usage scenarios:

- Applications can not be linked to a system or SAplex directly, they must be linked to an application group (APG). This simplifies linking, just link a group vs. many applications. An application group can be invisible to operators (linking only)
- It is recommended to design common and system specific application groups, same for SAplex.
- Application groups can be built to match business functions, cluster and enterprise-wide
- Aggregated APG state is derived from the statuses of its members and nature
- Active APGs are a manageable resource for goal driven automation
  - Can be source or target of dependencies which simplifies definition
  - Availability and suspend requests are propagated to members
- Server APG has variable availability and satisfactory targets to adjust number of members
- Move APG has an availability target of 1 and allows “moving” members
- Home and backup systems can be controlled by preference values, status, and recovery mode

# Application group type and nature



- Type is either system or sysplex
- Three ‘natures’ of groups
  - BASIC: The group is available when all of its resource members are available
  - MOVE: The group is available when one resource member is available
  - SERVER: The group is available when one or more resource members are available
- Groups can be active or passive

© Copyright IBM Corporation 2019

8-5

## Group type and nature

There are two types of application groups:

- **System application groups:** All members must run on a single system.
- **Sysplex application groups:** Members can run on any system in the SAplex that an automation manager controls. Typically, the members are applications which can be moved across systems in the SAplex.

Within a System Automation for z/OS managed SAplex, a named system application group can be defined on multiple systems in the SAplex. The names of sysplex application groups must be unique within the SAplex.

The **nature of an application group** describes its availability behavior:

- How it ensures that the correct number of applications are started to satisfy the requirements of the group
- How applications are selected to start or stop
- How the status of the group is determined from the status of its members

**BASIC:** The group is available when all of its resource members are available.

**MOVE:** The group is available when one resource member is available.

**SERVER:** The group is available when one or more resource members are available.

**Groups can be active or passive.** A passive group does not propagate votes to its members.

## Characteristics of an application group

- The automation manager controls application groups
- They can have these items:
  - Dependency links
  - Runtokens
  - Service periods
  - Triggers
  - Monitors through HasMonitor relationship
- They can be linked to multiple systems or SAplices
- They are System Automation for z/OS resources
  - Resource type is application group; designator APG
  - Status is determined, based on group-type, goal achievement, and the status of members
  - Can be monitored, started, or stopped as a group

Applications can be linked to more than one application group

© Copyright IBM Corporation 2019

6

### Characteristics of an application group

Application groups have many of the characteristics that are associated with applications:

- Resource dependency relationships
- Links to schedules (Group members do not inherit schedule values.)
- Runtokens

Like applications, you can enter start and stop requests for application groups. The start or stop of a group is a propagation of requests to its members. Members of a qualified application group inherit the runmode qualification of the group. A trigger does not prevent the start or stop of a group. A trigger is checked when the manager sends an order to a group member. Members can inherit triggers from the application group.

Application groups can be linked to multiple systems or SAplices in an enterprise. Application groups that are linked to multiple systems or SAplices are logically different; and different automation managers control them. Similar to an application, each application group has a value for each of the automation manager statuses. These values are derived from the status of the members of the group. Additionally, server group status is correlated against policy definitions for the *availability target* and *satisfactory target*. Applications can be linked to multiple groups;

however, use caution when taking advantage of this capability. For better performance, define service periods and triggers against an application group instead of its members.

## System application group

- Contains members from the same system
  - Applications
  - Other system application groups
    - Same group can be defined on multiple systems
    - Cannot contain a sysplex application group or monitor resource
- Uses naming convention  
`Automation_name/APG/system_name`
  - An example is `AIRLINE_RES/APG/SYSA`

© Copyright IBM Corporation 2019

8-7

### *System application group*

All members of system application groups must be on a single system with the exception of the resource a resource reference is referencing to.

Members are:

- Applications
- System application groups (nesting)
- Resource references

System Application Groups can also contain nested System Application Groups. Because a System Application Group is confined to a single system, it cannot contain a nested Sysplex Application Group. The resource name is displayed in the operator panels (INGGROUP) and in the customization dialog panels. The policy entry name, `AIRLINE_RES` in this example, must be unique within the PDB. In many cases, the automation name (defined in the APPLGROUP INFO policy item for the APG) is the same as the entry name. The default setting for the automation name is the entry name. The `automation_name`, `AIRLINE_RES`, does not have to be unique. Because this is a System Application Group, you can define the same automation name on multiple systems. When the group name is qualified with a system name, `AIRLINE_RES/APG/SYSA`, for example, it forms a unique representation of the group.

. The following resources are generated when the members are linked to the APG:

- *automation\_name/APG/system\_name*
- *member\_name/APL/system\_name*

## Sysplex application group

- Contains members from any system within the SAplex
  - Applications on multiple systems
  - Other system application groups
  - Other sysplex application groups
- Can be linked to more than one SAplex
- Uses naming convention Automation\_name/APG
  - An example is AIRLINE\_RES/APG
  - Does not require the system name because it is not linked to a particular system
  - Name must be unique within the sysplex

### Sysplex application group

A sysplex application group contains members from multiple systems within the sysplex. The members can be Applications, System application groups, or nested sysplex application groups. Again, notice the naming convention. It is displayed in both the operator panels, for instance, INGGROUP, and the customization dialog. AIRLINE\_RES/APG refers to the AIRLINE\_RES sysplex application group. Because this is a sysplex application group, you can define the same automation name on multiple SAplexes.

The following resources are generated when the members are linked to the APG:

- *automation\_name/APG/[system\_name]* The third part shown in brackets is not created for SYSPLEX APGs.
- *member\_name/APL/system\_name*

## Notes on the automation name

- An application group without an automation name is supported; however, it is not automated
  - No monitoring
  - No status
- No APG resource is created
- Initially, the default setting for this field is the entry name

© Copyright IBM Corporation 2019

8-9

### *Notes on the automation name*

An application group without an automation name is supported; however, it is not automated

- No monitoring
- No status

No APG resource is created. Initially, the default setting for this field is the entry name.

## Group behavior: Active versus passive

- Groups can be active or passive
- Active group (default)
  - Votes are propagated to its members; status of members can be changed
  - Supports monitoring and automation
- Passive group
  - No votes are propagated to its members  
Preference Value = 0
  - Can be used by relationships
  - Passive groups are monitored only

### Group behavior: Active versus passive

No votes are propagated to any member of a *Passive* group. The group must be controlled manually outside of the normal SA z/OS processes. The default behavior is *Active*. Most groups are Active groups. Their votes are propagated to their members, thereby supporting monitoring and automation of the group and its members.

## Server group

- Availability target (AVT): Desired number of available members
- Satisfactory target (SAT): Minimum number of members that must be active for the group to have a satisfactory status.  
Less than or equal to the AVT
- Preference value: Priority value that is used to identify which members to start to reach Availability Target
- Automation manager uses algorithm to decide which members should be started to meet the Availability Target:  
Effective Preference Value (EPV) = Base Preference Value (BPV) defined in the customization dialog + Adjusted Preference Value (APV) set by an operator using the INGGROUP command + bonus value set by the Automation Manager
- Group mode: Normal or recovery

### Server group

Several factors are used to determine how a server group is managed:

- **Availability Target:** The number of server group members that must be active for the group to be considered available. The Availability Target (AVT) is defined when you define the group in the customization dialog. The AVT can also be overridden by an operator, by using the INGGROUP command. The AVT can also be a negative number, which is called *relative numbering*. Using a negative AVT means that all but a number of selected Resources must be available before the current APG-derived Resource is considered available.
- **Satisfactory Target:** The minimum number of server group members that must be active for the automation manager to consider the group Compound Status as SATISFACTORY. The Satisfactory Target is also specified in the customization dialog and can be overridden using the INGGROUP command.
- **Preference Value:** A number assigned to the group member that identifies whether it should be started or stopped to meet the Availability Target. Preference Values can also be used to define regular and backup members. The Effective Preference Value is determined by adding a *bonus* value from the automation manager plus any adjustments made by the operator to the Base Preference Value defined in the customization dialog.

- **Group Mode:** A group is in *Normal mode* when it meets its Availability Target with only regular members active. A group is in *Recovery mode* if backup members are started to meet the Availability Target, because a regular member failed. Backup members are typically identified with a Base Preference Value less than 601.

## Availability Targets and Satisfactory Targets

The AVT can range from zero to all members of a server group. The AVT is measured by members and not by Applications. If a group has a nested group as a member, the availability of the nested group counts as only one available member of the main group, even though the nested group might have many members defined to it.

The Satisfactory Target can be null, meaning that no specific Satisfactory Target is defined for the server group. Both Availability Target and Satisfactory Target can be specified in a *delta notation* format. For example, the Availability Target can be equal to -1 (minus 1), which means that all members within the server group except one must be available. When the Satisfactory Target is equal to -1 (minus 1), then the server group is in a satisfactory state when the number of available members is greater than or equal to AVT-1.

## Preference Values

A member Effective Preference Value (EPV) is determined by the sum of the three individual Preference Values (base, adjusted, and AM bonus). First, your system administrator defines a Base Preference Value (BPV), such as 700. Then, to meet an Availability Target, an operator might adjust the BPV by defining an Adjusted Preference Value (APV), for example, 200, which results in a value of 900. If the Application is selected to meet the Satisfactory Target, the automation manager adds a bonus value of 250 to the resulting value of 900. An Effective Preference Value (EPV) of 1150 has the following results:

$$\text{EPV} = \text{BPV} + \text{APV} + \text{AM bonus}$$

$$\text{EPV} = 700 + 200 + 250 = 1150$$



**Note:** The AM bonus of +250 is a result of two bonus values from the manager.

When the automation manager attempts to start an Application to meet the Availability Target, it chooses one with the highest resulting Preference Value. In this example, 700 for a BPV plus 200 for an APV yields a resulting Preference Value of 900. If that Preference Value is the highest, then the automation manager issues a MakeAvailable vote for that member to start it.

## Preference values and their effect on group policy

| Value | Description  |
|-------|--|
| 3100  | The resource is always selected regardless of the state of the resource.   |
| 2900  | The resource is always selected unless the observed status of the resource is SYSGONE, or its system is stopping and there is a viable alternative resource. It will still be selected if its system has been excluded or it has an observed status of HardDown. |
| 2400  | The resource is always selected unless the observed status of the resource is SYSGONE or HARDDOWN, or its system is stopping and there is a viable alternative resource. It will still be selected if its system has been excluded.                              |
| 1000  | The resource is always started, other resources <= 700 are stopped.  |
| 900   | The resource is started when the group is started. Members with preference 500 will be stopped in favor of this resource.  |
| 700   | The resource is started when the group is started and is not stopped unless a higher preference (1000) alternative becomes available.  |
| <600  | (Used as a threshold.) The resource with the preference value of 599 and lower is not selected when the group starts. The resource is selected only if the group is in recovery mode.  |
| 500   | Like 100, however it will be stopped when an alternative with a calculated preference >750 is available.   |
| 300   | Like 100, however it will be stopped when an alternative with a calculated preference >550 is available.   |
| 100   | The resource is selected only when the group enters recovery mode. Stopped as soon as better alternative (for example, preference 500 and greater) becomes available.  |
| 1     | The member is always deselected and always has a MakeUnavailable vote that is propagated to it.  |
| 0     | The member is passive and never receives any vote that is propagated by the group. It cannot be started or stopped.  |

© Copyright IBM Corporation 2019

8-12

### Preference values and their effect on group policy

This table summarizes the meaning of several preference values. The preference value affects the role of a resource. The maximum preference value is 3200. Any value greater than 3200 is changed to the maximum.

In addition to the above table, if a resource has an effective preference value that is greater than 2600, it keeps its effective preference value. If the system is SysGone or HardDown and the preference is between 2000 and 2599 then the effective preference value is set to 1 if there is a viable alternative.

Primary group members are members with a base preference value of 600 or more. Backup group members are members with a base preference value of 2 - 599 and are selected only if the group is in *recovery mode*.

Base preference values of zero and one are special cases. A base preference value of one means that the member is always deselected, and always receives a MakeUnavailable vote. This value can be useful if complete manual control of a move process is required.

A base preference value of zero means that the member is a passive member and never receives a vote. If the preference value of a member is changed and the member becomes passive, all votes to the member are withdrawn.

For greater detail, see *IBM System Automation for z/OS: Defining Automation Policy*. See also “Controlling Application Move with Preference Values” in *IBM System Automation for z/OS User’s Guide*.

## Automation manager bonus values

| Value | Description  |
|-------|--|
| + 225 | The member is AVAILABLE or WASAVAILABLE  |
| + 220 | The member is STARTING, STOPPING or DEGRADED                                     |
| + 175 | The sticky bonus is on and the adjusted preference is greater than 1000          |
| + 25  | The sticky bonus is on and the adjusted preference is less than or equal to 1000 |
| - 400 | The member is SYSGONE and the adjusted preference is greater than 1500           |

- The *sticky bonus* favors members in the group that were active last
- More than one bonus value can be assigned

### Automation manager bonus values

This table summarizes the bonus values that the automation manager assigns. The purpose of the sticky bonus is to give a higher priority to keeping or restarting a member on a system where it ran before. The sticky bonus is added when a group member is selected. It stays on until another member is selected.

In most cases, the automation manager adds more than one bonus. For example, if an application is available or was available, the manager adds 225 points. Then based on the original base preference value, the manager might also add another 25 or 175 points as the sticky bonus.

Basic group members have a default base preference value of 700 assigned to them. The operator cannot adjust the value. Therefore, the adjusted preference value is always zero. Preference values apply only to move groups and server groups.

If a basic group member is available, the automation manager adds 225 to make it 925:

$$\text{Effective preference value (925)} = \text{base preference value (700)} + \text{adjusted preference value (0)} + \text{automation manager bonus (225)}$$

If a server group member has an adjusted preference value of 2800, the automation manager assigns it a bonus value of +175. This sticky bonus is added to the base preference value and any adjusted preference value, for a total of 2975. For adjusted preference values of less than 1000, the sticky bonus is +25.

## Group mode: Normal versus recovery

- Normal
  - All selected members have preference values of 600 or higher
  - When started, activates only members with preference values 600 or higher
- Recovery
  - An active member or its system fails
  - At least one selected member has a preference value less than 600 but greater than one
  - When all started applications have a preference value 600 or higher, the group leaves recovery mode

A group is in Normal mode when all of its active members have a preference value greater than 600. When a group is started, it selects only members with preference values greater than 600.

If an active member or its system fails, the group enters recovery mode. If at least one selected member has a preference value of less than 600, the group is in recovery mode. When all started members have preference values > 600, the group leaves recovery mode.

## Basic steps to define a group

- Create application group
- Link application group to systems or sysplexes
- Link applications to application group
- Define group behavior
- Modify resource links
  - Delete existing resource links
  - Link further resources (application group resources)
- Customize preferences (for server and move groups):
  - Availability target
  - Satisfactory target
  - Base preference values

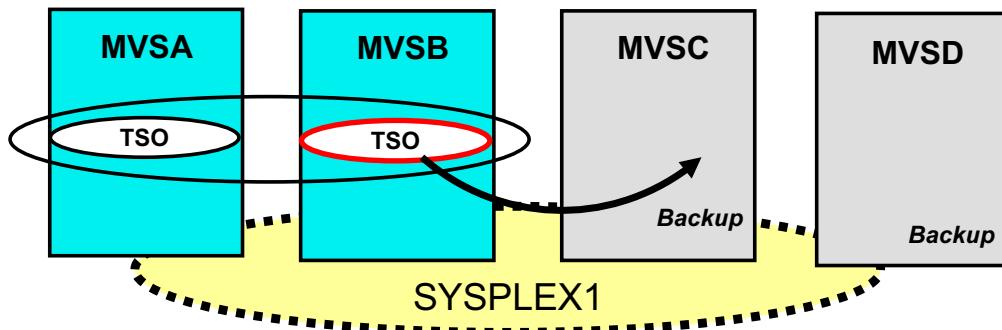
### Basic steps to define a group

On this slide are the steps that you use to define Application Groups. In step 1, the Application Groups are created. In step 2, the Application Groups are then linked to their systems (for System Application Groups) or Sysplex Groups (for Sysplex Application Groups). The customization dialog automatically create internal resources that represent each Application in each relevant system and each Application Group. In step 3, the Applications for each group are SELECTed. In step 4, you can define information about whether or not the group is active (default) or passive. In step 5, you can delete some of these automatic links. For example, some Applications in a Sysplex Group might not be valid for all systems in the Sysplex Group. You can also add other resources into the group definition; for example, other Application Groups.

Also, for Server and Move Groups, you can change the Preference Values for each member on each system or each sysplex. For server groups, you can change the Availability and Satisfactory Target preferences. Because Application Groups are resources, you can also define other policies, such as relationships.

# Lesson 2. Define a sysplex server group

## Lesson 2: Define a sysplex server group



### Example:

- Application group TSOGROUP with one application (TSO) on all systems
- Members typically run on system 1 or 2 because of higher base preference value
- If one of the members fails, start a backup on system 3 or 4 with *higher* preference for system 3

© Copyright IBM Corporation 2019

8-16

### What this lesson is about

this lesson describes how define and use sysplex server group.

### Objectives

After completing this lesson, you should be able to define and use sysplex server group.

### References

*SC34-2717 System Automation for z/OS Version 4 Release 1 Defining Automation Policy*  
*SC34-2718 System Automation for z/OS Version 4 Release 1 User's Guide*

This example defines a sysplex server group for TSO, called TSOGROUP. The group members are the TSO Application for all four systems. This example assumes that the TSO Application is already

defined. For example, use option **2** (GRP) in the customization dialog to define the SAplex. The sysplex group used in this example is SYSPLEX1.

## Step 1: Create the server group

```

AOFGDYN3                               Define New Entry
Command ==> _____
Define new entry of type ApplicationGroup
Entry name . . . . . : TSOGROUP
Type . . . . . : SYSPLEX (SYSTEM SYSPLEX)
Nature . . . . . : SERVER (BASIC MOVE SERVER)
Default Preference . . . . . : *DEF (0 to 3200, *DEF)
Automation Name . . . . . : TSOGROUP
Automatically link . . . . . : YES (for Application-Resources)
Behavior . . . . . : ACTIVE (ACTIVE PASSIVE)
Prepare Move . . . . . : _____ (YES NO)
Move Mode . . . . . : _____ (PARALLEL SERIAL)

Short Description . . . : TSO susplex server group
Long Description 1 . . . : _____
Long Description 2 . . . : _____
Long Description 3 . . . : _____
Long Description 4 . . . : _____
Long Description 5 . . . : _____

```

## Define TSOGROUP as a sysplex Server Group

Press PF3 to create

Note: After you define it, you cannot change the group type unless you delete and redefine the group.

© Copyright IBM Corporation 2019

8-17

### **Step 1: Create the server group**

The first step is to create the group definition. Select the APG policy object (Application Groups, option 5) from the Entry Type Selection panel. Use the **NEW** command to define a new group, in this case a server group.

The **Entry name** field is the name used within the customization dialog to manage this definition.

The **Automation Name** is the name used for the server group that is visible to operators. It is typical, but not necessary, to use the same name. The entry name can be up to 20 characters, and the automation name can be up to 11 characters.

The group nature can be Basic, Move, or Server. This example is defining a sysplex server group. The default Preference Value (\*DEF) is used initially. The default Preference Value is assigned to all resources for this group when they are automatically created. The default value is 700.

Therefore, all members are eligible to be started when the group status is NORMAL. This setting is also known as the ***Base Preference Value***.

You can also specify whether any changes to the group composition automatically causes changes to the definition of the server group. If you choose **NO** for this specification and add a new system to the group specification, it does not create new resources that are part of the server group.

You can also indicate whether the group is Active or Passive. A passive group does not create votes for its members, although it does maintain status. Typically, groups are defined as active. When this specification is done, press PF3.

## TSOGROUP policy panel

| AOFGEPOL<br>Command ==>                                | Policy Selection                                    | Entry created<br>Scroll ==> CSR          |
|--|---|--|
| Entry Type : ApplicationGroup<br>Entry Name : TSOGROUP | PolicyDB Name : NEW_PDB<br>Enterprise Name : SAPLEX |  |
| Action   | Policy Name   | Policy Description                       |
| DESCRIPTION  | Enter description                                   | Define applicationgroup information      |
| APPLGROUP INFO   | Select applications for sysplex APG                 | Select trigger                           |
| APPLICATIONS   | Select service period                               | Define relationships                     |
| TRIGGER  | Select resources and set preferences                | Define messages and user data            |
| SERVICE PERIOD   | List resources generated for this entry             |  |
| RELATIONSHIPS  | List resources where this entry is a member         |  |
| MESSAGES/USER DATA                                     |   |  |
|  | -----RESOURCES-----                                 |  |
| RESOURCES  |   |  |
| GENERATED RESOURCES                                    |   |  |
| MEMBER OF  |   |  |
|  | WHERE USED  | List sysplex groups linked to this entry |
|  | COPY  | Copy data from existing entry            |

Remaining tasks edit the Application Group policy for TSOGROUP:  
*WHERE USED, APPLICATIONS, RESOURCES, and APPLGROUP INFO*

### TSOGROUP policy panel

On this slide is the Application Group Policy Selection panel for the sysplex server group, TSOGROUP. On the next several slides, you see how to define policy definitions for these policies:

- WHERE USED
- APPLICATIONS
- RESOURCES
- APPLGROUP INFOrmation

## Step 2: Link group to SAplex

| AOFGXWHU         |            | Where Used                       | Row 1 to 1 of 1 |
|------------------|------------|----------------------------------|-----------------|
| Command ==>      |            |                                  | SCROLL==> CSR   |
| Entry Type       | Entry Name | PolicyDB Name<br>Enterprise Name |                 |
| ApplicationGroup | TSOGROUP   | NEW_PDB<br>SAPLEX                |                 |
| Action           | Status     | Name                             | Type            |
|                  | SELECTED   | SYSPLEX1                         | GRP             |

**WHERE USED** policy: Link TSOGROUP to the SYSPLEX1 sysplex group

### Step 2: Link group to SAplex

Select the WHERE USED policy item to link this sysplex server group to the SYSPLEX1 sysplex group. In this case, there is only one SAplex defined. You must type an **S** next to the SAplex and press Enter to link this group to the SAplex. The Status column is updated to display SELECTED. Press PF3 when done. You return to the Application Group Policy Selection panel.

## Step 3: Link application to group

| AOFGXC4E    |                    | Applications for ApplicationGroup | Row 76 to 80 of 80            |
|-------------|--------------------|-----------------------------------|-------------------------------|
| Command ==> |                    |                                   | SCROLL==> <a href="#">CSR</a> |
| Entry Type  | : ApplicationGroup | PolicyDB Name                     | : NEW_PDB                     |
| Entry Name  | : TSOGROUP         | Enterprise Name                   | : SAPLEX                      |
| Action      | Status             | Application                       |                               |
|             | SELECTED           | TSO                               |                               |
|             |                    | VLF                               |                               |
|             |                    | VTAM                              |                               |
|             |                    | WEBSRV                            |                               |
|             |                    | ZFS                               |                               |

**APPLICATIONS** policy: Defines members of TSOGROUP; in this case, only one member, the TSO Application

### Step 3: Link application to group

Use the APPLICATIONS policy item to select the Applications that are part of this group. In this case, only TSO is selected. Press PF3 when you are done. Resources for TSO are automatically created for all systems that are part of the sysplex group, SYSPLEX1.

At this point, you have defined a TSO APL resource in each of the four systems that are members of the SYSPLEX1 group, and each is a part of the server group. Each of these resources has a Base Preference Value of 700, the default. The Base Preference Values are defined by using the RESOURCES policy item shown in Step 5: Set availability target and preferences.

## Step 4: Define owner and behavior

| Application Group Information  |  |
|--|--|
| A0FGAPAO   | Command ==> _____                            |
| Entry Type : ApplicationGroup  | PolicyDB Name : NEW_PDB                      |
| Entry Name : TSOGROUP  | Enterprise Name : SAPLEX                     |
| The following field was specified when the Application Group was defined and cannot be modified: |  |
| Application Group Type   | : SYSPLEX                                    |
| Nature . . . . .   | <u>SERVER</u> (BASIC MOVE SERVER)            |
| Default Preference . . . . .   | <u>*DEF</u> (0 to 3200, *DEF)                |
| Automation Name . . . . .  | <u>TSOGROUP</u>                              |
| Behavior . . . . .   | <u>ACTIVE</u> (ACTIVE PASSIVE)               |
| Desired Available . . . . .  | _____ (ALWAYS ONDEMAND ASIS)                 |
| Monitor for IPL complete . . . . .   | _____ (YES NO)                               |
| Prepare Move . . . . .   | <u>YES</u> (YES NO)                          |
| Move Mode . . . . .  | _____ (PARALLEL SERIAL)                      |
| Status Determination . . . . .   | _____ (CSONLY)                               |
| Inform List . . . . .  | _____ (SDF EIF E2E IOM ITM SMF TTT USR NONE) |
| Owner . . . . .  | <u>J. Doe</u>                                |
| Info Link . . . . .  | _____  |
| Automatically link APL-Resources into APG . . . . .  | <u>YES</u> (YES NO)                          |

**APPLGROUP INFO** policy: Define Desired Available, Move Mode, Prepare Move, and Owner policies

### Step 4: Define owner and behavior

The APPLGROUP INFO policy item for TSOGROUP is shown on the slide. *Behavior* is either PASSIVE (monitored only) or ACTIVE (default, monitored and automated). The *Desired Available* field defines the default desired status for the Application group. A value of ALWAYS indicates that the group is started when possible. A value of ONDEMAND indicates that the group is started manually by an operator, from a Service Period, or a propagated vote from a group.

The *Move Mode* field defines when new group members are started in relation to the stopping of existing group members. A value of PARALLEL indicates that new members are started as old members are stopping. A value of SERIAL indicates that new members are started after the old members stop.

*Status Determination* specifies how the compound status of the application group is determined. Using CSONLY indicates that 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 compound status of the application group is determined from several statuses, like observed status, desired status, automation status, or startability status.

The **Inform List** field defines the components to receive status updates for this resource. In this case, no entry is defined. The system administrator might define a default Inform List at the system of sysplex level in the Sysplex Defaults policy, for example. The **Owner** field specifies information for the operator, such as whom to contact in case of an error. The **Info Link** field can be used to specify an URL where additional information about the location of the Application Group.

## Step 5: Set availability target and preferences

| AOFGARS2                               |                                     | Select Resources |            | Row 1 to 4 of 4<br>SCROLL==> CSR |
|--|-------------------------------------|------------------|------------|----------------------------------|
| Command ==> _____                      |                                     |                  |            |                                  |
| Entry Type : ApplicationGroup          | Entry Name : TSOGROUP               |                  |            |                                  |
| Sysplex : SYSPLEX1                     |                                     |                  |            |                                  |
| Number of selected Resources : 4       |                                     |                  |            |                                  |
| Availability Target. . . . . : 2       | (-3 to 4 , *ALL)                    |                  |            |                                  |
| Satisfactory Target. . . . . : 1       | (Level at which group is available) |                  |            |                                  |
| Show all Resources . . . . . : NO      | (YES NO)                            |                  |            |                                  |
| Show only Resources with String. _____ |                                     |                  |            |                                  |
| Action                                 | Preference                          | Resource Name    | Entry Name |                                  |
|  | 700                                 | TSO/APL/MVSA     | TSO        |                                  |
|  | 700                                 | TSO/APL/MVSB     | TSO        |                                  |
|  | 600                                 | TSO/APL/MVSC     | TSO        |                                  |
|  | 300                                 | TSO/APL/MVSD     | TSO        |                                  |

### RESOURCES policy:

- Members are preselected with default preference of 700 (\*DEF in step 1)
- MVSC is set to 600 so that it is not selected at group startup
- MVSD is a backup and is only used for *Recovery Mode*. Its preference is now 300
- AVT is two members. SAT is one member

### Step 5: Set availability target and preferences

Select the RESOURCES policy item. You must set the Availability Target (AVT) for the group, the Satisfactory Target for the group, and the Base Preference Values for each member. You can set the AVT either as an absolute number (such as 2) or as a number that is relative to the current number of selected resources (such as minus 2 [-2]). You can also specify \*ALL for the AVT target, which means that SA z/OS tries to make all members available. The current APG derived resource is considered as available only when all members are available.

The Satisfactory Target specifies the number of members in the group that must be active before the automation manager considers the group to be in a satisfactory state. The value can be equal to or less than the AVT value, but it cannot be more. It can also be relative to the AVT value, such as minus 1 (-1).

The value for both the Availability Target and Satisfactory Target can be negative, meaning that the value is relative to the number of group members. This method of defining your Application Groups is beneficial, because you might not have to change these values when you add a new member to the group. Initially, each selected resource is shown in a list with their Base Preference Value. You can then alter the composition of the group and the Preference Values of individual resources.

You can add or delete other resources by specifying filters:

- *Show all Resources* (YES or NO)
- *Show only resources with String*

You can use these fields to see additional resources, which can then be added into the group. Add a resource by typing a valid Preference Value against it, or use the command P (primary) to set a value of 700 or D (secondary) to set a value of 400.

Each resource initially has the default Base Preference Value from the earlier panel, 700 in this case. You can alter the value of any or all at this time. Because two specific Applications are to be preferred, the TSO Applications for systems 1 and 2 are left with values of 700. Because TSO on MVSD runs only when the group is in Recovery Mode, its value is set to 300. TSO on MVSD starts only if the group goes to Recovery Mode. Use the row command M to remove a preference value. Press PF3 to save these definitions.

The earlier setting for *Automatically Link Application Resources* is still YES. If another system, MVSE, is added to the Sysplex Group, a new resource (TSO/APL/MVSE) is automatically added to the group with the default priority. Its selected population rises by 1. If this is not what you want, and TSO is to run only on the four selected systems (regardless of any change in the sysplex composition) set the *Automatically Link* value to NO.

INGROUP for TSO server group

**INGKYGRA** SA z/OS - Command Dialogs  
 Domain Id . : AOFDA INGGROUP Date . . . : 03/25/19  
 Operator Id : INGC102 Sysplex = SYSPLEX1 Time . . . : 21:21:08

Specify or revise the following data:

Target => \_\_\_\_\_ System name, domain ID or sysplex name

Action => \_\_\_\_\_ EXCLUDE-AVOID-INCLUDE or ACTIVATE-PACIFY or  
 ADJUST or RESET DEFAULT or OVERRIDES, POLICY or  
 RECYCLE-CANCEL

Opt.Parms => \_\_\_\_\_

Group(s) => **TSOGROUP/APG**

**INGKYGRB** SA z/OS - Command Dialogs Line 1 of 4  
 Domain Id . : AOFDA INGGROUP Date . . . : 03/25/19  
 Operator Id : INGC102 Sysplex = SYSPLEX1 Time . . . : 21:22:49

Group: TSOGROUP/APG Nature: Server Passive: NO Suspend:  
 Description: TSO sysplex server group  
 Excluded :  
 Avoided :  
 Mode : Normal Availability Target: 2 Adjust: 0 Result=> 2  
 Satisfactory Target=> 1 Adjust: 0  
 Rolling Recycle: None

| Name | Type | System | Pref | Adj | Result | Avl | Eff | Stat | Act | Sus |
|------|------|--------|------|-----|--------|-----|-----|------|-----|-----|
| TSO  | APL  | MVSA   | 700  | 0   | 700    | Yes | 950 | Sel  | --  | --  |
| TSO  | APL  | MVSB   | 700  | 0   | 700    | Nst | 0   | Uns  | --  | --  |
| TSO  | APL  | MVSC   | 600  | 0   | 600    | Nst | 0   | Uns  | --  | --  |
| TSO  | APL  | MVSD   | 300  | 0   | 300    | Nst | 0   | Uns  | --  | --  |

© Copyright IBM Corporation 2019

8-23

## *INGROUP for TSO server group*

To verify that the TSO server group is created correctly, perform the following tasks:

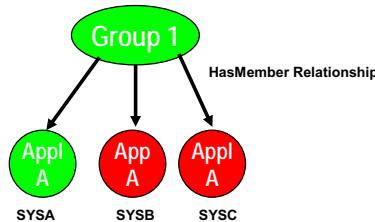
- Use INGGROUP to display the server group.
  - Use the ADJUST action to view member preference values.

This example shows INGGROUP for the TSO server group. Four instances of the TSO Application are defined. The Availability Target is 2, and the Satisfactory Target is 1. The automation manager attempts to keep two group members active to meet the AVT. The group is UNSATISFACTORY when the number of active members falls below the SAT. TSO on MVSD has a Base Preference Value of 300. It is used if one of the other members fails or if an operator changes the AVT. The operator can use the panel to adjust the satisfactory target and result values.

# Lesson 3. Groups with move nature

## Lesson 3: Groups with move nature

- Behaves like a server group with an availability target of 1
- Only one member is started
- Members are typically instances of the same application
- Active member can be changed
- The move mode can be one of these types:
  - Serial: One member is stopped before another is started
  - Parallel: Members are started and stopped simultaneously



© Copyright IBM Corporation 2019

8-24

### What this lesson is about

This lesson reviews groups with move nature

### Objectives

After completing this lesson, you should be able to implement groups with move nature.

### References

SC34-2717 System Automation for z/OS Version 4  
Release 1 Defining Automation Policy

The third group nature is a Move Group. A Move Group is typically defined when only one instance of an Application is required within a sysplex (server group with AVT=1). A good example is a web server. The automation agent on each system requires a definition of the web server Application so that it can manage the Application. All instances of the web server application are defined to a Move Group, with BPV set to indicate one regular host and possible backup hosts. Typically, Move Group members are Basic Groups. The move mode policy controls when the new member is

started: at the same time the old member is being stopped (PARALLEL), or after the old member is completely stopped (SERIAL). The default value is PARALLEL.

# Initiating application moves

- External events
  - System leaves the SAplex (SA z/OS only)
  - Application fails or has recovered
- Policy goals
  - INGGROUP command options
    - Exclude
    - Avoid
    - Include
- Manual actions (Targeted moves)
  - Manually move applications to another system using SYSPLEX APGs  
For a SYSTEM APG just another member is selected.
    - INGMOVE: Select the target member to start the application
    - INGGROUP: Adjust preference values for application group members

## Initiating application moves

You can initiate application moves in one of three ways: externally, goal driven, or targeted.

Externally initiated moves can be a reaction to the loss of a system or when a member becomes unusable. Externally initiated moves can also be triggered when a preferred member becomes usable again, *or when a system rejoins the SAplex automated by System Automation for z/OS*.

In goal driven moves, the choice of members might vary. A member can be excluded, avoided, or included when deciding which members to start. Such actions are usually part of a planned move to accommodate a scheduled outage.

Targeted moves occur when group members are chosen manually at run time. The INGGROUP command can be used to modify member preference values.

Instead of the INGGROUP command, the INGMOVE command can be used to initiate immediate or planned moves. Using the INGMOVE command is simpler than calculating the results of adjusting preference values. Application moves apply to both move groups and server groups.

## Move APG policy: Move mode

```

AOFGAPAO          Application Group Information
Command ==>                              

Entry Type : ApplicationGroup      PolicyDB Name : OPSU7EX1
Entry Name : RVXMOVE             Enterprise Name : OPSU7EX1

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

Nature. . . . . MOVE           (BASIC MOVE SERVER)
Default Preference: . . . . . *DEF   (0 to 3200, *DEF)
Automation Name . . . . . RVXMOVE
Behavior. . . . . ACTIVE        (ACTIVE PASSIVE)
Desired Available . . . . .    (ALWAYS ONDEMAND ASIS)
Monitor for IPL complete. . . .
Prepare Move. . . . .          (YES NO)
Move Mode . . . . . SERIAL     (PARALLEL SERIAL)
Status Determination. . . . .
Inform List . . . . .          (SDF EIF E2E IOM ITM SMF TTT USR NONE)
Owner
Info Link : . . . . . .
Automatically link APL-Resources
into APG. . . . . YES         (YES NO)

```

**Move mode** policy determines when votes are generated:

- PARALLEL: New member started simultaneously while old member is stopped (default)  
Faster, but not good for applications that require only one active instance at any given time
- SERIAL: New member started *after* old member is stopped completely
- Also applies to server group

© Copyright IBM Corporation 2019

8-26

Move APG policy: Move mode

Definition of the move mode policy:

The move mode policy controls when the new member is started: at the same time the old member is being stopped (PARALLEL), or after the old member is completely stopped (SERIAL). The default value is PARALLEL.

This also applies to server groups.

## Move APG policy: Prepare move=YES

```
AOFGAPAO          Application Group Information
Command ==> _____
Entry Type : ApplicationGroup      PolicyDB Name : OPSU7EX1
Entry Name : RVXMOVE               Enterprise Name : OPSU7EX1
The following field was specified when the Application Group was
defined and cannot be modified:
Application Group Type : SYSPLEX
Nature. . . . . : MOVE           (BASIC MOVE SERVER)
Default Preference. . . . . : *DEF   (0 to 3200, *DEF)
Automation Name . . . . . : RVXMOVE
Behavior. . . . . : ACTIVE        (ACTIVE PASSIVE)
Desired Available. . . . . : _____ (ALWAYS ONDEMAND ASIS)
Monitor for IPL complete. . . : _____ (YES NO)
Prepare Move. . . . . : SERIAL    → (YES NO)
Move Mode . . . . . : SERIAL     (PARALLEL SERIAL)
Status Determination. . . . . : _____ (CSONLY)
Inform List. . . . . : _____      (SDF EIF E2E IOM ITM SMF TTT USR NONE)
Owner. . . . . : _____
Info Link. . . . . : _____
Automatically link APL-Resources
into APG. . . . . : YES         (YES NO)
```

**Prepare move=YES:** Currently active group member is not stopped until its replacement member is *ready to be started*. For example, all parents are available (default)

Also applies to server group

© Copyright IBM Corporation 2019

8-27

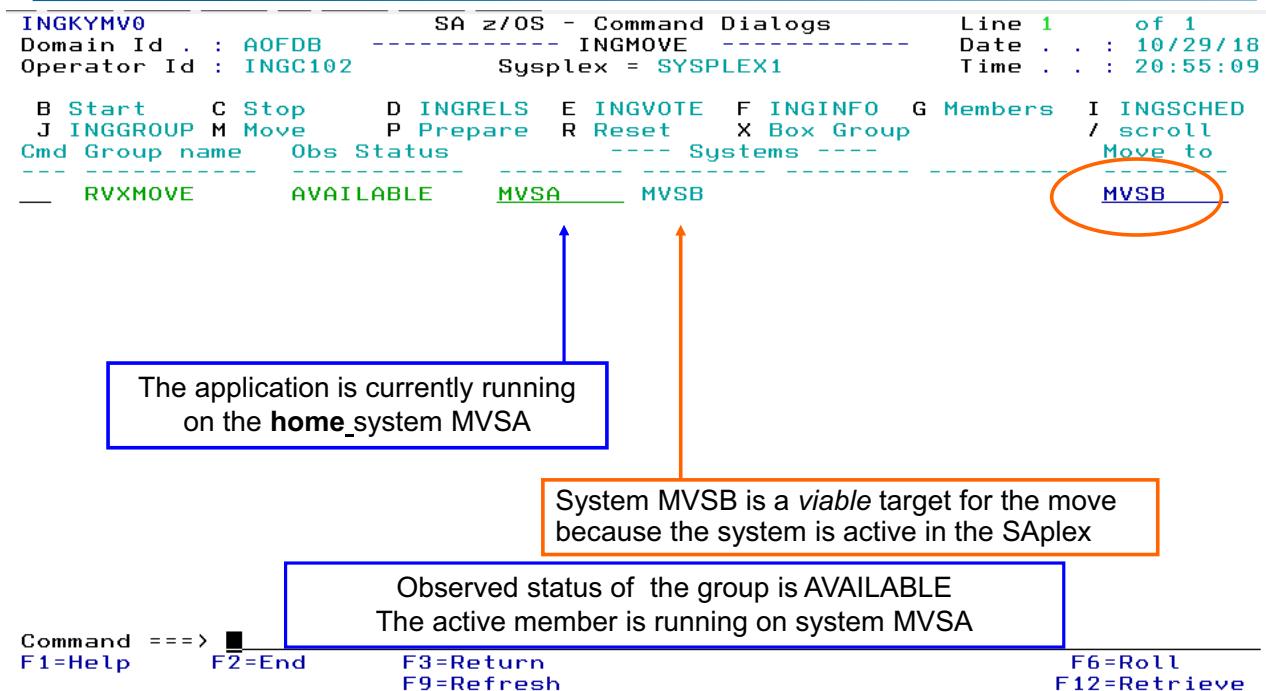
### Move APG policy: Prepare move=YES

*Prepare move=YES:* Currently active group member is not stopped until its replacement member is ready to be started. For example, all parents are available (default)

This also applies to server groups.

*Status determination* specifies how the compound status of the application group is determined. CSONLY indicates that 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 compound status of the application group is determined from several statuses, like observed status, desired status, automation status, or startability status.

## INGMOVE example in a SAplex



© Copyright IBM Corporation 2019

8-28

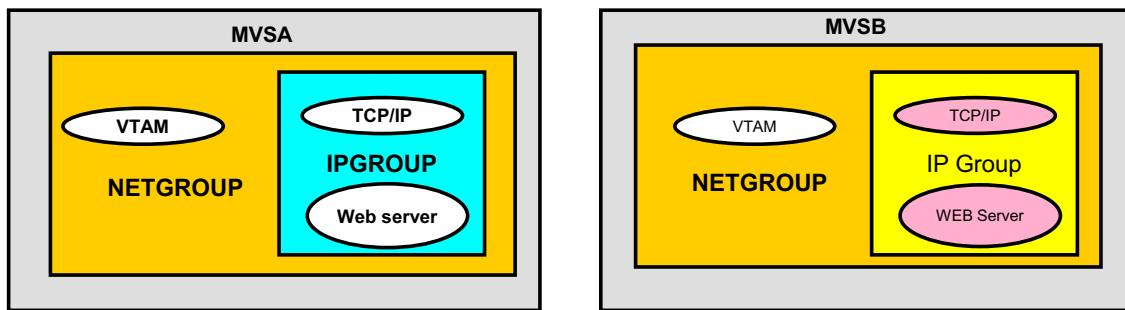
### INGMOVE example in a SAplex

This example of the **INGMOVE** command shows group, RVXMOVE with members that are linked on two systems, MVSA and MVSB. The application that is active on MVSA and MVSB is a viable target for a move of the application. The underscore () character under system MVSA indicate that MVSA is the home system for the application. The group member on MVSA has a higher base preference value than the member on MVSB. The system with the highest base preference value is the home system. More than one home system can be defined for a move group. Unless it is a home system, the system with the highest adjusted preference value is shown in reverse video. Row command **M** is used for immediate moves. The target system for the move is specified in the Move to column. For delayed moves, use row command **P**. The systems are color-coded as follows:

- Green: The application is currently running on this system.
- Turquoise: Available systems to which the application can be moved. These group members are considered viable.
- Red: The system is unavailable. A move to the system is not possible.
- Yellow: Indicates that the group member is marked not startable. A move to that system is not possible.
- Pink: The system is excluded as a target for the move.
- White: The system is avoided as a target for the move.

# Lesson 4. Define a nested Basic System Group

## Lesson 4: Define a nested Basic System Group



### Example

- Application Group NETGROUP with members VTAM and IPGROUP
- IPGROUP is nested group with members TCP/IP and Web server
- Same configuration on MVSA and MVSB systems
- MVSC and MVSD do not contain these groups

#### What this lesson is about

This lesson reviews nested basic system groups.

#### Objectives

After completing this lesson, you should be able to implement nested basic system groups.

#### References

*SC34-2717 System Automation for z/OS Version 4 Release 1 Defining Automation Policy*

This scenario has two basic system Application Groups on two of four system. The groups are used on MVSA and MVSB but not on MVSC and MVSD. The first system Application Group, IPGROUP, consists of the Web server and TCP/IP Applications. The second system Application Group, NETGROUP, has two members: VTAM and the IPGROUP Application Group. IPGROUP is a nested Application Group. Assume that IPGROUP is defined. In the next slides, you learn how to define NETGROUP.

## Step 1: Create the basic group

|  |   |
|--|---|
| <b>AOFGDYN3</b>                                  | <b>Define New Entry</b>                       |
| <b>Command ==&gt;</b>                            |   |
| <b>Define new entry of type ApplicationGroup</b> |   |
| <b>Entry name . . . . .</b>                      | <u><b>NETGROUP</b></u>                        |
| <b>Type . . . . .</b>                            | <u><b>SYSTEM</b></u> (SYSTEM SYSPLEX)         |
| <b>Nature . . . . .</b>                          | <u><b>BASIC</b></u> (BASIC MOVE SERVER)       |
| <b>Default Preference . . . . .</b>              | (0 to 3200, *DEF)                             |
| <b>Automation Name . . . . .</b>                 | <u><b>NETGROUP</b></u>                        |
| <b>Automatically link . . . . .</b>              | <u><b>YES</b></u> (for Application-Resources) |
| <b>Behavior . . . . .</b>                        | <u><b>ACTIVE</b></u> (ACTIVE PASSIVE)         |
| <b>Prepare Move . . . . .</b>                    | (YES NO)                                      |
| <b>Move Mode . . . . .</b>                       | (PARALLEL SERIAL)                             |
| <b>Short Description . . . . .</b>               | <u><b>Network: VTAM &amp; TCPIP</b></u>       |
| <b>Long Description 1 . . . . .</b>              |   |
| <b>Long Description 2 . . . . .</b>              |   |

Define NETGROUP as a basic system Application Group. Press **PF3** to create

### Step 1: Create the basic group

The first step is to create the group definition. Select the APG policy object (Application Groups, option 5) on the Entry Type Selection panel. Use the **NEW** command to define a new group, in this case a Basic Group. The default Application Group type is SYSTEM. This setting does not have to change. The default nature is BASIC and is correct for this scenario also. The default behavior is ACTIVE and is also correct for this scenario.

## NETGROUP policy panel

| AOFGEPOL                      |                          | Policy Selection                            | Entry created |
|-------------------------------|--------------------------|---|---------------|
| Command ==> _____             |                          | Scroll ==> CSR                              |               |
| Entry Type : ApplicationGroup | PolicyDB Name : NEW_PDB  |   |               |
| Entry Name : NETGROUP         | Enterprise Name : SAPLEX |   |               |
| Action                        | Policy Name              | Policy Description                          |               |
| _____                         | DESCRIPTION              | Enter description                           |               |
| _____                         | APPLGROUP INFO           | Define applicationgroup information         |               |
| →                             | APPLICATIONS             | Select applications for system APG          |               |
| _____                         | TRIGGER                  | Select trigger                              |               |
| _____                         | SERVICE PERIOD           | Select service period                       |               |
| _____                         | RELATIONSHIPS            | Define relationships                        |               |
| _____                         | MESSAGES/USER DATA       | Define messages and user data               |               |
| -----                         | -----RESOURCES-----      |   |               |
| →                             | RESOURCES                | Select resources and set preferences        |               |
| _____                         | GENERATED RESOURCES      | List resources generated for this entry     |               |
| _____                         | MEMBER OF                | List resources where this entry is a member |               |
| -----                         | -----                    |   |               |
| →                             | WHERE USED               | List systems linked to this entry           |               |
| _____                         | COPY                     | Copy data from existing entry               |               |

Remaining tasks edit the Application Group policy for NETGROUP: *WHERE USED*, *APPLICATIONS*, and *RESOURCES*

### NETGROUP policy panel

The Application Group Policy Selection panel for the system Application Group, NETGROUP is shown on the slide. On the next several slides, you see how to define policy definitions for these policies:

- WHERE USED
- APPLICATIONS
- RESOURCES

This example omits defining the APPLGROUP INFO policy item, thereby eliminating the setting of the Owner, Behavior, Inform List, and Desired Available policies for NETGROUP. Default values are used.

## Step 2: Link groups to systems

| A0FGXWHU          |                    | Where Used                       |           |
|-------------------|--------------------|----------------------------------|-----------|
| Command ==> _____ |                    | Row 1 to 4 of 4<br>SCROLL==> CSR |           |
| Entry Type        | : ApplicationGroup | PolicyDB Name                    | : NEW_PDB |
| Entry Name        | : NETGROUP         | Enterprise Name                  | : SAPLEX  |
| Action            | Status             | Name                             | Type      |
| _____             | SELECTED           | MVSA                             | SYS       |
| _____             | SELECTED           | MVSB                             | SYS       |
| _____             |                    | MVSC                             | SYS       |
| _____             |                    | MVSD                             | SYS       |

**WHERE USED** policy: Link NETGROUP to each system

### Step 2: Link groups to systems

Use the WHERE USED policy item to select which systems this system Application Group is linked to. In this example, select MVSA and MVSB by typing an **S** next to the system to link this group to the defined systems. The Status column is updated to display SELECTED. MVSC and MVSD remain blank. Press PF3 when done. You return to the Application Group Policy Selection panel. You see messages from the creation of Application Group resources for each system that is selected by the WHERE USED attribute.

## Step 3: Link Applications to group

| AOFGXC4E    |                    | Applications for ApplicationGroup |           | Row 76 to 80 of 80 |
|-------------|--------------------|-----------------------------------|-----------|--------------------|
| Command ==> |                    |                                   |           | SCROLL==> CSR      |
| Entry Type  | : ApplicationGroup | PolicyDB Name                     | : NEW_PDB |                    |
| Entry Name  | : NETGROUP         | Enterprise Name                   | : SAPLEX  |                    |
| Action      | Status             | Application                       |           |                    |
|             |                    | TSO                               |           |                    |
|             |                    | VLF                               |           |                    |
|             | SELECTED           | VTAM                              |           |                    |
|             |                    | WEBSRV                            |           |                    |
|             |                    | ZFS                               |           |                    |

### APPLICATIONS policy:

- Select VTAM as a member of the group
- IPGROUP APG is added in the next step

### Step 3: Link Applications to group

This slide illustrates how to link the VTAM Application as a member of the NETGROUP system Application Group. After you link the VTAM Application into the group and press PF3, Application resources are created for VTAM on each system.

## Step 4: Modify resource links

| A0FGARS1   | Select System                                       | Row 1 to 2 of 2<br>SCROLL==> CSR |
|--|---|----------------------------------|
| Command ==> _____                                      |   |                                  |
| Entry Type : ApplicationGroup<br>Entry Name : NETGROUP | PolicyDB Name : NEW_PDB<br>Enterprise Name : SAPLEX |                                  |
| Action      System                                     | Selected Resources                                  |                                  |
| s            MVSA                                      | 1   |                                  |
| s            MVSB                                      | 1   |                                  |

### RESOURCES policy:

- VTAM is the only resource selected as a member of NETGROUP
- Select each system to add the IPGROUP APG as a member of the NETGROUP APG

### Step 4: Modify resource links

NETGROUP contains only one resource, VTAM, on each of the two systems. Based on the original description, the IPGROUP APG must be added to the NETGROUP. You select each system and the IPGROUP on each system. You see a screen like the one on the next slide (for each system). You can define a group on multiple systems, but its membership does not have to be the same across all systems. Therefore, the group composition can be different on each system.

## Modify resource links: Add IPGROUP

| AOFGARS3   |          | Select Resources      |  | Row 5 to 17 of 17<br>SCROLL==> CSR                        |
|--|----------|-----------------------|--|---|
| Command ==>  |          |                       |  |   |
| Entry Type : ApplicationGroup<br>System : MVSA   |          | Entry Name : NETGROUP |  | Change Show all Resources to YES to display eligible APGs |
| Number of selected Resources : 2<br>Show all Resources . . . . . YES<br>Show only Resources with String: . |          | YES NO                |  |   |
| Action   | Status   | Resource Name         | Entry Name                                 |   |
|  | SELECTED | IPGROUP/APG/MVSA      | IPGROUP                                    |   |
|  |          | MYSERV1/APG/MVSA      | MYSERV1                                    |   |
|  |          | OM_CICS/APG/MVSA      | OM_CICS                                    |   |
|  |          | OM_DR2/APG/MVSA       | OM_DR2                                     |   |
|  |          | OM_                   | Select IPGROUP/APG/MVSA to add to NETGROUP |   |
|  |          | OM_                   | Repeat the process for the MVSB system     |   |
|  |          | OM_MQ/APG/MVSA        | OM_MQ                                      |   |
|  |          | OM_NV/APG/MVSA        | OM_NV                                      |   |
|  |          | OM_ZOS/APG/MVSA       | OM_ZOS                                     |   |
|  |          | SSH_DAEMON/APG/MVSA   | SSH_DAEMON                                 |   |
|  |          | TELNET_3270/APG/MVSA  | TELNET_3270                                |   |
|  | SELECTED | VTAM/APL/MVSA         | VTAM                                       |   |

© Copyright IBM Corporation 2019

8-35

### Modify resource links: Add IPGROUP

Specify **YES** for the **Show all Resources** field. You see all possible Application Groups that can be selected into this group (NETGROUP). Because this is a system Application Group, you see all APGs defined for the selected system (with automation names). Type an **S** to select the IPGROUP on the system, and link it into the NETGROUP system Application Group. Repeat this step for each system.

# INGGROUP for NETGROUP APG

| INGKYGRB                           |                         | SA z/OS - Command Dialogs |            | Line 1 of 2         |        |     |     |      |     |     |
|------------------------------------|-------------------------|---------------------------|------------|---------------------|--------|-----|-----|------|-----|-----|
| Domain Id . : A0FDA                |                         | INGGROUP                  |            | Date . . : 03/25/19 |        |     |     |      |     |     |
| Operator Id : INGC102              |                         | Sysplex = SYSPLEX1        |            | Time . . : 21:58:35 |        |     |     |      |     |     |
| Group: NETGROUP/APG/MVSA           | Nature: Basic           | Passive: NO               | Suspend:   |                     |        |     |     |      |     |     |
| Description: Network: VTAM & TCPIP |                         |                           |            |                     |        |     |     |      |     |     |
| Excluded :                         |                         |                           |            |                     |        |     |     |      |     |     |
| Avoided :                          |                         |                           |            |                     |        |     |     |      |     |     |
| Mode : Normal                      | Availability Target: 2  | Adjust: 0                 | Result=> 2 |                     |        |     |     |      |     |     |
|                                    | Satisfactory Target=> 2 | Adjust: 0                 |            |                     |        |     |     |      |     |     |
| Rolling Recycle: N/A               |                         |                           |            |                     |        |     |     |      |     |     |
| Name                               | Type                    | System                    | Pref       | Adj                 | Result | Avl | Eff | Stat | Act | Sus |
| IPGROUP                            | APG                     | MVSA                      | 700        | 0                   | 700    | Yes | 925 | Sel  | --  | --  |
| VTAM                               | APL                     | MVSA                      | 700        | 0                   | 700    | Yes | 925 | Sel  | --  | --  |

© Copyright IBM Corporation 2019

8-36

## INGGROUP for NETGROUP APG

To verify that the System Basic Group was created correctly, perform these tasks:

- View the Application Group policy from the automation agent.
- Use the INGGROUP command.
- View the Policy Data Base report.

Use INGGROUP to display information about NETGROUP. In this example, NETGROUP has two members, IPGROUP and VTAM. You can also see that IPGROUP is another Application Group (Type=APG) and VTAM is an Application (Type=APL).

# Student exercises

## Unit 8

1. Create and work with a Basic Application Group
2. Create and work with a Server Application Group



© Copyright IBM Corporation 2019

8-37

### *Student exercises*

Perform the exercises for this unit.

## Summary

---

Now that you have completed this unit, you should be able to

- Describe application group scenarios
- Use the Customization Dialog to define the following Application Groups:
  - Sysplex
  - System
  - Basic
  - Server
  - Move

Now that you have completed this unit, you should be able to:

- Describe application group scenarios
- Use the customization dialog to define the following Application Groups:
  - Sysplex
  - System
  - Basic
  - Server
  - Move



## 9 End-to-end automation

---



### Unit 9 End-to-end automation (SA z/OS only)



© Copyright IBM Corporation 2019  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

---

**What this unit is about**

This unit covers cross sysplex and end-to-end automation, its architecture, customization, policy definition, operation user interfaces, and scenarios. Use objectives.

**How you check your progress**

**References**

*SC34-2750-02 System Automation for z/OS Version 4 Release 1 End-to-End Automation*  
[https://www.ibm.com/support/knowledgecenter/SSWRCJ\\_4.1.0/com.ibm.safos.doc\\_4.1/manuals.html](https://www.ibm.com/support/knowledgecenter/SSWRCJ_4.1.0/com.ibm.safos.doc_4.1/manuals.html)

# Objectives

---

After completing this unit, you should be able to perform the following tasks:

- Explain the end-to-end automation architecture
- Describe and customize the end-to-end adapter
- Describe and customize the end-to-end agent
- Describe the Universal Automation Adapter
- Explain reference resources and remote domains
- Implement cross sysplex and end-to-end automation

After completing this unit, you should be able to perform the following tasks:

- Explain the end-to-end automation architecture
- Describe and customize the end-to-end adapter
- Describe and customize the end-to-end agent
- Describe the Universal Automation Adapter
- Explain reference resources and remote domains
- Implement cross sysplex and end-to-end automation

# Lesson 1. End-to-end automation overview

## Lesson 1: End-to-end automation overview

End-to-end automation:

- Functions
- Architecture
- Concepts
- Roles

### What this lesson is about

This lesson covers an end-to-end automation overview.

### Objectives

After completing this lesson, you should be able to describe the end-to-end automation functions, architecture, concepts, and roles.

### References

*SC34-2716 System Automation for z/OS Version 4  
Release 1 Installation and Planning*

## End-to-end automation

### Goals and functions

Implement an **end-to-end command center** with:

- **End-to-end operations** interface in SMU or NetView with easy operations at the business application level
- **End-to-end automation** to manage cross system, cross platform dependencies

Functions:

- **Start and stop** business applications including remote resources
- **Monitor business applications** including remote resources
- **Compose business applications** that have high availability across multiple automation domains
- Manage a group of remote resources
- **High availability** using recovery, server- and move groups
- **Alerting**

© Copyright IBM Corporation 2019

9-4

*End-to-end automation goals and functions*

With end-to-end automation, you can Implement an **end-to-end command center** with:

- **End-to-end operations** interface in SMU or NetView with easy operations at the business application level
- **End-to-end automation** to manage cross system, cross platform dependencies

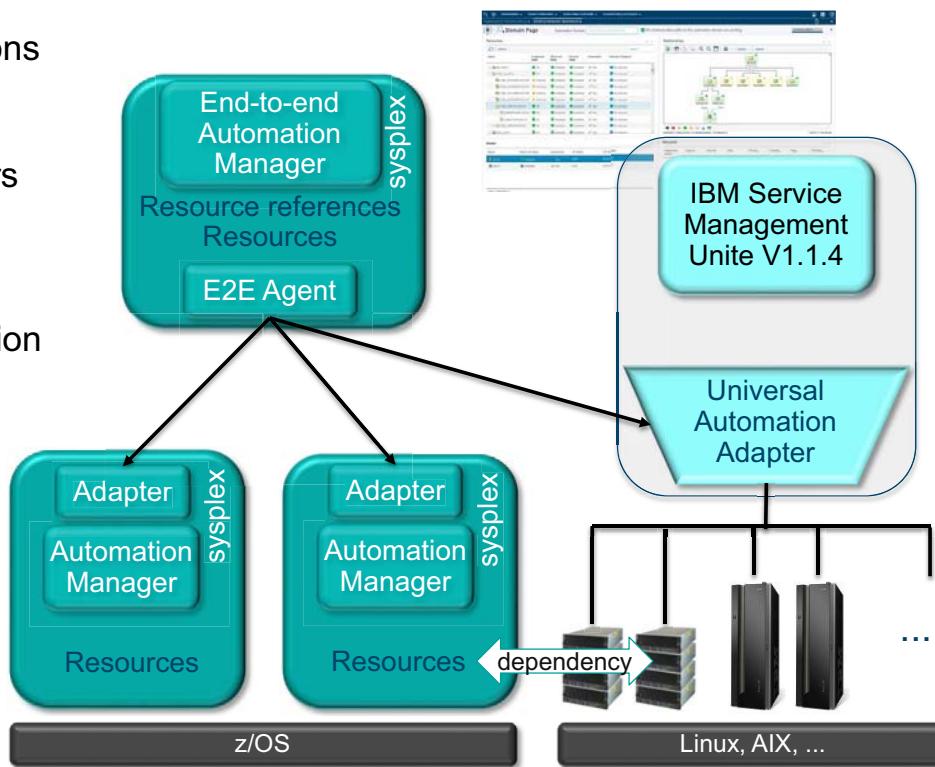
Functions:

- Start and stop business applications including remote resources
- Monitor business application including remote resources
- Compose business applications that have high availability across multiple automation domains
- Manage a group of remote resources
- High availability using recovery, server- and move groups. You have man SA policy functions available for remote resources
- Alerting on SDF or using the SA INGALERT infrastructure

# End-to-end automation Architecture

Control applications running on z/OS sysplexes and on distributed servers

Manage cross platform automation dependencies



© Copyright IBM Corporation 2019

9-5

## End-to-end automation architecture

Before using end-to-end (E2E) automation, you must have activated it. If activated, the primary automation manager (PAM) also becomes the **E2E manager** and automates the resources in other SAplexes or distributed servers.

Originally, you define a relationship between resources that all run in the same SAplex, for example, hasParent relationship between a child and its parent. With SA z/OS 4.1, you can define a cross-sysplex relationship to resources outside of the local SAplex. Along with APAR OA55386, you can also define a cross platform relationship to resources on remote non-z/OS systems.

The **automation adapters** and **Universal Automation Adapter** send events and receive synchronous requests for execution from the E2E manager.

This allows you to:

- Control applications running on z/OS sysplexes and on distributed servers
- Manage cross platform automation dependencies

# End-to-end automation Concepts

- E2E manager
  - Manages dependencies between resources in different automation domains
  - The PAM becomes the end-to-end automation manager
  - Does not replace the local automation products on a remote domain
  - Requests to start or stop resource references that point to remote resources
- E2E agent
  - Executes start/stop orders
  - Collects status of resources running on remote automation domains
  - Interfaces with one or multiple remote automation adapters
- Service Management Unite (SMU) Server
  - Connects to E2E automation adapters to provide SA dashboards
  - Connects to local Universal Automation Adapter to manage remote non z/OS resources
  - Provides policy with start, stop, and monitor commands for non z/OS resources
- E2E automation adapter
  - Connects an SA z/OS automation domain with the SMU server or with the E2E agent
- Universal Automation Adapter
  - Connects an automation domain with the SMU server or with the E2E agent
  - Connects to the remote non-z/OS systems using Secure Shell (SSH)
  - Invokes start or stop commands for remote resources on behalf of E2E automation manager

© Copyright IBM Corporation 2019

9-6

## End-to-end automation concepts

- **E2E manager**
  - Manages dependencies between resources in different automation domains
  - The PAM becomes the end-to-end automation manager
  - Does not replace the local automation products on a remote domain
  - Requests to start or stop resource references that point to remote resources

- **E2E agent**

In conjunction with the E2E manager, there is a new address space, which represents the E2E agent. The E2E agent assists the E2E manager by

- Executing start/stop orders
- Collecting status of resources running on remote domains
- Interfacing with one or multiple remote automation adapters

- **Service Management Unite (SMU) Server**

- Connects to E2E automation adapters to provide SA dashboards
- Connects to local Universal Automation Adapter to manage remote non z/OS resources

- Provides policy with start, stop, and monitor commands for non z/OS resources
- **E2E automation adapter**
  - Connects a remote domain with the SMU server or with the E2E agent
  - Monitors status of resources within the automation domain
  - Sends resource status change events
  - Starts and stops resources within the automation domain
  - Provides information about resources in response to queries
- **Universal Automation Adapter**
  - Connects a remote domain with the SMU server or with the E2E agent
  - Connects to the remote non-z/OS systems using Secure Shell (SSH)

Invokes start or stop commands for remote resources on behalf of the E2E automation manager

# End-to-end automation

## Concepts (cont.)

- Remote resource
  - Is a real resource on a remote automation domain
  - Managed either by SA z/OS E2E automation adapter or by the Universal Automation Adapter
- Resource reference
  - Resource of type REF that references a real resource on a remote automation domain
  - The real resource is either on a remote SAplex and managed by SA z/OS E2E automation adapter, or on a non-z/OS domain and managed by the Universal Automation Adapter
  - Can be added to groups (APG) and used for dependencies
  - Displayed in INGLIST, INGINFO...
- Automation domain (DMN)
  - SAplex connected to the E2E automation agent by the E2E automation adapter
  - Non-z/OS domain connected to the E2E automation agent by the Universal Automation Adapter
  - Displayed in INGLIST, INGINFO...

| SA z/OS - Command Dialogs |           |                   |     |             |              |                    |           |             |           |
|---------------------------|-----------|-------------------|-----|-------------|--------------|--------------------|-----------|-------------|-----------|
| Domain Id . . .           |           | operator Id . . . |     | domain name |              | Sysplex = AOC4PLEX |           | Line 1 of 4 |           |
| A Update                  | B         | H DISPTRG         | I   | J INGSCHED  | K INGGROUP   | L INGCICS          | E INGVOTE | F INGINFO   | G Members |
| R Resume                  | S Suspend | T INGTWS          | S   | P           | Q INGPMS     | R INGPWHY          | X INGWYS  | M DISPMTTR  | P INGPAC  |
| CMD Name                  | Type      | System            | Sus | Sus         | Compound     | Desired            | / scroll  | Observed    | Nature    |
| LINUXENV                  | DMN       |                   |     |             | SATISFACTORY | AVAILABLE          |           | AVAILABLE   |           |
| SMU                       | REF       | LINUXENV          |     |             | SATISFACTORY | AVAILABLE          |           | AVAILABLE   |           |
| VNC                       | REF       | LINUXENV          |     |             | SATISFACTORY | AVAILABLE          |           | AVAILABLE   |           |
| WEB_SRV                   | REF       | LINUXENV          |     |             | PROBLEM      | AVAILABLE          |           | HARDDOWN    |           |

reference resource name
domain name where REF resource belongs to

© Copyright IBM Corporation 2019

9-7

End-to-end automation concepts (cont.)

- **Remote Resource**
  - Is a real resource on a remote domain
  - Managed either by the remote SA z/OS or by the end-to-end automation manager using the Universal Automation Adapter
- **Resource Reference**
  - Resource of type REF that references a real resource on a remote domain
  - Can be added to groups (APG) and used for dependencies
  - The status of the resource reference represents the status of the remote resource
  - Displayed in INGLIST, INGINFO
  - Manageable in INGREQ, INGSET, INGSCHED...
- **Remote Domain (DMN)**
  - SAplex connected to the E2E automation agent by the E2E automation adapter
  - Non-z/OS domain connected to the E2E automation agent by the Universal Automation Adapter

- Displayed in INGLIST, INGINFO...

## End-to-end automation

### End-to-end manager, agent, and adapter roles

- **End-to-end manager** makes decisions
  - Sends start/stop orders
- **E2E agent**
  - Registers for E2E automation
  - Executes start and stop request
  - Check health
  - (Un)subscribes for events
  - Receives events
  - Feeds status changes into PAM
- **E2E adapter**
  - Translates and executes requests
  - Sends status change events
  - Persistent connection data

© Copyright IBM Corporation 2019

9-8

*End-to-end manager, agent, and adapter roles*

End-to-end manager makes decisions

- Sends start/stop orders

E2E agent

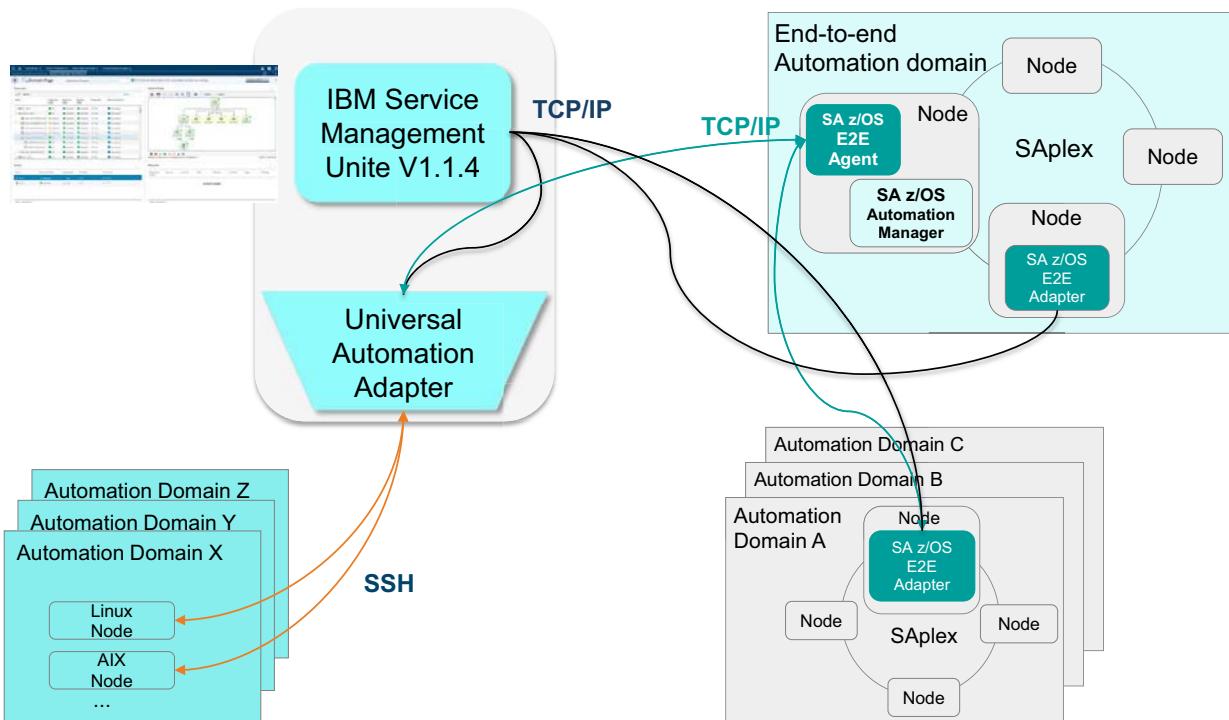
- Registers for E2E automation
- Executes start and stop request
- Check health
- (Un)subscribes for events
- Receives events
- Feeds status changes into PAM

E2E adapter

- Translates and executes requests
- Sends status change events
- Persistent connection data

# End-to-end automation

## Architecture details



© Copyright IBM Corporation 2019

9-9

### End-to-end automation architecture details

Each SA z/OS automation domain represents an SAplex. It contains one or multiple z/OS systems with System Automation for z/OS and the end-to-end automation adapter. The automation adapter can run on only one system in the SAplex. The system where it runs is automatically selected as the primary agent. This is shown in command INGAMS by YES in column E2E.

### Configuration and activation

The end-to-end agent can be configured as part of the SA Configuration Assistant. To activate the end-to-end automation, you must start the E2E agent address space. During initialization, the E2E agent loads the REF and DMN objects from the automation policy and registers these resources to the primary automation manager and thereby activates end-to-end automation for the corresponding REF and DMN resources.

### Security

When the end-to-end automation manager issues requests against remote resources, it must authenticate itself to the first-level automation domains that host the remote resources. For authentication, the end-to-end automation manager uses the user credentials (user ID and password) that are specified in the credential file of the E2E agent.

## End-to-end adapter

The successful initialization of the automation adapter on a system makes the System Automation NetView/Agent to be the automation agent that communicates with the PAM, wherever it is. INGAMS command shows this indication.

### Synchronous Communication

The automation adapter receives a request from the SMU server or from the E2E agent and schedules it via PPI communication to an SA z/OS task execution request processor that runs on the automated operator function E2EOPER or E2EOPRnn.

### Asynchronous Communication

The automation adapter provides an EIF event receiver and an EIF event emitter. SA z/OS acts as an asynchronous data provider and sends SA z/OS specific events via E/AS to the EIF event receiver of the automation adapter. The automation adapter converts the events to a format understood by SMU server or by the E2E agent. The formatted events are sent to one or both targets through the EIF emitter component.

### The NetView Event/Automation Service (E/AS)

The message adapter service of the NetView event/automation service (E/AS) is used to convert and forward messages from NetView to a designated event server, such as the end-to-end automation adapter. The end-to-end automation adapter requires a separate E/AS address space for its own. INGEVE2E is specified as the PPI receiver name in E/AS initialization member IHSAINIT.

# Lesson 2. Prerequisites, planning, and activation

## Lesson 2: Prerequisites, planning, and activation

- Prerequisites
- Planning
- Configuration assistant
- Activation steps

### What this lesson is about

This lesson provides details about end-to-end automation prerequisites, planning, configuration assistant, and activation.

## Objectives

After completing this lesson, you should be able to describe the prerequisites of end-to-end automation, to do the planning and activation steps

## References

*SC34-2750-02 System Automation for z/OS Version 4 Release 1 End-to-End Automation*  
*SC27-8747-00 System Automation for z/OS Version 4 Release 1 Service Management Unite Automation Installation and Configuration Guide*

# End-to-end automation

## Prerequisites

- SA z/OS 4.1 with APAR OA55386
- Java™ Runtime Environment (JRE) 1.6 or higher installed on z/OS
  - The **32-bit** Java version is required
  - It is recommended to use IBM Java
  - Customizable with environment variable in configuration files
- The Event/Automation Service (E/AS) component of NetView
- Full z/OS UNIX System Services (USS) with USS file system
- SSI address space with PPI function of NetView
- TCP/IP
- RACF if authentication is enabled

### *End-to-end automation prerequisites*

The following prerequisites are required for the automation adapter and agent:

- SA z/OS 4.1 and IBM® Tivoli® NetView® and z/OS as described in the program directory of SA 4.1.
- **Java™ Runtime Environment (JRE) 1.6** or higher installed on z/OS. The 32-bit Java version is required. It is recommended to use IBM Java.  
If the default java runtime cannot be changed, set the environment variable E2E\_JAVA\_HOME in the automation adapter configuration file ingadapter.properties to the required java runtime. If the default java runtime cannot be changed, set the environment variable E2EAGT\_JAVA\_HOME in the automation agent configuration file inge2eagt.properties to the required java runtime.
- The Event/Automation Service (E/AS) component of NetView.
- Full z/OS UNIX System Services (USS) with USS file system
- SSI address space with PPI function of NetView
- TCP/IP
- RACF if authentication is enabled.

# End-to-end automation

## Planning

- Identify applications with dependencies to other SAplexes or domains or non z/OS applications
  - Identify resources and their owners
  - Get buy-in of owners and command center
  - Analyze dependencies
  - Model e2e application: groups, members, dependencies
  - Gather start, stop, and monitor commands for e2e resources (outside of SAplexes), required user-ids, ssh certificates
- Choose SAplex(es) with E2E role
  - Good candidates are always on SAplex(es) with resources having dependencies to other SAplexes or domains

### *End-to-end automation planning*

Identify applications with dependencies to other SAplexes or domains or non z/OS applications:

- Identify resources and their owners
- Get buy-in of command center, owners and management
- Analyze dependencies for startup, shutdown, recovery and location.
- Model e2e application: groups, members, dependencies into SA policy
- Gather start, stop, and monitor commands for e2e resources (outside of SAplexes), required user-ids, ssh certificates. They can be defined using the e2e policy in the UAA.

### Choose SAplex(es) with E2E role

Good candidates are always on SAplex(es) with resources having dependencies to other SAplexes or domains. Otherwise implement a standalone SAplex only for e2e.

# End-to-end automation

## Configuration assistant - OA52610

|  |   |   |                                       |                                       |  |          |  |          |  |          |  |          |  |
|--|---|---|---------------------------------------|---------------------------------------|--|----------|--|----------|--|----------|--|----------|--|
| <pre> *   ... Do you want to connect your local SAplex to an SMU *   application? * *   If you intend to connect your local SAplex to an SMU *   application then provide the IP address of the system hosting *   the SMU application. Based on this information the ready to *   be used components will be generated into the CONFLIB data *   set.  * *   Option .....: sa_e2e_smu_ip *   Required ....: No *   Default .....: None *   Example.....: smu.compan *                 ...: 204.0.114. *                 ...: 2001:db7:0 * sa_e2e_smu_ip= </pre> | <b>INGDOPT (SINGSAMP)</b><br><br><table border="0"> <tr> <td style="vertical-align: top;">           End-To-End Agent<br/>           Dynamic VIPA for End-To-End Agent<br/> <br/>           End-To-End Adapter         </td><td style="vertical-align: top; padding-left: 20px;">           INGXEAAGT<br/>           INGDVIPA<br/> <br/>           INGXADPT         </td></tr> <tr> <td colspan="2" style="text-align: center; padding-top: 20px;">           NetView Event Automation Service E/AS         </td></tr> <tr> <td colspan="2" style="text-align: center; padding-top: 10px;">           IHSAEVNT         </td></tr> <tr> <td colspan="2" style="text-align: center; padding-top: 10px;">           IHSAINIT         </td></tr> <tr> <td colspan="2" style="text-align: center; padding-top: 10px;">           IHSAMCFG         </td></tr> <tr> <td colspan="2" style="text-align: center; padding-top: 10px;">           INGMFMTE         </td></tr> </table> | End-To-End Agent<br>Dynamic VIPA for End-To-End Agent<br><br>End-To-End Adapter | INGXEAAGT<br>INGDVIPA<br><br>INGXADPT | NetView Event Automation Service E/AS |  | IHSAEVNT |  | IHSAINIT |  | IHSAMCFG |  | INGMFMTE |  |
| End-To-End Agent<br>Dynamic VIPA for End-To-End Agent<br><br>End-To-End Adapter  | INGXEAAGT<br>INGDVIPA<br><br>INGXADPT   |   |                                       |                                       |  |          |  |          |  |          |  |          |  |
| NetView Event Automation Service E/AS  |   |   |                                       |                                       |  |          |  |          |  |          |  |          |  |
| IHSAEVNT   |   |   |                                       |                                       |  |          |  |          |  |          |  |          |  |
| IHSAINIT   |   |   |                                       |                                       |  |          |  |          |  |          |  |          |  |
| IHSAMCFG   |   |   |                                       |                                       |  |          |  |          |  |          |  |          |  |
| INGMFMTE   |   |   |                                       |                                       |  |          |  |          |  |          |  |          |  |

© Copyright IBM Corporation 2019

9-13

### End-to-end automation configuration assistant - OA52610

Consider using the configuration assistant. The e2e infrastructure is supported with APAR OA52610. The configuration assistant can generate the procedures the customized configuration files for adapter, agent, and E/AS.

## End-to-end automation

### Activation steps

1. Define domain (DMN) and resource reference (REF) objects in policy
  2. Link REF(s) to the DMN
  3. Link DMN(s) to the SAplex with E2E role
  4. Build the automation configuration file
  5. Load the automation configuration file the SAplex with E2E role
- The PAM that has loaded DMN and REF objects is not yet an active E2E manager
6. Start the E2E agent address space on the e2e SAplex
    - Enables the PAM to become the E2E Manager
  7. Start the E2E adapters on the remote domain(s)

#### *End-to-end automation activation steps*

- Policy
  - a. Define domain (DMN) and resource reference (REF) objects in policy.
  - b. Link REF(s) to the DMN.
  - c. Link DMN(s) to the SAplex with E2E role.
  - d. Optionally create sysplex groups and link resource references to them. System groups should NOT be used.
  - e. Build the automation configuration file.
  - f. Load the automation configuration file the SAplex with E2E role. The PAM that has loaded DMN, REF objects is not yet an active E2E manager.
- Start the E2E agent address space on the e2e Saplex, which enables the PAM to become the E2E Manager. Check with INGAMS command.
- Start the E2E adapters on the remote domain(s).

# Lesson 3. End-to-end agent

## Lesson 3: End-to-end agent

- The E2E Agent is a moveable address space in SAplex
- New address space as USS/Java application
- May run on any system in the SAplex
- Only one E2E agent per SAplex
- Move group in sample policy \*E2E
- If moveable, a DVIPA is required to listen for events from remote domain

### What this lesson is about

This lesson provides details about the end-to-end automation agent.

### Objectives

After completing this lesson, you should be able to describe the end-to-end adapter customization, to start and stop the end-to-end agent, and to implement authentication to a remote domain.

### References

*SC34-2750-02 System Automation for z/OS Version 4 Release 1 End-to-End Automation*

Prior to V4.1, SA z/OS automation manager automated resources in the same sysplex (or SAplex). Now the SA z/OS automation manager is able to automate resources between multiple sysplexes (or SAplexes) and cross platform resources.

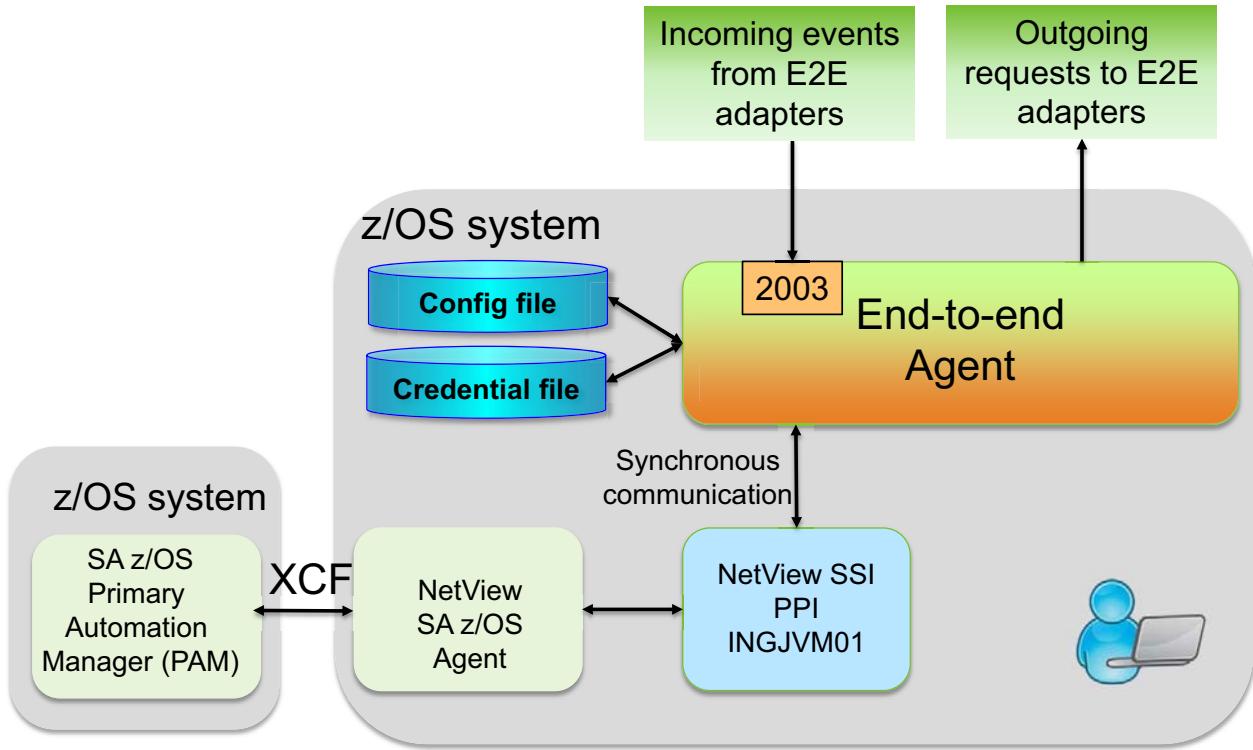
In this case, the primary automation manager (PAM) is also called E2E manager. This cross sysplex/platform automation is also called end-to-end automation.

In conjunction with the E2E manager, there is a new address space, which represents the E2E agent. The E2E agent assists the E2E manager by executing start and stop orders or collecting status of resources running on remote automation domains. The E2E agent interfaces with the remote automation adapter.

The E2E Agent is a New address space as USS/Java application which is moveable in SAplex. It may run on any system in the SAplex, only one E2E agent per SAplex is required. If moveable, a DVIPA is required to listen for events from remote domain

A move group and all other required policy is available in sample policy \*E2E.

## End-to-end agent



© Copyright IBM Corporation 2019

9-16

### End-to-end agent

End-to-end automation can be connected or disconnected to an automation domain. You connect an automation domain to end-to-end automation when you start the corresponding E2E automation adapter on the remote domain and the E2E agent is able to establish a TCP/IP connection to this E2E automation adapter.

There is a synchronous and asynchronous communication path from the E2E agent to the remote automation adapter. INGJVM01 is the PPI receiver queue name used by the E2E agent.

A check-health algorithm combined with heartbeat events determines periodically the observed status of the DMN. The observed status of the DMN object is AVAILABLE only if synchronous and asynchronous communication states are OK. Otherwise, the observed status is UNKNOWN or HARDDOWN if a problem was detected. It is only at the first time after a cold or warm start of the PAM that the observed status of a DMN is SYSGONE.

The E2E agent needs to be recycled when NetView recycles, when the NetView SSI address space recycles, when the PAM recycles or moves.

# Environment Setup for E2E agent Java component

- Java runtime version 1.6 or higher: 32-bit java runtime
  - E2EAGT\_JAVA\_HOME in inge2eagt.properties
- Install directory /install-root/
  - SMP/E installs E2E agent into the install directory, see SINGINST(INGMKDIR)
  - Per default, /usr/lpp/ing/adapter/, (same as for E2E adapter)
  - Read only
- Custom directory /custom-root/
  - Customer creates it and copies files
  - Read/write

## *Environment setup for E2E agent Java component*

The System Automation end-to-end agent is installed by default through SMP/E into the same zFS directory as used by the automation adapter. The customization directory might also be the same as that of the automation adapter.

The automation adapter and/or the end-to-end agent require a directory structure within the USS file system.

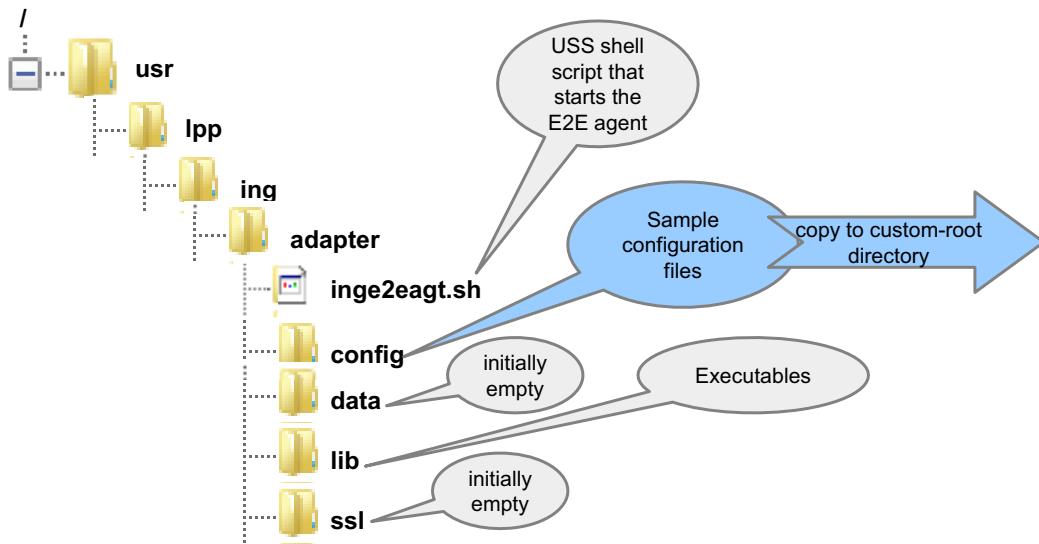
There is a properties file that defines environment variables:

- Automation adapter: ingadapter.properties
- End-to-end agent: inge2eagt.properties

Within these properties files, you can define:

- install-root directory which is normally read-only
- custom-root directory which contains all customized configuration files and the log files.

## Default install-root directory: /usr/lpp/ing/adapter/



© Copyright IBM Corporation 2019

9-18

*Default install-root directory: /usr/lpp/ing/adapter/*

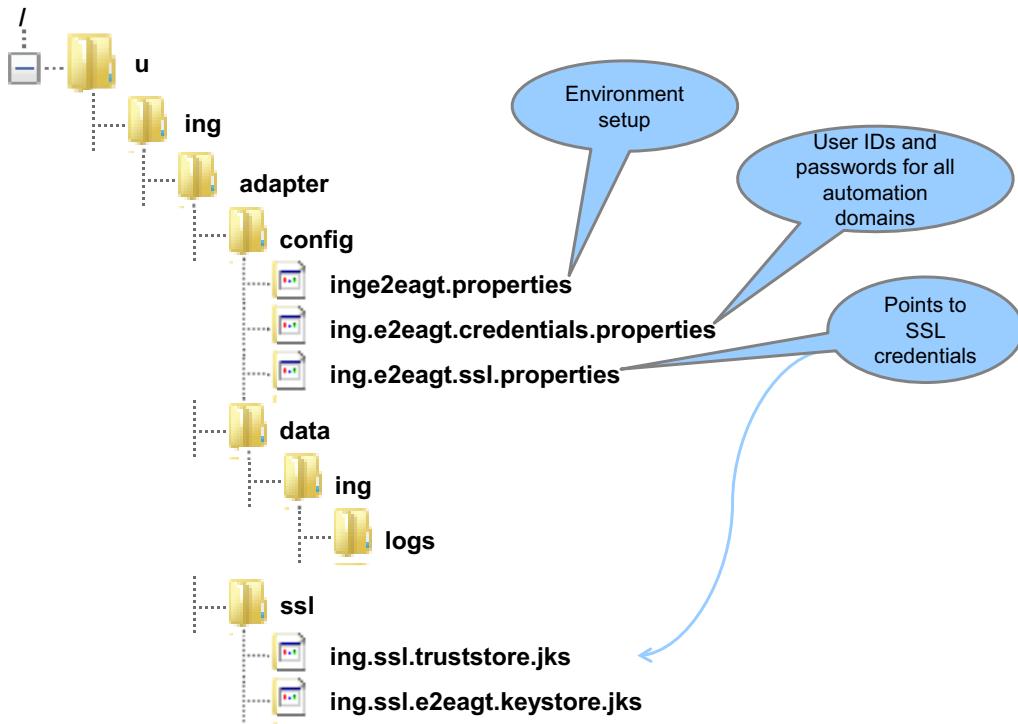
The SMP/E installation creates a default file structure for the various files that are associated with the automation adapter and/or the end-to-end agent, with the following subdirectories:

1. /usr/lpp/ing/adapter
2. /usr/lpp/ing/adapter/config
3. /usr/lpp/ing/adapter/data
4. /usr/lpp/ing/adapter/lib
5. /usr/lpp/ing/adapter/ssl

where the different paths are for:

1. Executable files, for example, the automation adapter start and stop scripts
2. Configuration files, for example, the master configuration file
3. Working files, for example, release information file or log files. This directory is initially empty.
4. JAR files and DLLs for the automation adapter
5. Security certificates. The directory is initially empty.

## Example custom-root directory: /u/ing/adapter/



© Copyright IBM Corporation 2019

9-19

*Example custom-root directory: /u/in/adapter/*

Create the following sub-directories for the custom-root directory:

1. /custom-root/adapter
2. /custom-root/adapter/config
3. /custom-root/adapter/data
4. /custom-root/adapter/ssl

Copy the following files from the install-root directory into the custom-root directory.

```
cd /install-root/adapter/config
cp /ingadapter.properties /custom-root/adapter/config/ingadapter.properties
cp /ing.adapter.properties /custom-root/adapter/config/ing.adapter.properties
cp /ing.adapter.plugin.properties
/custon-root/adpater/config/ing.adapter.plugin.properties
cp /ing.adapter.ssl.properties
/custon-root/adapter/config/ing.adapter.ssl.properties
```

## E2E agent environment configuration file

/custom-root/config/inge2eagt.properties

- Used by //STDENV DD statement in JCL proc INGXEAQT
- Variables that must be modified
  - INGAGT\_INSTALL\_ROOT=/usr/lpp/ing/adapter
  - INGAGT\_CUSTOM\_ROOT=/usr/lpp/ing/adapter
- Variables that shall very likely be modified
  - INGAGT\_JAVA\_HOME=/local/java/J6.0
  - INGAGT\_HOST=DVIPA.IP.ADDR
- Variables that you may want to modify
  - INGAGT\_E2EDOM=E2E Domain
  - INGAGT\_CKH\_INTERVAL=300
- Any other variable
  - Stay with default value
  - See also “E2E Automation” manual chapter 4

*E2E agent environment configuration file*

Modify the following environment variables within the configuration file ingadapter.properties.

```
2E_INSTALL_ROOT=/install-root/adapter
E2E_CUSTOM_ROOT=/custom-root/adapter
EE2E_JAVA_HOME=32-bit java-home-directory
```

Modify the following remaining files to your needs. For details about the parameters of the configuration files,

```
/custom-root/adapter/config/ing.adapter.properties
/custom-root/adapter/config/ing.adapter.plugin.properties
/custom-root/adapter/ssl/ing.adapter.ssl.properties
```

The automation adapter writes output data into custom-root/adapter/data.

## E2E agent JCL procedure

Copy INGXEAGT from ING.SINGSAMP to your proc-lib

- Customize to your needs
- DIRI and DIRC... is the install and custom root directory

```
//INGXEAGT PROC XID='.',
//              RID='01',
//              PORT='2003',
//              OPT='OFF',
//              SX=,
//              DIRI='/usr/lpp/ing/adapter',
//              DIRC='#custom-dir',
//              NAME='inge2eagt'
//CLEANUP EXEC PGM=BPXBATCH,REGION=0M,
//          PARM='SH rm &DIRC/data/std*.&SX.&RID..txt'
//RUNAGT  EXEC PGM=BPXBATCH,REGION=0M,TIME=1440,
//          PARM='SH &DIRI/&NAME..sh &XID &RID &PORT &OPT &SX'
//STDOUT  DD PATH='&DIRC/data/stdout.&SX.&RID..txt',
//          PATHOPTS=(OWRONLY,OCREAT),PATHMODE=(SIRWXU,SIRWXG)
//STDERR  DD PATH='&DIRC/data/stderr.&SX.&RID..txt',
//          PATHOPTS=(OWRONLY,OCREAT),PATHMODE=(SIRWXU,SIRWXG)
//SYSPRINT DD SYSOUT=*
//CEEUMP  DD SYSOUT=*
//*-----  
/* Configuration parameters overwrite defaults in the script
//*-----  
//STDENV DD PATH='&DIRC/config/&NAME..properties',
//          PATHOPTS=ORDONLY
```

STDENV points to the  
inge2eagt.properties

© Copyright IBM Corporation 2019

9-21

### E2E agent JCL procedure

You can use the JCL procedure INGXEAGT to start the end-to-end agent. This procedure is provided in the System Automation sample library. It starts the end-to-end agent via the USS shell script /usr/lpp/adapter/inge2eagt.sh. The shell script is read-only.

To customize the end-to-end agent, customize the parameters in the configuration file inge2eagt.properties. The DD STDENV in the JCL procedure points to this configuration file.

Copy the sample JCL procedure INGXEAGT from ING.SINGSAMP into your proclib. Rename it to a name of your choice and customize it. Follow the comments in the sample and specify the installation and customization paths in parameter DIRI and DIRC.

Below you find a sample z/OS command that starts the end-to-end agent as a started task:

```
S INGXEAGT,JOBNAM=E2EAGENT,XID=00,RID=01,PORT=2003,OPT=DEBUG,SX=SYS1
```

## E2E agent credential file

/custom-root/config/inge2eagt.credentials.properties

- E2E adapter logs on to remote automation domain
- SAF (e.g. RACF) user ID and password
  - Generic, used if not defined otherwise
  - Domain specific, used for specified automation domain only
- Password might be encrypted
  - See encryption tool in “E2E Automation” manual chapter 4

**inge2eagt.credentials.properties**

```
#--  
## Generic pair  
##  
user-generic=GENUSER  
password-generic=password  
##  
## Optional domain pairs  
##  
# user-<DOMAINNAME>=<SPEC_USERID>  
# password-<DOMAINNAME>=<SPEC_PASSWORD>  
user-SYSPLEX INGXSG34=BDOW  
password-SYSPLEX INGXSG34=password
```

© Copyright IBM Corporation 2019

9-22

### E2E agent credential file

The end-to-end agent credentials file contains for each automation domain three items: domain, user ID, password. The end-to-end agent needs these items to log on to a remote domain. The user ID and password is a SAF defined user ID (for example, RACF). It will be passed over the connection between the end-to-end agent and the end-to-end automation adapters. User ID and password is used for the end-to-end automation adapter authentication checking and for third party System Automation authorization checking.

The parameters user-generic and password-generic define the default user ID and password used for all domains if not defined otherwise.

Each domain that is defined in the policy needs to have either a specific or a generic credential statement. For each specified user ID, you must also specify a password. The user ID must be a SAF user ID that is valid on the remote z/OS domain and it must have an OMVS segment defined.

The password might be AES-encrypted or clear text. You can use the script ingencrypt.sh to encrypt the clear test password to AES.

## Start the E2E agent

- Sample proc SINGSAMP(INGXEAGT)
  - Use MVS start command
  - Minimum parameters are jobname and XCF group id

```
INGXEAGT, JOBNAME=E2EAGENT, XID=00  
INGXEAGT, JOBNAME=E2EAGENT, XID=00, RID=01, PORT=2003, OPT=DEBUG, SX=SYSA
```

Description of the parameters:

**XID** The two-character XCF group ID of the associated System Automation/NetView agent

**RID** The two-character suffix of the generated PPI receiver ID, for example, INGJVM01

**PORT** The EIF receiver port for the EIF events sent by the end-to-end adapters

**SX** Optional, may be used for shared zFS. Makes output files unique per system.

It causes the end-to-end agent to write log files into a data directory that depends on the value of SX. It might be the system name or any other user specified differentiators.

For example, stderr.SYSA01.txt The log files will be written into /customization-path/data/SYSA/ing/logs/

**OPT** Optional, this parameter is for debugging only

### Start the E2E agent

The end-to-end agent is a USS process started via started task using BPXINIT. You use the sample JCL procedure ING.SINGSAMP(INGXEAGT) to start the end-to-end agent. The following sample z/OS command starts the end-to-end agent as a started task:

```
S INGXEAGT, JOBNAME=E2EAGENT, XID=00, RID=01, PORT=2003, OPT=DEBUG, SX=SYSA
```

Description of the parameters. RID and PORT have meaningful defaults (RID=01, PORT=2003):

- **XID** The two-character XCF group ID of the associated System Automation/NetView agent. XID must be specified to connect to the System Automation/NetView agent. You can only omit XID if you use the default XCF group ID, which is blank
- **RID** The two-character suffix of the generated PPI receiver ID, for example, INGJVM01.
- **PORT** The EIF receiver port for the EIF events sent by the end-to-end automation adapters.
- **SX** May be used for shared zFS. It causes the end-to-end agent to write log files into a data directory that depends on the value of SX. It might be the system name or any other user specified differentiators. For example, stderr.SYSA01.txt and the log files are written in /customization-path/data/SYSA/ing/logs/.
- **OPT** This parameter is for debugging only.

## Starting the E2E agent

```

Canzlog TAG=(NVMMSG,MVSMMSG,DOM)          05/25/19 00:02:31 -- 00:02:38
00:02:31 IEF196I           VIRT- ALLOC:      6M SHRD:      0M
00:02:32 INGX9730I Starting Automation Agent Java Adapter Script: JOBNAME=E2EAG
ENT XCFID=SA RID=01 INSTALL_ROOT=/usr/lpp/ing/adapter CUSTOM_ROOT=/MVS
A/u/e2e
00:02:32 DOM TCB
00:02:34 $HASP100 BPXAS    ON STCINRDR
00:02:34 $HASP373 BPXAS    STARTED
00:02:34 IEF403I BPXAS - STARTED - TIME=00.02.34
00:02:35 BPXP024I BPXAS INITIATOR STARTED ON BEHALF OF JOB E2EAGENT RUNNING IN
00:02:36 INGX9802I INGXLogger has successfully been initialized using configura
tion file ingjava.jlog.properties from path '/usr/lpp/ing/adapter/conf
ig', logger output path '/MVSA/u/e2e/data/ing_logs'.*
00:02:36 INGX9874I System Automation agent Java adapter 'E2EAGENT' is active.#
00:02:36 CNM493I INGMMSG02 : 00000804 : MSGID=INGX9874I : ACTIVMSG UP=NO
00:02:38 INGX9875I System Automation agent Java adapter 'E2EAGENT' is ready.#
00:02:38 ING278I System Automation E2E Agent cannot find resources of type REF.
00:02:38 ING278I System Automation E2E Agent cannot find resources of type DMN.
00:02:38 ING278I System Automation E2E Agent cannot find resources of type REF.
00:02:38 ING278I System Automation E2E Agent cannot find resources of type DMN.
00:02:38 INGX9634I System Automation E2E agent 'E2EAGENT' the user ID or passwo
rd is not correct specified or credentials file cannot be loaded from

```

**ACTIVE message**

**No E2E resources found**

**E2E credentials
file not
customized**

### Starting the E2E agent

Above you see an unsuccessful attempt to start the E2E Agent.

You see the ACTIVE message INGX98741I, but the agent terminates because:

1. No E2E resources found
2. E2E credentials file not customized

## E2E agent terminates

```
Canzlog TAG=(NVMSG,MVSSMSG,DOM) 05/25/19 00:02:38 -- 00:02:41
00:02:38 rd is not correct specified or credentials file cannot be loaded from
/MVSA/u/e2e/config/inge2eagt.credentials.properties*
00:02:38 AOF760E NOVALUE CONDITION TRAPPED IN INGRE2E LINE 762 (CODE N/A)
DESCRIPTION LRC
00:02:38 AOF760E NOVALUE CONDITION TRAPPED IN INGRE2E LINE 762 (CODE N/A) DESCRIPT
00:02:38 DSI052I INGRXCJ1 COMMAND SELF-TERMINATED BY OPERATOR REQUEST
00:02:38 AOF760E HALT CONDITION TRAPPED IN INGRXCMJ LINE 478 (CODE N/A)
DESCRIPTION CLIST HALTED
00:02:38 AOF760E HALT CONDITION TRAPPED IN INGRXCMJ LINE 478 (CODE N/A) DESCRIPT
00:02:38 DSI052I INGRXCMJ COMMAND SELF-TERMINATED BY OPERATOR REQUEST
00:02:39 INGX9632I System Automation E2E agent 'E2EAGENT' terminates: 'Initiali
00:02:40 INGX9877I System Automation agent adapter 'E2EAGENT' has been stopped
00:02:40 CNM493I INGMSG02 : 00000812 : MSGID=INGX9877I : TERMMMSG FINAL=YES,ABE
00:02:41 DOM TCB
00:02:41 INGX9731I Finished Automation Agent Java Adapter script: RC=4
```

Final  
termination  
message

© Copyright IBM Corporation 2019

9-25

*E2E agent terminates*

Scrolling down in the Canzlog, you see that the E2E credentials file has not been customized. Therefore the E2E Agent terminates with final termination message INGX98771I

# E2E agent terminates again

```
Canzlog TAG=(NVMSG,MVSSMSG,DOM) 05/27/19 14:44:39 -- 14:44:44
14:44:39 $HASP100 BPXAS    ON STCINRDR
14:44:39 $HASP373 BPXAS    STARTED
14:44:39 IEF403I BPXAS - STARTED - TIME=14.44.39
14:44:39 BPXP024I BPXAS INITIATOR STARTED ON BEHALF OF JOB E2EAGENT RUNNING IN
14:44:39 $HASP100 BPXAS    ON STCINRDR
14:44:39 $HASP373 BPXAS    STARTED
14:44:39 IEF403I BPXAS - STARTED - TIME=14.44.39
14:44:39 BPXP024I BPXAS INITIATOR STARTED ON BEHALF OF JOB E2EAGENT RUNNING IN
14:44:39 $HASP100 BPXAS    ON STCINRDR
14:44:39 $HASP373 BPXAS    STARTED
14:44:39 IEF403I BPXAS - STARTED - TIME=14.44.39
14:44:39 BPXP024I BPXAS INITIATOR STARTED ON BEHALF OF JOB E2EAGENT RUNNING IN
14:44:40 INGX9802I INGXLogger has successfully been initialized using configura
        tion file ingjava.jlog.properties from path '/usr/lpp/ing/adapter/conf
        ig', logger output path '/MVSA/u/e2e/data/ing/logs'.*
14:44:40 INGX9874I System Automation agent Java adapter 'E2EAGENT' is active.*/
14:44:40 CNM493I INGMSG02 : 00000816 : MSGID=INGX9874I : ACTIVMSG UP=NO
14:44:42 INGX9871I System Automation agent RPC from Java to NetView failed: ING
        X9820E JNI function ingjippi failed with return code 1726.*/
14:44:44 INGX9877I System Automation agent adapter 'E2EAGENT' has been stopped
14:44:44 CNM493I INGMSG02 : 00000824 : MSGID=INGX9877I : TERMMMSG FINAL=YES,ABE
```

PPI error

Final  
termination  
message

© Copyright IBM Corporation 2019

9-26

*E2E agent terminates again*

Above you see another unsuccessful attempt to start the E2E Agent.

This time the communication to the PPI could not be established.

Therefore the E2E Agent terminates with final termination message INGX98771

## Check if E2E automation is active

| SA z/OS - Command Dialogs |        |                 |                |                         |              |           |         | Line 1 of 3           |
|---------------------------|--------|-----------------|----------------|-------------------------|--------------|-----------|---------|-----------------------|
| Domain Id . : AOFDA       |        |                 |                | INGAMS                  |              |           |         | Date . . . : 05/27/19 |
| Operator Id : INGC102     |        |                 |                | Sysplex = SYSPLEX1      |              |           |         | Time . . . : 19:33:15 |
| <b>Cmd:</b>               |        | A Manage        | B Show Details | C Refresh Configuration | D Diagnostic |           |         |                       |
| CMD                       | System | Member          | Role           | Status                  | Sysplex      | XCF Group | Release | Comm E2E              |
| —                         | MVSA   | MVSA\$\$\$\$\$2 | SAM            | READY                   | PLEX12       | INGXSGSA  | V4R1M0  | XCF                   |
| —                         | MVSA   | MVSA\$\$\$\$\$1 | PAM            | READY                   | PLEX12       | INGXSGSA  | V4R1M0  | XCF                   |
| —                         | MVSA   | MVSA            | AGENT          | READY                   | PLEX12       | INGXSGSA  | V4R1M0  | YES                   |

yes in E2E-column would indicate a “primary” agent

yes in E2E-column indicates PAM is E2E Manager

### Check if E2E automation is active

You can use INGAMS command to display all members of the SAplex.

If the PAM has indication YES in column E2E, then the end-to-end agent is running on this system and end-to-end automation is active. To check whether the NetView agent has been selected as the primary agent to work with the automation adapter, enter the INGAMS command to display the primary agent flag. The value YES in the last column indicates that the primary agent flag has been set for the agent.

## Stop the E2E agent

- INGE2E
  - INGE2E STOP\_E2EAGT
  - INGE2E CLEANUP\_E2EAGT
- SA/NetView Command
- Used in SA automation policy
- Normal termination
- Cleanup after abnormal termination (BROKEN or cancelled)
- E2E agent needs to be recycled when the automation configuration file is refreshed or PAM, agent recycles

### *Stop the E2E agent*

You can stop the end-to-end agent from the NetView console via the operational API command INGE2E STOP\_E2EAGT.

If the end-to-end agent USS process has been terminated abnormally, it has no chance to perform cleaning up allocated resources. Use the following command to clean up the end-to-end agent:

INGE2E CLEANUP\_E2EAGT.

The \*E2E SA automation policy uses above commands.

The E2E agent needs to be recycled when

- NetView recycles
- NetView SSI address space recycles
- PAM recycles
- PAM moves

While the E2E agent is being recycled, it will first be stopped. This sets the status of the DMN and its REF resources to UNKNOWN or WASAVAILABLE. While the E2E agent is being initialized, the status of the DMN and REF resources keep the same as they were previously stored in the PAM.

After the E2E agent is able to establish connection to the DMN, the status of the DMN and its REF resources will be updated.

If you refresh the automation manager configuration file, for example, via INGAMS REFRESH, the E2E agent will be recycled automatically by System Automation when the REF or DMN configuration has changed.

However, recycling the E2E agent due to REFRESH will not cause to set the status of the DMN and its REF resources to UNKNOWN or WASAVAILABLE. They keep as they are and will be updated successively.

## Authentication to a remote domain

- Domain name is key (e.g. “SYSPLEX INGXSG34”) to find correct connection data
- E2E agent checks adapter security flag

```
Remote Domain...
  Name          : KEYAPLEX INGXSG34
  Operational Info : OK - The connection to the domain was successfully established.
  Adapter OS    : z/OS
    Adapter Hostname : boekeya.boeblingen.de.ibm.com
    Adapter Port   : 2001
    Adapter SSL    : false
    Adapter Security : true
```

- E2E agent reads file `inge2eagt.credentials.properties`

```
user-SYSPLEX INGXSG34=BDOW
password-SYSPLEX INGXSG34=password
```

- Pass user ID, password to any E2E request to be sent to remote domain
- If authentication fails then DMN or REF STATUS becomes HARDDOWN

### *Authentication to a remote domain*

When the end-to-end automation manager issues requests against remote resources, it must authenticate itself to the first-level automation domains that host the remote resources. For authentication, the end-to-end automation manager uses the user credentials (user ID and password) that are specified in the credential file of the E2E agent.

If the referenced resource that is targeted by the request is hosted by an automation domain for which specific user credentials have been specified, the automation manager uses these credentials for authentication. If no specific user credentials for the domain are specified in the credential file, the automation manager uses the generic credentials that must be specified in the credential file. If the user ID and password cannot be validated by the automation adapter on the remote domain, the resource reference will be set to HARDDOWN.

The end-to-end agent credentials file is a USS file. The sample files resides in `/usr/lpp/ing/adapter/config/inge2eagt.credentials.properties`. Copy the credentials file to `/customization-path/config` and adapt it to your needs.

# Lesson 4. End-to-end adapter

## Lesson 4: End-to-end adapter

- End-to-end adapter
  - Customization steps
  - Configuration Files
- Enable NetView Event Automation Service (E/AS)
- Customize SA and NetView
- Start End-to-end adapter
- INGE2E command provides the operational API

### What this lesson is about

This lesson provides details about the end-to-end adapter.

### Objectives

After completing this lesson, you should be able to describe the end-to-end adapter customization, to enable the NetView Event Automation Service (E/AS), to customize SA and NetView, to start the end-to-end adapter and use the operational API.

### References

*SC34-2750-02 System Automation for z/OS Version 4 Release 1 End-to-End Automation*

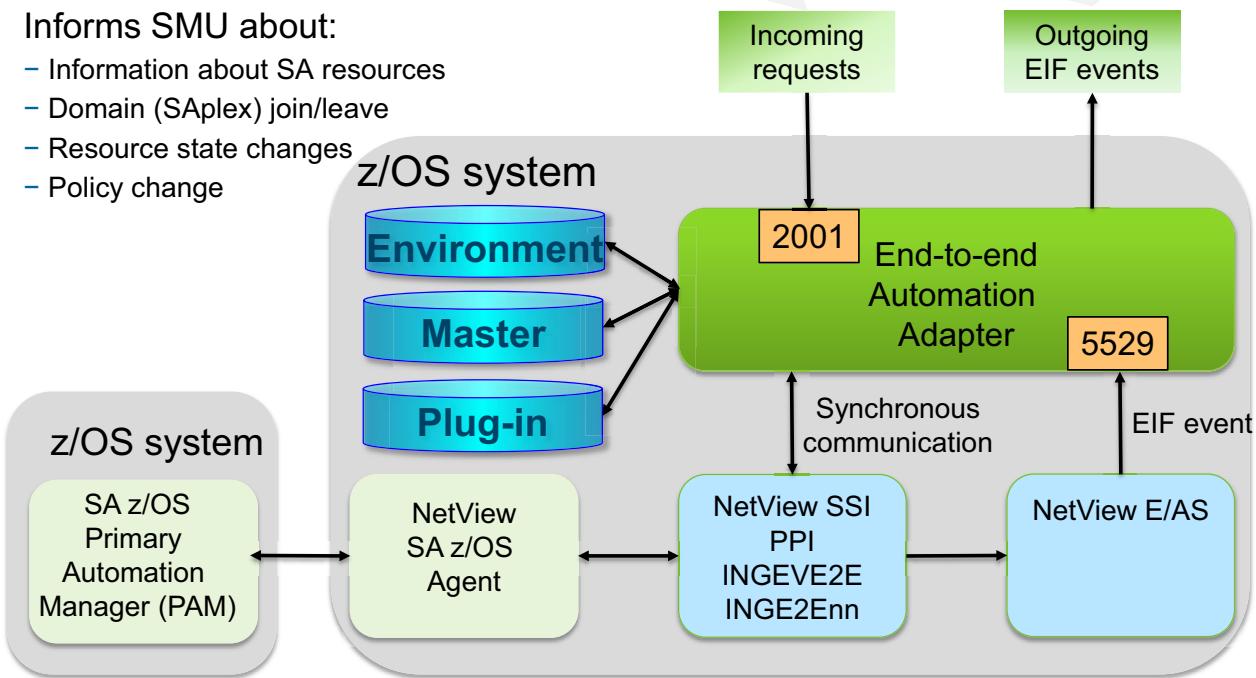
# End-to-end adapter

Request issuer and event receiver can be WebUI (SMU) and E2E agent

Executes requests

Informs SMU about:

- Information about SA resources
- Domain (SAplex) join/leave
- Resource state changes
- Policy change



© Copyright IBM Corporation 2019

9-31

*End-to-end adapter*

The automation adapter connects the automation domain with the SMU server or with the E2E agent. Each SA z/OS automation domain represents an SAplex. It contains one or multiple z/OS systems with System Automation for z/OS and the end-to-end automation adapter. The automation adapter can run on only one system in the SAplex. The system where it runs is automatically selected as the “primary” agent.

This is shown in command INGAMS by YES in column E2E.

The purpose of the automation adapter is to:

- Monitor status of resources within the automation domain.
- Send resource status change events.
- Start and stop resources within the automation domain.
- Provide information about resources in response to queries.

The automation adapter provides the interface between the end-to-end automation manager and SA z/OS as the first level automation. It communicates with the local automation agent on the same system, selected as the primary agent. For synchronous communication it uses the program-to-program interface (PPI), for the asynchronous communication path the NetView event/automation service is used.

## Communication

The automation adapter sends events and receives synchronous requests for execution. Above figure shows that the automation adapter can send events to two different targets depending on customization. The event target is the SMU server and/or the E2E agent.

## Configuration Files

The following configuration files are delivered with the automation adapter:

- “The Automation Adapter Environment Configuration File”  
ing.adapter.properties
- “The Automation Adapter Master Configuration File”  
ing.adapter.properties
- “The Automation Adapter Plug-in Configuration file”  
ing.adapter.plugin.properties
- “The SSL Configuration File”  
ing.adapter.ssl.properties
- “The JAAS Configuration File”  
ing.adapter.jaas.properties

# End-to-end adapter

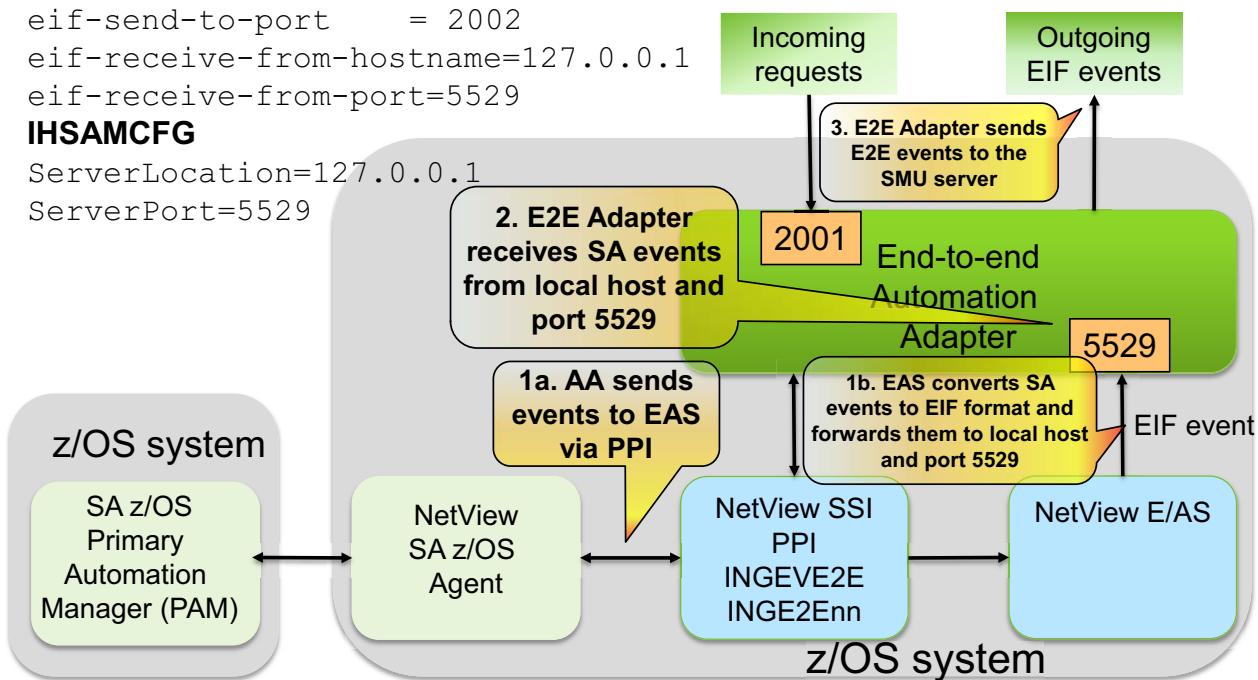
## Events

### ing.adapter.properties

```
eif-send-to-hostname=E2Eserver_host  
eif-send-to-port      = 2002  
eif-receive-from-hostname=127.0.0.1  
eif-receive-from-port=5529
```

### IHSAMCFG

```
ServerLocation=127.0.0.1  
ServerPort=5529
```



© Copyright IBM Corporation 2019

9-32

### End-to-end adapter events

1a. Agent sends events to EAS via PPI

1b. EAS converts SA events to EIF format and forwards them to local host and port 5529

2. E2E Adapter receives SA events from local host and port 5529

3. E2E Adapter sends E2E events to the SMU server

*eif-send-to-hostname*: The host name or IP address of the SMU server. The host name that is to be used by the automation adapter to send EIF events. It must fit the naming convention of the original EIF ServerLocation parameter.

Parameters *eif-send-to-port* in *ing.adapter.properties* must match the port and hostname of the SMU server. The port that is to be used by the automation adapter to send EIF events. It must fit the naming convention of the EIF ServerPort parameter.

*eif-receive-from-hostname*: The host name to be used by the automation adapter to receive EIF events.

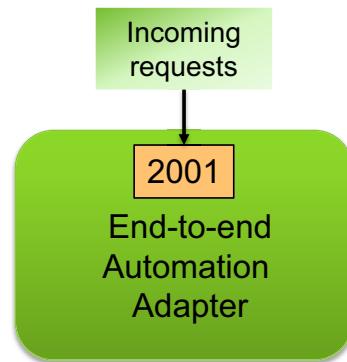
*eif-receive-from-port*: The port that is to be used by the automation adapter to receive EIF events.

The definitions in IHSAMCFG for the E/AS NetView Message Adapter Service must match.

# End-to-end adapter

## Config parameters: Requests

1. eez-remote-contact-port: E2E Adapter listens on port 2001 in order to receive requests and send back response
2. eez-max-connections determines number of requests that can be processed concurrently
3. eez-operator-authentication determines if each request need to be authentified (user/passw)
4. eez-remote-contact-over-ssl Determines if security keys (SSL) are to be used for requests
5. eez-remote-contact-activity-interval-seconds Determines if missing SMU server activity terminates the adapter after certain period of time
6. eez-initial-contact-retry-interval-minutes Determines how long adapter waits for initial contact to SMU server before join-event



© Copyright IBM Corporation 2019

9-33

### End-to-end adapter configuration parameters: Requests

**eez-remote-contact-port:** The port that belongs to the host name that is defined in eez-remote-contact-hostname.

**eez-max-connections:** The number of concurrent working threads. Ideally, this should be the same as the number of NetView end-to-end automation tasks.

**eez-operator-authentication:** If this is set to false, the automation adapter uses the SA z/OS-provided JAAS default login module for operator ID and password validation. otherwise it uses the RACF-provided user ID checking routine.

**eez-remote-contact-over-ssl:** The automation adapter uses SSL (true) or plain (false) socket connection for remote contact to the SMU server or end-to-end agent.

**eez-remote-contact-activityinterval-seconds:** Setting this parameter to 0 means it never expires. max indicates the maximum interval that can be defined in Java.

**eez-initial-contact-retry-interval-minutes:** Zero (0) means never ending. Otherwise the automation adapter stops after the specified number of minutes. The automation adapter attempts to send an AdapterJoinEvent to the server every minute. If the SMU server responds with an acknowledgement request, the automation adapter continues with normal initialization.

# End-to-end adapter

## Customization steps

- Customize SA and Netview
- Customize adapter USS shell script
  - Modify adapter installation path and optionally other parameters
- Customize Adapter Configuration Files
  - Adapter Master Configuration File
  - SA z/OS specific Configuration File
  - SSL, JAAS and Logger Configuration Files
- Define Security
  - Authentication of User
  - Authorization of Request
- Enable Adapter High Availability
  - Define adapter as SA resource and define end-to-end move group
  - Start/Stop adapter via SA
  - See \*E2E AddOn PDB and manual for details

9-34

### *End-to-end adapter customization steps*

- Customize SA and Netview
- Customize adapter USS shell script
  - Modify adapter installation path and optionally other parameters
- Customize Adapter Configuration Files
  - Adapter Master Configuration File
  - SA z/OS specific Configuration File
  - SSL, JAAS and Logger Configuration Files
- Define Security
  - Authentication of User
  - Authorization of Request
- Enable Adapter High Availability using \*E2E AddOn policy
  - Define adapter as SA resource and define end-to-end move group
  - Start/Stop adapter via SA

## Enable NetView Event Automation Service (E/AS) Manual setup without SA Configuration Assistant

- Run EAS address space of its own that forwards SA events to e2e adapter
  - Sample job: NETVIEW.V5R1M0.SCNMUXMS( IHSAEVNT )
- EAS global initialization file
  - DD IHSSMP3: Default member name is IHSAINIT  
You may overwrite default via proc parameter INITFILE=newname
  - Specify PPI receive id: PPI=INGEVE2E
    - Overwriting default requires to setup AAO: AOF\_E2E\_EAS\_PPI
  - Start EAS message adapter
    - remove statement: NOSTART TASK=MESSAGEA
  - NO other EAS services need to be started (disable starting any other tasks)
- EAS message adapter configuration file
  - DD IHSSMP3: Default member name is IHSAMCFG  
You may overwrite default via proc parameter MSGCFG=newname
  - Specify event target IP address 127.0.0.1 and port 5529
  - Specify message adapter format file
    - Copy member ING.SINGSAMP( INGMFMTE ) to DD IHSSMP3
  - Specify some more event related parameters such as
    - connection\_oriented, switch off caching and set max event size to 4096

9-35

### *Enable NetView Event Automation Service (E/AS)*

The end-to-end automation adapter uses the NetView message adapter service of E/AS as a gateway between the primary system automation agent and the end-to-end automation adapter. The NetView message adapter service of E/AS converts incoming messages from NetView into EIF events and forwards them to the automation adapter.

Setup with SA Configuration Assistant is recommended.

## Customize SA and NetView

- Enable PPI for SA Communication task in DSIPARM INGXINIT
  - PPI=YES
  - PPIBQL=3000
- Use SA customization dialog to define automated operator functions dedicated for end-to-end automation
  - Execute requests
    - E2EOPER AUTE2E
    - E2EOPR01 ... E2EOPR99 AUTE2E01 .. AUTE2E99
  - Forward events
    - EVTOPER AUTEVT1 + AUTEVT2
- Corresponding NetView tasks are already defined in SA sample ING.SINGSAMP(AOFOPFSO)

9-36

### Customize SA and NetView

Enable the PPI for the SA Communication task in DSIPARM INGXINIT and in the NetView SSI.

Use SA customization dialog to define automated operator functions dedicated for end-to-end automation using \*E2E AddOn policy.

Corresponding NetView tasks are already defined in SA sample ING.SINGSAMP(AOFOPFSO).

# End-to-end adapter

## Successful start

```
STATMON.BROWSE      ACTP   NETWORK LOG FOR 06/11/19 (19162) COLS 035 112 18:58 A
                                              DOMAIN: AOFDB SCROLL ==> CSR
+---4---+---5---+---6---+---7---+---8---+---9---+---10---+---11
E $HASP100 BPXAS    ON STCINRDR
E $HASP373 BPXAS    STARTED
E IEF403I BPXAS - STARTED - TIME=18.57.19
E INGX9706I Trying to drop possible open, unused connections. Processing port:
C BLOG
" EEEZC0003I Base output directory for serviceability related files (for
" example, message log files and trace files) has been set to /MVSBU/e2
" e/adapter/data/MVSB/eez/*"
- CNM493I INGMSG02 : 00002114 : MSGID=EEZA0100I : ACTIVMSG UP=NO
E EEZA0100I The adapter has been started*
- CNM493I INGMSG02 : 00002118 : MSGID=EEZA0101I : AOFCPMSG
E EEZA0101I The adapter is active*
- CNM493I INGMSG02 : 00002138 : MSGID=EEZA0111I : AOFCPMSG
E EEEZA0111I The plug-in is starting: class com.ibm.ing.sam.INGXPlugin *
" INGX9802I INGXLogger has successfully been initialized using configura
" tion file ing.adapter.jlog.properties from path /usr/lpp/ing/adapter/c
" onfig.*
" INGX9902I INGXPluginLogger has successfully been initialized using con
" figuration file ing.adapter.jlog.properties from path /usr/lpp/ing/ada
" pter/config.*
- CNM493I INGMSG02 : 00002142 : MSGID=EEZA0112I : ACTIVMSG UP=YES
E EEEZA0112I The plug-in has been started: class com.ibm.ing.sam.INGXPlugin *
- CNM493I INGMSG02 : 00002122 : MSGID=EEZA0102I : AOFCPMSG
E EEEZA0102I The adapter is ready*
E EEZA0116I The status of the event sender changed: Address=/10.31.189.184, Po
E EEZA0118I The connection to the management server 10.31.189.184 : 2003 has b
```

ACTIVE  
message

Captured  
messages

UP  
message

Connection to  
remote E2E  
agent  
established

© Copyright IBM Corporation 2019

2-37

*End-to-end adapter: Successful start*

Connection to remote E2E agent established!

## INGE2E command provides the operational API

| Parameters            | Adapter | Agent | Description   |
|-----------------------|---------|-------|---|
| <b>RESET</b>          | X       |       | <b>Clean up</b> in case of normal (IEF404I) or abnormal (IEF450I) termination   |
| <b>RESYNC</b>         |         | X     | <b>Resynchronize</b> event notification with the first-level automation   |
| <b>RECYCLE</b>        | X       |       | <b>Recycle</b> adapter MOVE group   |
| <b>SETDBGLVL</b>      | X       |       | Set ON <b>debug</b> level for the adapter REXX backend  |
| <b>VERIFY</b>         | X       |       | CPATH=custom_path JOBEAS=jobname - <b>Verifies the installation</b>   |
| <b>PING</b>           | X       |       | <b>Ping hostnames</b> specified in the adapter configuration file   |
| <b>DISPCFG</b>        | X       |       | Display <b>configuration files</b> from the custom-root directory CPATH   |
| <b>PREPARE</b>        | X       |       | IPATH, CPATH, SERVER, JAVA<br><b>Prepares the USS configuration files</b>   |
| <b>STOP_E2EAGT</b>    |         | X     | <b>Stop</b> the E2E agent in a normal manner. This command is used in the SA automation policy. It's normally not issued manually by operators                |
| <b>CLEANUP_E2EAGT</b> |         | X     | <b>Cleans up the E2E agent</b> when the E2E agent is not terminated normally. It deregisters the E2E agent from the automation manager and cleans up CGlobals |
| <b>DEBUG_E2EAGT</b>   |         | X     | Switches on <b>debugging</b> the E2E agent for the REXX trace and for the Java logger   |

9-38

### INGE2E command provides the operational API

The INGE2E command provides the operational API for the end-to-end automation adapter and the end-to-end agent. It can be issued on the NetView console. Some commands, like RESET or STOP\_E2EAGT are used in the SA automation policy.

The INGE2E command can also prepare the USS configuration files for the end-to-end automation adapter and completes the following:

1. creates necessary subdirectories within the custom-root directory
2. Copies all necessary configuration files from the install-root to the custom-root
3. Updates the configuration files with the input parameters.

VERIFY verifies the installation of the end-to-end automation adapter. Requires that you have copied and customized the end-to-end automation adapter configuration files.

You have to specify the jobname of the E/AS address space and the customization path of the end-to-end automation adapter.

# Lesson 5. End-to-end automation policy

## Lesson 5: End-to-end automation policy

- \*E2E Best Practices Policy
- DMN and REF automation policy

### What this lesson is about

This lesson covers the \*E2E Best Practices Policy and how to define end-to-end automation policy.

### Objectives

After completing this lesson, you should be able to define end-to-end automation policy.

### References

*SC34-2750-02 System Automation for z/OS Version 4  
Release 1 End-to-End Automation*  
*SC27-8747-00 System Automation for z/OS Version 4  
Release 1 Service Management Unite Automation  
Installation and Configuration Guide*

## \*E2E Best Practices Policy

| Entries of selected Add-on Policies |                  |              |               |  | Row 1 to 15 of 15                                      |
|-------------------------------------|------------------|--------------|---------------|--|--|
|                                     |                  |              |               |  | To be used for E2E agent<br>To be used for E2E adapter |
| 1 Modify data                       | 2 Check data     | 3 Run import | 4 View report |  |  |
| Action                              | Entry Name       | Type         | C D           | Short Description                      |  |
|                                     | SYSPLEX1         | GRP          |               | Placeholder. Original in *BASE         |  |
|                                     | SYS1             | SYS          |               | Placeholder. Original defined in *BASE |  |
|                                     | SYS2             | SYS          |               | Placeholder. Original defined in *BASE |  |
|                                     | SYS3             | SYS          |               | Placeholder. Original defined in *BASE |  |
|                                     | E2EADPT          | APG          |               | E2E Automation Adapter with E/AS       |  |
|                                     | E2EADPT_X        | APG          |               | E2E Automation Adapter Move Group      |  |
|                                     | E2EAGENT         | APG          |               | E2E Automation Agent with DVIPA        |  |
|                                     | E2EAGENT_X       | APG          | *             | E2E Automation Agent Move Group        |  |
|                                     | C DVIPA          | APL          | *             | Placeholder. Original defined in *BASE |  |
|                                     | E2EADPT          | APL          |               | E2E Automation Adapter                 |  |
|                                     | E2EAGENT         | APL          |               | E2E Automation Agent                   |  |
|                                     | E2EDVIPA         | APL          |               | Dynamic Virtual IP Address             |  |
|                                     | E2EEAS           | APL          |               | Event Automation Service               |  |
|                                     | E2EADPT_AUTOOPS  | AOP          |               | Automated Functions for E2E Adapter    |  |
|                                     | E2EAGENT_AUTOOPS | AOP          |               | Automated Functions for E2E Agent      |  |

© Copyright IBM Corporation 2019

9-40

### \*E2E Best Practices Policy

Import the Automation Operators (AOP) entry E2EADPT\_AUTOOPS (Automated Functions for E2E) from the \*E2E best practices policy. Link it to the systems where you want to start the end-to-end automation adapter.

Import the Automation Operators (AOP) entry E2EAGENT\_AUTOOPS (Automated Functions for E2E agent) from the \*E2E best practices policy. Link it to the systems where you want to start the end-to-end agent.

After having customized the automation policy, the BUILD process has to be executed to create the control files, and the configuration has to be refreshed with the INGAMS command.

## Summary DMN and REF resources

Any resource outside of the local SAplex requires a REF resource:

- Resource reference: name/REF/dmnnname
- 11 char name of your choice, 8 char (short) DMN name
- Remote resource key, e.g. child2/APL/SYS1
- Can be a member of sysplex groups and can have relationships
- Link to one DMN resource

Any remote SAplex or non SA domain requires a DMN resource:

- Automation domain: dmname/DMN
- 8 char (short) DMN name, 64 char long remote automation domain name
  - It is originally defined in the E2E adapter configuration file
  - Example for a default SA automation domain name: "PLEX12 INGXSGSB" if PLEX12 is the sysplex name and SB is the SA XCF group suffix

### Summary DMN and REF resources

Any resource outside of the local SAplex requires a REF resource implemented by the Resource References entry type. A REF resource can be a member of sysplex groups and can have relationships. You can link a REF resource only to one DMN resource.

Any remote SAplex or non SA domain requires a DMN resource implemented by the Remote Domains entry type.

The name of the remote domain must match the name in the E2E adapter configuration. For SA z/OS subplexes, the default is plexname INGXSGid, where "plexname" is the name of the physical sysplex and "id" is the XCF group ID which might be blank.

## Define E2E Policy: DMN Remote Domains entry type

```

COMMANDS ACTIONS VIEW HELP
Entry Name Selection
Command ==> new mvsb Scroll ==> CSR
No entries currently exist. Use the NEW command to create an entry.
Entry Type : Remote Domain PolicyDB Name : E2E
Enterprise Name : E2E

Action Entry Name Short Description
***** Bottom of data *****
COMMANDS HELP
AOFGDYN2 Define New Entry
Command ==>
Define new entry of type Remote Domain
Entry name . . . . . PLEX34
Domain Name . . . . .
PLEX34 INGXSGSB
Short description . . . System MVSB in SAplex SYSPLEX2
Long description 1 . . . MVSB is cloned to multiple physical sysplexes
Long description 2 . . . PLEXab corresponds to VM guests:
Long description 3 . . . a (odd): system MVSA
Long description 4 . . . b (even): system MVSB
Long description 5 . . .

```

© Copyright IBM Corporation 2019

9-42

### Define E2E Policy: DMN Remote Domains entry type

Definition of a remote domains entry type is similar to other entry types. Select option 50 in panel AOFGETYP and enter the NEW command followed by the entry type name.

You then have to specify a Domain Name which must match the name in the E2E adapter configuration of the remote domain.

# DMN Remote Domains entry type

## Policies and linkage to SAplex

ACTIONS HELP

| Policy Selection           |                       |   | Entry created<br>Scroll ==> CSR |
|----------------------------|-----------------------|---|---------------------------------|
| Command ==> [ ]            |                       |   |                                 |
| Entry Type : Remote Domain | PolicyDB Name : E2E   |   |                                 |
| Entry Name : MVSB          | Enterprise Name : E2E |   |                                 |
| Action                     | Policy Name           | Policy Description                      |                                 |
|                            | DESCRIPTION           | Enter description                       |                                 |
|                            | DOMAIN INFO           | Define domain information               |                                 |
|                            | REFERENCES            | Select references to remote resources   |                                 |
|                            | WHERE USED            | List sysplex group linked to this entry |                                 |
|                            | COPY                  | Copy data from an existing entry        |                                 |
| ***** Bottom of data ***** |                       |   |                                 |

Available Policy Selection

COMMANDS ACTIONS VIEW HELP

| Where Used                 |                       |          | Row 1 to 2 of 2<br>SCROLL ==> CSR |
|----------------------------|-----------------------|----------|-----------------------------------|
| Command ==> [ ]            |                       |          |                                   |
| Entry Type : Remote Domain | PolicyDB Name : E2E   |          |                                   |
| Entry Name : MVSB          | Enterprise Name : E2E |          |                                   |
| Action                     | Status                | Name     | Type                              |
|                            | SELECTED              | SYSPLEX1 | GRP                               |
|                            |                       | SYSPLEX2 | GRP                               |
| ***** Bottom of data ***** |                       |          |                                   |

Link to SAplex  
SYSPLEX2

© Copyright IBM Corporation 2019

9-43

DMN Remote Domains entry type: Policies and linkage to SAplex

## DOMAIN INFO

You can use this policy item to define a remote domain (for example a z/OS sysplex) where resources are located which are in the scope for end-to-end automation.

## REFERENCES

You can use this policy item to select resource references which represent resources that are located on a remote domain.

## Define E2E Policy: REF Resource Reference entry type

**COMMANDS ACTIONS VIEW HELP**

**Entry Name Selection**

Command ==> new rv02      Scroll ==> CSR

No entries currently exist. Use the NEW command to create an entry.

Entry Type : Resource Reference      PolicyDB Name : E2E  
Enterprise Name : E2E

| Action                     | Entry Name | Short Description |
|----------------------------|------------|-------------------|
| ***** Bottom of data ***** |            |                   |

**Define New Entry**

Command ==>

Define new entry of type Resource Reference

Entry Name . . . . . rv02

Specify the remote resource:

Name . . . . . rv02

Type . . . . . apl      (APG APL MTR REF  
IBM.RemoteApplication)

System/Node . . . . . mvsb

Short Description . . . Reference to RV02 in MVSB  
Long Description 1 . . . Reference to RV02 in SAplex SYSPLEX2

9-44

Define E2E Policy: REF Resource Reference entry type

This entry type defines a reference to a resource running on a remote domain (for example a z/OS sysplex). A resource reference refers to a remote resource, that is, a resource defined on a remote domain that is different from the domain, where the resource reference itself is defined. Definition of a Resource Reference entry type is similar to other entry types. Select option 51 in panel AOFGETYP and enter the NEW command followed by the entry type name.

**Name** is the name of the remote resource. It must match the remote resource name. For non SA z/OS remote resources this is the value of 'Resource name' as specified in the Universal Automation Adapter policy.

**Type** is the type of the remote resource.

For SA z/OS remote resources this is typically APL for an application, APG for an application group, MTR for a monitor resource, or REF for a resource reference.

For non SA z/OS remote resources *IBM.RemoteApplication* needs to be specified.

**System/Node** is the name of the system or node where the remote resource is running.

For SA z/OS remote resources, the system where the remote resource is running is specified. Leave this field blank if the remote resource is a sysplex wide group. For non SA z/OS remote resources this is the value of 'node' as specified in the Universal Automation Adapter policy.

# REF Resource Reference entry type

## Policies and linkage to a domain

ACTIONS HELP

| Policy Selection                |                 |                                  | Entry created<br>Scroll ==> CSR   |
|---------------------------------|-----------------|----------------------------------|---|
| Command ==> _____               |                 |                                  |   |
| Entry Type : Resource Reference |                 | PolicyDB Name : E2E              |   |
| Entry Name : RV02               |                 | Enterprise Name : E2E            |   |
| Action                          | Policy Name     | Policy Description               | Available Policy Selection<br>Only relationships, no start or stop policy |
| _____                           | DESCRIPTION     | Enter description                |   |
| _____                           | REMOTE RESOURCE | Define remote resource           |   |
| _____                           | RELATIONSHIPS   | Define relationships             |   |
| _____                           | WHERE USED      | List domain linked to this entry |   |
| _____                           | COPY            | Copy data from an existing entry |   |
| ***** Bottom of data *****      |                 |                                  |   |
| AOFGXWHU Where Used             |                 | Row 1 to 3 of 3<br>SCROLL==> CSR |   |
| Command ==> _____               |                 |                                  |   |
| Entry Type : Resource Reference |                 | PolicyDB Name : E2E              |   |
| Entry Name : RV02               |                 | Enterprise Name : E2E            |   |
| Action                          | Status          | Name                             | Type  |
| _____                           | SELECTED        | PLEX12                           | DMN   |
| _____                           |                 | PLEX34                           | DMN   |
| _____                           |                 | PLEX56                           | DMN   |

© Copyright IBM Corporation 2019

9-45

Define E2E Policy: REF Resource Reference entry type

## REMOTE RESOURCE

You can use this policy item to define the remote resource, which is the 'real' name of the resource on the remote domain (for example a z/OS sysplex). In addition you can define similar to entry type Application and others:

- Inform List,
- Owner
- Info Link

Don't use system symbols or system automation symbols in any of the fields.

## RELATIONSHIPS

This policy item has the same functionality as for entry type Application and others.

### Where Used

You can link a Resource Reference only to a single remote domain and not to APGs, however APGs can select REFs as members.

# Lesson 6. End-to-end user interface

## Lesson 6: End-to-end user interface

- SA / NetView
- SMU

### What this lesson is about

This lesson provides details about the end-to-end automation user interfaces in SA / NetView and SMU.

### Objectives

After completing this lesson, you should be able to the end-to-end automation user interfaces.

### References

*SC34-2750-02 System Automation for z/OS Version 4 Release 1 End-to-End Automation*  
*SC27-8747-00 System Automation for z/OS Version 4 Release 1 Service Management Unite Automation Installation and Configuration Guide*

# End-to-end automation

## User Interface

### SA commands

- INGAMS
- INGLIST
- INGINFO
- INGREQ
- INGSET

| Command  | REF | DMN | Description   |
|----------|-----|-----|---|
| INGLIST  | X   | X   | Display automation domains and resource references                            |
| INGINFO  | X   | X   | Display detailed information about automation domains and resource references |
| INGREQ   | X   |     | Start and stop resource references  |
| INGSET   | X   |     | Set status of a resource reference  |
| INGSCHED | X   |     | Creates a schedule override for a resource reference                          |

### SMU Web Interface

- REF
- E2E automation domain
- Cross domain relationships

#### *End-to-end automation User Interface*

End-to-end resources can be managed with many of the same SA commands that manage applications or application groups.

INGLIST displays DMN and REF resources.

REF resources can be started or stopped with INGREQ.

INGSET sets the status of a resource reference.

Although you can not link a resource reference to a service period, you can use INGSCHED to create a schedule override for a resource reference.

## INGLIST showing DMN and REF resources

| INGKYST0    |           | SA z/OS - Command Dialogs |              |             |           |           |             | Line 25 of 46 |            |           |          |            |            |  |  |  |  |  |
|-------------|-----------|---------------------------|--------------|-------------|-----------|-----------|-------------|---------------|------------|-----------|----------|------------|------------|--|--|--|--|--|
| Domain Id . | : AOFDA   | INGLIST                   |              |             |           |           |             | Date . . .    | : 06/11/19 |           |          |            |            |  |  |  |  |  |
| Operator Id | : INGC102 | Sysplex = SYPLEX1         |              |             |           |           |             | Time . . .    | : 18:59:38 |           |          |            |            |  |  |  |  |  |
| A Update    | B Start   | C Stop                    | D INGRELS    | E INGVOTE   | F INGINFO | G Members | H DISPTRG   | I INGSCHED    | J INGGROUP | K INGCICS | L INGIMS | M DISPMTTR | P INGPAC   |  |  |  |  |  |
| R Resume    | S Suspend | T INGTWS                  | U User       | X INGWHY    | / scroll  | Observed  | CMD Name    | Type          | System     | Sus       | Compound | Desired    | Nature     |  |  |  |  |  |
| OMVS        | APL       | MVSA                      | SATISFACTORY | AVAILABLE   | AVAILABLE | AVAILABLE | PLEX12      | DMN           |            |           |          |            | SYSGONE    |  |  |  |  |  |
| PLEX12      | DMN       |                           | INHIBITED    | AVAILABLE   | AVAILABLE | AVAILABLE | PLEX56      | DMN           |            |           |          |            | AVAILABLE  |  |  |  |  |  |
| PLEX56      | DMN       |                           | SATISFACTORY | AVAILABLE   | AVAILABLE | AVAILABLE | RACF        | APL           | MVSA       |           |          |            | AVAILABLE  |  |  |  |  |  |
| RACF        | APL       | MVSA                      | SATISFACTORY | AVAILABLE   | AVAILABLE | AVAILABLE | RESOLVER    | APL           | MVSA       |           |          |            | AVAILABLE  |  |  |  |  |  |
| RMF         | APL       | MVSA                      | SATISFACTORY | AVAILABLE   | AVAILABLE | AVAILABLE | RMFGAT      | APL           | MVSA       |           |          |            | AVAILABLE  |  |  |  |  |  |
| RMFGAT      | APL       | MVSA                      | SATISFACTORY | AVAILABLE   | AVAILABLE | AVAILABLE | RV_E2E_MOVE | APG           |            |           |          |            | AVAILABLE  |  |  |  |  |  |
| RV_E2E_MOVE | APG       |                           | SATISFACTORY | AVAILABLE   | AVAILABLE | AVAILABLE | RVBASICA    | APG           | MVSA       |           |          |            | MOVE BASIC |  |  |  |  |  |
| RVBASICA    | APG       | MVSA                      | SATISFACTORY | AVAILABLE   | AVAILABLE | AVAILABLE | RV01        | APL           | MVSA       |           |          |            | AVAILABLE  |  |  |  |  |  |
| RV01        | APL       | MVSA                      | SATISFACTORY | AVAILABLE   | AVAILABLE | AVAILABLE | RV02        | REF           | PLEX56     |           |          |            | SOFTDOWN   |  |  |  |  |  |
| RV02        | REF       | PLEX56                    | SATISFACTORY | UNAVAILABLE | SOFTDOWN  | SOFTDOWN  | SDSF        | APL           | MVSA       |           |          |            | AVAILABLE  |  |  |  |  |  |
| SDSF        | APL       | MVSA                      | SATISFACTORY | AVAILABLE   | AVAILABLE | AVAILABLE | Command ==> |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |
|             |           |                           |              |             |           |           |             |               |            |           |          |            |            |  |  |  |  |  |

# INGLIST showing DMN and REF resources

## Scrolled to the right

| SA z/OS - Command Dialogs |             |               |                                    | Line 25    | of 46               |
|---------------------------|-------------|---------------|------------------------------------|------------|---------------------|
| INGLIST                   |             |               |                                    | Date . . . | : 06/11/19          |
| Sysplex = SYSPLEX1        |             |               |                                    | Time . . . | : 21:20:15          |
| A Update                  | B Start     | C Stop        | D INGRELS                          | E INGVOTE  | F INGINFO G Members |
| H DISPTRG                 | I INGSCHED  | J INGGROUP    | K INGCICS                          | L INGIMS   | M DISPMTR P INGPAC  |
| R Resume                  | S Suspend   | T INGTWS      | U User                             | X INGWHY   | / scroll            |
| <b>CMD Name</b>           | <b>Type</b> | <b>System</b> | <b>Description</b>                 |            |                     |
| OMVS                      | APL         | MVSA          | Unix System Services subsystem     |            |                     |
| PLEX12                    | DMN         |               | System MVSB in SAplex SYSPLEX2     |            |                     |
| PLEX56                    | DMN         |               | System MVSB in SAplex SYSPLEX2     |            |                     |
| RACF                      | APL         | MVSA          | Resource Access Control Facility   |            |                     |
| RESOLVER                  | APL         | MVSA          | TCP/IP Name Resolver               |            |                     |
| RMF                       | APL         | MVSA          | Resource Measurement Facility      |            |                     |
| RMFGAT                    | APL         | MVSA          | RMF Monitor III Data Gatherer      |            |                     |
| RV_E2E_MOVE               | APG         |               | e2e Move Group for RV01 and RV02   |            |                     |
| RVBASICA                  | APG         | MVSA          | RV Apps APG for system A           |            |                     |
| RV01                      | APL         | MVSA          | RV01 only in system MVSA           |            |                     |
| RV02                      | REF         | PLEX56        | Reference to RV02 in MVSB          |            |                     |
| SDSF                      | APL         | MVSA          | System Display and Search Facility |            |                     |

Command ==> ■

F1=Help F2=End F3=Return F4=DISPSTAT F5=Filters F6=Roll  
 F7=Backward F8=Forward F9=Refresh F10=Previous F11=Next F12=Retrieve

System MVSB in 2 physical sysplexes

Sysplex move group used as e2e group

RV02/REF/MVSB in other SAplex

INGLIST showing DMN and REF resources. Scrolled to the right

If you scroll INGLIST to the right you can read the Descriptions.

## INGINFO showing remote domain

```
INGKYINO          SA z/OS - Command Dialogs      Line 07   of 139
Domain Id . : AOFDA  ----- INGINFO -----  Date . . : 05/27/19
Operator Id : INGC102        Sysplex = SYSPLEX1  Time . . : 19:37:01

Resource    => MVSB/DMN           format: name/type/system
Target      => _____ System name, domain ID or sysplex name

Automation     : IDLE
Startability    : INHIBITED
Compound        : INHIBITED      Last changed : 2019-05-27 13:47:01
Health Status   : N/A

E2E Agent...
Runs on         : MVSA

Remote Domain...
Name            : MVSB
Operational Info : DOMAIN_HAS_NEVER_JOINED - The domain has never sent a
                   join event. Therefore the connection data is not
                   available for this domain.

Command ==> [ ] F1=Help F2=End F3=Return F4=INGDMN F6=Roll
F7=Backward F8=Forward F9=Refresh F12=Retrieve
```

A yellow callout box points to the 'Operational Info' line, highlighting the text 'The domain has never sent a join event. Therefore the connection data is not available for this domain.' The box contains the text 'remote domain never joined'.

© Copyright IBM Corporation 2019

9-50

### INGINFO showing remote domain

INGINFO can show remote domains and provides operational information.

Here the remote domain never joined.

## INGINFO showing connected remote domain

```

INGKYINO          SA z/OS - Command Dialogs      Line 11   of 129
Domain Id . : AOFDA    ----- INGINFO ----- Date . . : 06/11/19
Operator Id : INGC102      Sysplex = SYSPLEX1     Time . . : 19:01:20

Resource      => PLEX56/DMN      format: name/type/system
Target        => _____      System name, domain ID or sysplex name

E2E Agent...
Runs on       : MVSA

Remote Domain...
Name          : PLEX56 INGXSGSB
Operational Info : OK - The connection to the domain was successfully
                   established.

Adapter OS      : z/OS
Adapter Hostname : mvsch26.ilsvpn.ibm.com
Adapter Port     : 2001
Adapter SSL      : false
Adapter Security : true

Command ==> ■
F1=Help F2=End F3=Return F4=INGDMN F6=Roll
F7=Backward F8=Forward F9=Refresh F12=Retrieve

```

**Remote adapter hostname, port, SSL, and security**

**Press F4 for INGDMN**

**Connection to remote domain in sysplex PLEX56 established**

© Copyright IBM Corporation 2019

9-51

INGINFO showing connected remote domain

INGINFO can show remote domains and provides operational information.

Here the remote domain is connected.

Press F4 to jump to the INGDMN panel.

## E2E agent can not connect to remote E2E adapter

```
INGKE2E1          SA z/OS - Command Dialogs      Line 1    of 5
Domain Id . : AOFDA ----- INGDMN ----- Date . . : 05/27/19
Operator Id : INGC102       System = MVSA           Time . . : 19:34:41
                                         Group ID= INGXSGSA
Resource          : MVSB/DMN
Automation Domain : MVSB
```

### Diagnostic Error Information

```
Failing command: INGRE2E GetDomain TYPE(PROPERTIES2) ANYWHERE(YES) DMN(MVSB)
Return Code      : 233
```

```
ING377I The E2E agent service INGRE2E failed with RC=233, REASON=No persistent
ING376I The E2E agent cannot connect to MVSB/DMN because the remote E2E adapt
```

```
INGKE2E1          SA z/OS - Command Dialogs      Line 1    of 5
Domain Id . : AOFDA ----- INGDMN ----- Date . . : 05/27/19
Operator Id : INGC102       System = MVSA           Time . . : 19:35:44
                                         Group ID= INGXSGSA
Resource          : MVSB/DMN
Automation Domain : MVSB
```

### Diagnostic Error Information

Remote E2E  
adapter not  
running.  
Press F11

```
t data available for domain
er is not running or not recycled
```

© Copyright IBM Corporation 2019

9-52

*E2E agent can not connect to remote E2E adapter*

Here the E2E agent can not connect to remote E2E adapter because the remote E2E adapter not running.

## INGDMN showing remote domain

```

INGKE2E1          SA z/OS - Command Dialogs      Line 1   of 23
Domain Id . : AOFDA    ----- INGDMN -----  Date . . : 06/11/19 .
Operator Id : INGC102   System = MVSA           Time . . : 19:02:33
                                         Group ID= INGXSGSA

Resource          : PLEX56/DMN
Automation Domain : PLEX56 INGXSGSB

Additional Domain Properties

Adapter Custom Directory      /MVS/.../e2e/adapter/
Adapter Data Directory        /MVS/.../e2e/adapter/data/MVS
Adapter Install Directory     /usr/lpp/ing/adapter
Adapter Java Runtime Version  1.6.0 j9jit24
Adapter Operating System      z/OS 02.02.00
Adapter Plugin Build Version SAZOS.4.1.0.20180518.1120, 0A51668,0A52358,0
Adapter Plugin Startup Time  2019-06-11 18:57:23 +0000
Adapter Startup Time         2019-06-11 18:57:22 +0000
Policy Activation Time       Tue Jun 11 18:54:43 UTC 2019
Policy Filename               SAZOS.ACFS.E2E

Primary Agent Name           MVS
Primary Agent NetView Domain AOFDB
Primary Agent NetView Version V6R2M1
Primary Agent UTC Offset     +0000
Primary Agent Version        V4R1M0
Primary Manager Name         MVS$$$$
Primary Manager Start Type   COLD
Primary Manager Start Up Time 2019-06-10 19:57:18 +0000
Primary Manager System Name  MVS
Sysplex Name                 PLEX56
Saplex Name                  SYSPLEX2
SMU Host                     SMUserver_host:2002

```

The primary agent runs the e2e agent and makes the AM an e2e AM

Remote adapter info

© Copyright IBM Corporation 2019

9-53

### INGDMN showing remote domain

INGDMN can show remote domain information.

The data displayed refers to the E2E automation adapter or the universal automation adapter for this remote automation domain. The list of additional domain properties may vary depending on the version of the remote E2E adapter or universal automation adapter.

If the connection was successfully established the panel displays "Additional Domain Properties".

If the connection cannot be established the panel displays "Diagnostic Error Information".

## Use INGMOVE to “move” member from MVSA into sysplex PLEX56

The screenshots illustrate the use of the INGMOVE command to move a member from system MVSA to domain PLEX56.

**Screenshot 1:** Shows the initial state where the member 'MVSA' is selected in the 'Members' column. A callout box indicates 'sysplex move group has selected member in MVSA'.

| Command Dialogs |                    | Line 1 of 1           |           |                     |           |            |
|-----------------|--------------------|-----------------------|-----------|---------------------|-----------|------------|
| INGMOVE         | Sysplex = SYSPLEX1 | Date . . . : 06/11/19 |           |                     |           |            |
|                 |                    | Time . . . : 19:09:29 |           |                     |           |            |
| B Start         | C Stop             | D INGRELS             | E INGVOTE | F INGINFO           | G Members | I INGSCHED |
| J INGGROUP      | M Move             | P Prepare             | R Reset   | X Box Group         |           | / scroll   |
| Cmd             | Group name         | Obs                   | Status    | ----- Systems ----- |           |            |
| RV_E2E_MOVE     |                    | AVAILABLE             | MVSA      | PLEX56              | PLEX56    |            |

**Screenshot 2:** Shows the 'Move to' field set to 'PLEX56'. A callout box indicates 'Member in domain PLEX56 selectable'.

**Screenshot 3:** Shows the member 'MVSA' moved to domain 'PLEX56'. A callout box indicates 'Verification. Press F10 to GO'.

© Copyright IBM Corporation 2019

9-54

Use INGMOVE to “move” member from MVSA into sysplex PLEX56

A sysplex move group is used as e2e group. It has RV01 in system MVSA and RV02 in domain PLEX56 as members.

The sysplex move group member is moved from MVSA to domain PLEX56.

# REF resources on SMU

The image displays two separate instances of the SMU Domain Page. Both instances show a table of resources with columns for Name, Compound State, Observed State, Desired State, Automated status, and Operator Request.

**Top Screenshot (Automation Domain: AOC4PLEX INGXSGA4):**

| Name                  | Compound State | Observed State | Desired State | Automated | Operator Request |
|-----------------------|----------------|----------------|---------------|-----------|------------------|
| CSM_AOCA/REF/AOCAPLEX | OK             | Available      | Available     | Yes       | No request       |
| CSM_AOCB/REF/AOCAPLEX | Warning        | Unknown        | Available     | Suspended | No request       |
| CSM_AOCC/REF/AOCAPLEX | OK             | Available      | Available     | Yes       | No request       |
| CSM_AOCD/REF/AOCAPLEX | Warning        | Unknown        | Unavailable   | Yes       | Stop             |

**Bottom Screenshot (Automation Domain: AOCAPLEX INGXSGA0):**

| Name             | Compound State | Observed State | Desired State | Automated | Operator Request |
|------------------|----------------|----------------|---------------|-----------|------------------|
| CSM_GRP/APG/AOCB | Warning        | Unknown        | Available     | Yes       | No request       |
| CSM_GRP/APG/AOCC | OK             | Available      | Available     | Yes       | No request       |
| CSM_GRP/APG/AOCD | Warning        | Unknown        | Available     | Yes       | No request       |

© Copyright IBM Corporation 2019

9-55

## REF resources on SMU

SMU has full support of resource references.

If you select “View Remote Resources” resource references are displayed and can be managed as other resources.

# Lesson 7. End-to-end automation scenarios

## Lesson 7: End-to-end automation scenarios

- Cross sysplex automation examples
  - Monitor an end-to-end application
  - Db2, IMS, and WebSphere in separate sysplexes
  - Log management with Beta 92
  - Provider – Clients setup
- Extending automation scope from sysplexes to distributed platforms & cloud applications

### What this lesson is about

This lesson describes end-to-end automation scenarios.

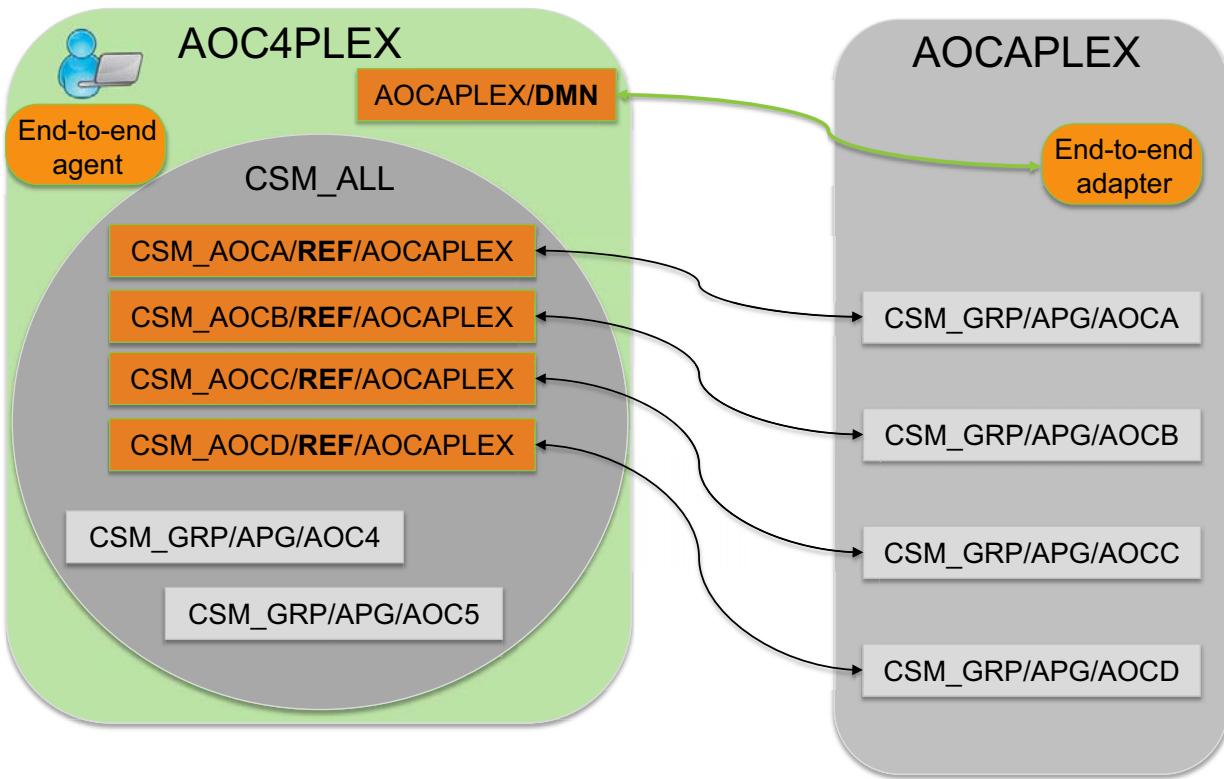
### Objectives

After completing this lesson, you should be able to describe end-to-end automation scenarios.

### References

## Cross sysplex automation example

### Monitor an end-to-end application



© Copyright IBM Corporation 2019

9-57

*Cross sysplex automation example: Monitor an end-to-end application*

In automation domain AOC4PLEX on the left we have reference resources pointing to automation domain AOCAPEX. The reference resources are a member of an application group that also contains members of type APG, application group, of the local SAplex.

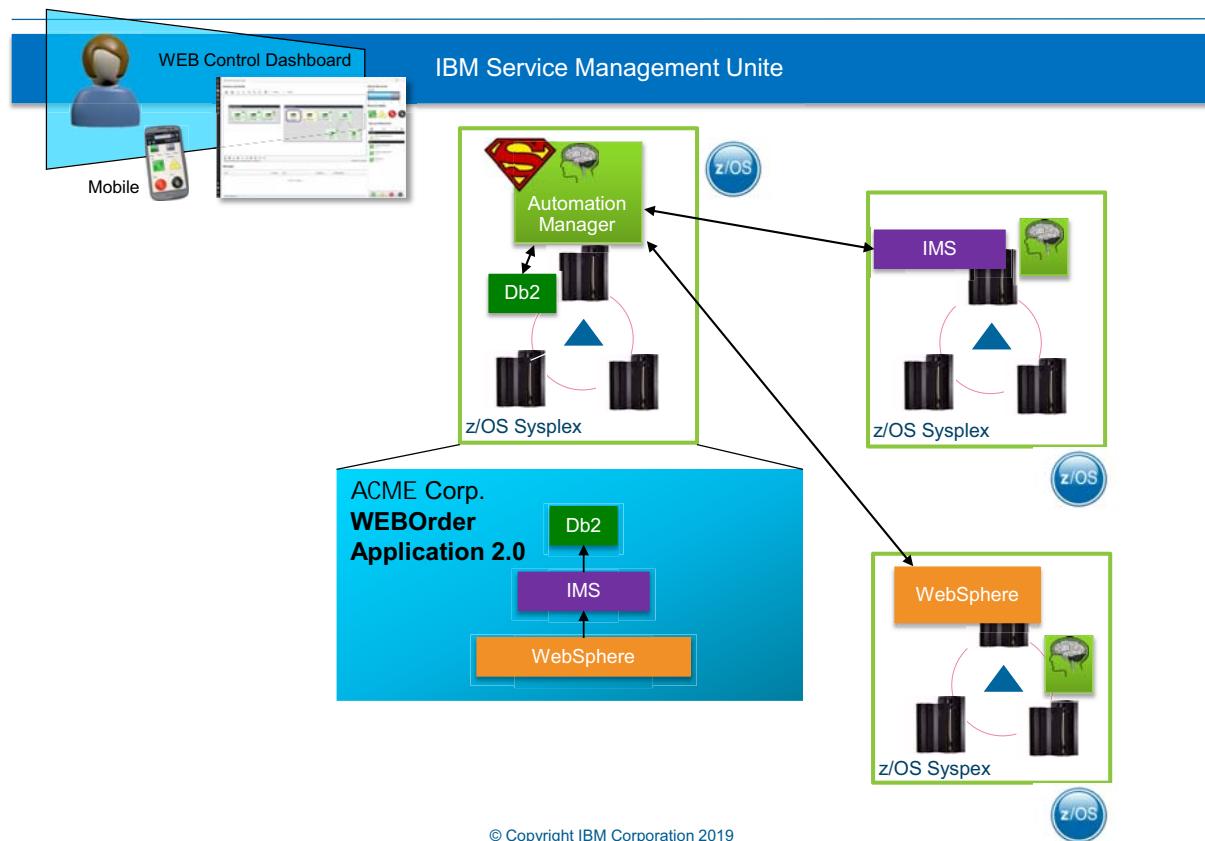
In automation domain AOCAPEX you have the real resources of type APG, application group running on different systems AOCA to AOCD.

Automation domain AOCAPEX is defined as DMN resource in automation domain AOC4PLEX which runs the E2E agent.

Automation domain AOCAPEX has to run the E2E adapter.

The communications storage manager (CSM) is a VTAM® component that allows authorized host applications to share data with VTAM and other CSM users without the need to physically copy the data. As it is important it is beneficial to monitor it as an end-to-end application although no dependencies exist.

## Extending automation scope across sysplexes



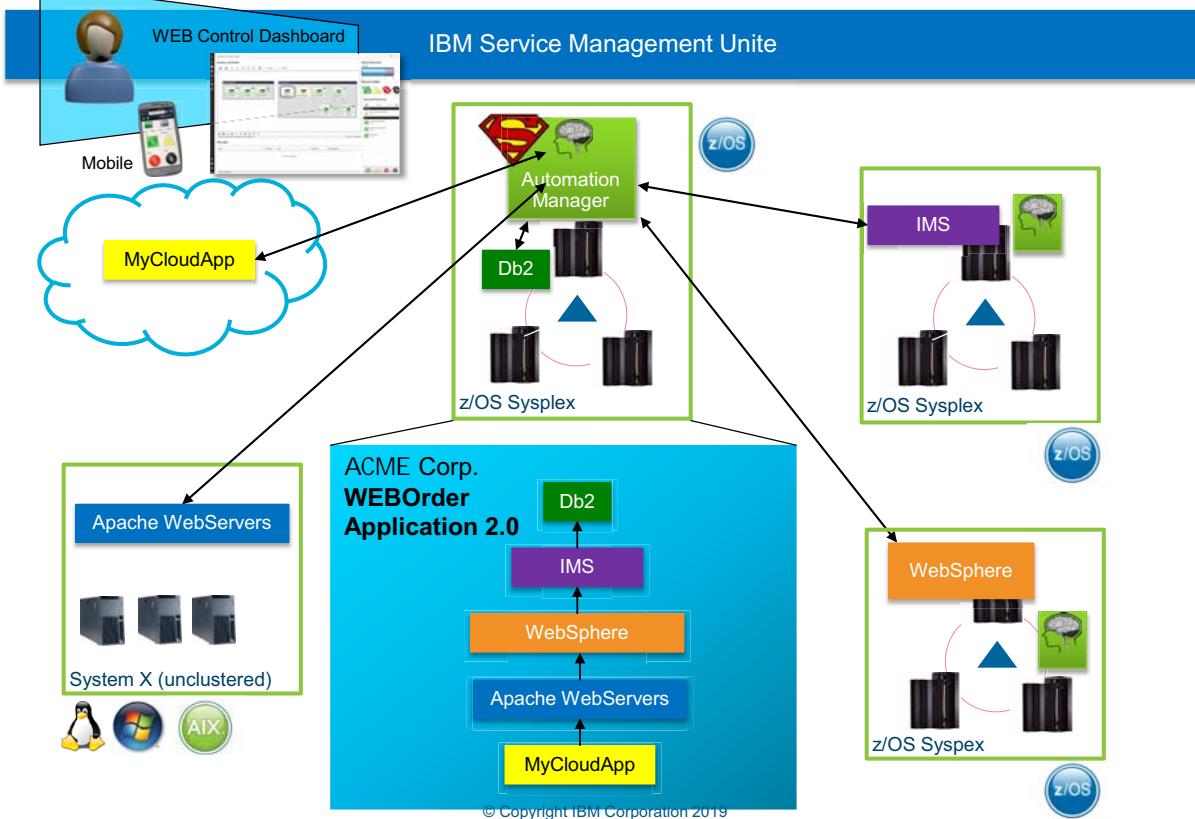
© Copyright IBM Corporation 2019

9-58

### Extending automation scope across sysplexes

This is a triple sysplex scenario. Db2, IMS, and WebSphere run in separate sysplexes and are aggregated into the “WEBOrder application 2.0” business application.

## Extending automation scope from sysplexes to distributed platforms & cloud applications

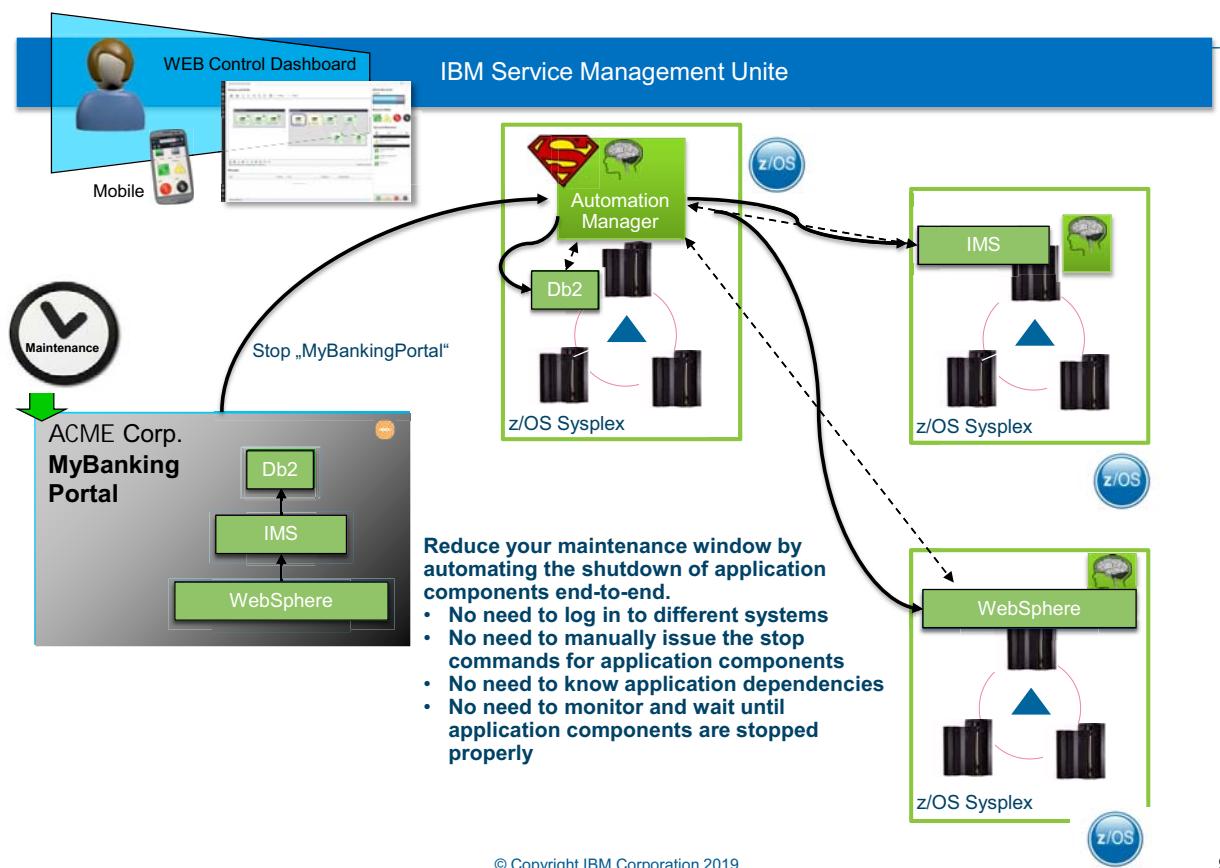


9-59

*Extending automation scope from sysplexes to distributed platforms & cloud applications*

This is the previous triple sysplex scenario. Db2, IMS, and WebSphere, which run in separate sysplexes, Apache WebServers on an unclustered System x and the MyCloudApp in the cloud are aggregated into the “WEBOrder application 2.0” business application.

## Sample scenario: Stop „MyPortal“ for maintenance



9-60

### Sample scenario: Stop „MyPortal“ for maintenance

This is a triple sysplex scenario. Db2, IMS, and WebSphere run in separate sysplexes and are aggregated into the “MyBankingPortal” business application.

Reduce your maintenance window by automating the shutdown of application components end-to-end.

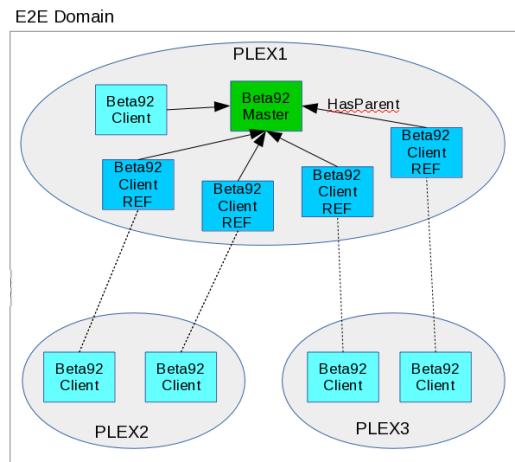
- No need to log in to different systems
- No need to manually issue the stop commands for application components
- No need to know application dependencies
- No need to monitor and wait until application components are stopped properly

# Cross sysplex scenario

## Log management with Beta 92



- Many customers are using **Beta 92** for job log management across their enterprise
- The vendor product has a single **master** and multiple **client** components in the enterprise



- Clients depend on master ( $\rightarrow$  HasParent)
- E2E automation allows to easily manage all components across the whole enterprise
- Because clients are started after and stopped before the master, **no false alarming takes place!**

### *Log management with Beta 92*

Many customers are using Beta 92 for job log management across their enterprise. The vendor product has a single master and multiple client components in the enterprise.

Beta 92 clients depend on the Beta 92 master which is taken care of by a HasParent relationship.

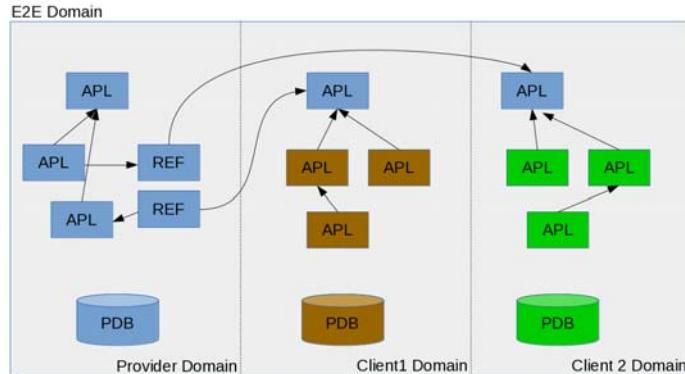
E2E automation allows to easily manage all components across the whole enterprise. No false alarming takes place because clients are started after and stopped before the master.

## Cross sysplex scenario Provider – Clients setup



### Provider – Clients setup

- The **Provider** offers individual SAplexes for its clients
- *client specific resources are dependent on infrastructure resources*



- Infrastructure resources are represented as normal (APL, APG, etc.) resources in the Client PDBs and as REF resources in the Provider PDB
- Having an E2E domain, the **Provider** can manage all infrastructure pieces himself and can create E2E views about the status of the E2E-domain
- **Clients** in turn can manage their resources independently as long as they follow the guidelines for using infrastructure resources

#### Cross sysplex scenario Provider – Clients setup

The Provider offers individual SAplexes for its clients. Infrastructure resources are required for client-specific resources.

Infrastructure resources are represented as normal (APL, APG, etc.) resources in the Client PDBs and as REF resources in the Provider PDB.

Having an E2E domain, the provider can manage all infrastructure pieces himself and can create E2E views about the status of the E2E domain.

Clients in turn can manage their resources independently as long as they follow the guidelines for using infrastructure resources.



## Summary

---

Now that you have completed this unit, you should be able to perform the following tasks:

- Explain the end-to-end automation architecture
- Describe and customize the end-to-end adapter
- Describe and customize the end-to-end agent
- Describe the Universal Automation Adapter
- Explain reference resources and remote domains
- Implement cross sysplex and end-to-end automation

Now that you have completed this unit, you should be able to perform the following tasks:

- Explain the end-to-end automation architecture
- Describe and customize the end-to-end adapter
- Describe and customize the end-to-end agent
- Describe the Universal Automation Adapter
- Explain reference resources and remote domains
- Implement cross sysplex and end-to-end automation



## **10 Performance-based automation with OMEGAMON**



## **Unit 10**

## **Performance-based automation with OMEGAMON**



© Copyright IBM Corporation 2019

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

**What this unit is about**

This unit contains information about using SA z/OS and the OMEGAMON products for performance-based automation: exception-based monitoring and event-based monitoring. You learn how to define the OMEGAMON Applications, sessions between SA z/OS and the OMEGAMON Classic monitors, and monitor resources to support performance-based automation. The unit also contains information about the SA z/OS monitoring agent and the Tivoli Enterprise Portal workspaces.

**How you check your progress**

You can check your progress in the lab exercises.

**References**

*SC34-2717 System Automation for z/OS Version 4  
Release 1 Defining Automation Policy*

*SC34-2647 System Automation for z/OS Version 4*

*Release 1 User's Guide*

[https://www.ibm.com/support/knowledgecenter/SSWRC\\_J\\_4.1.0/com.ibm.safos.doc\\_4.1/manuals.html](https://www.ibm.com/support/knowledgecenter/SSWRC_J_4.1.0/com.ibm.safos.doc_4.1/manuals.html)

# Objectives

After completing this unit, you should be able to perform the following tasks:

- Customize the PDB for OMEGAMON Applications
  - Import the \*ITM add-on policy
  - Define and manage the sessions between OMEGAMON and SA z/OS
- Issue OMEGAMON commands from NetView
- Describe exception-based monitoring
- Define OMEGAMON Monitor resources
- Describe event-based monitoring
- Describe how to set up and use the Tivoli Enterprise Portal workspaces

## Objectives

After completing this unit, you should be able to perform the following tasks:

- Customize the PDB for OMEGAMON Applications
- Import the \*ITM add-on policy
- Define and manage the sessions between OMEGAMON and SA z/OS
- Issue OMEGAMON commands from NetView
- Describe exception-based monitoring
- Define OMEGAMON Monitor resources
- Describe event-based monitoring
- Describe how to set up and use the Tivoli Enterprise Portal workspaces

# Lesson 1. Overview of OMEGAMON integration

## Lesson 1: Overview of OMEGAMON integration

- Use the \*ITM add-on policy to automate and manage OMEGAMON Applications
- Use OMEGAMON classic monitors for *exception-based* monitoring:  
Actively monitor for exception events
  - OMEGAMON monitor generates exception data when threshold is met
  - Filtering is based on installation-wide criteria

Note: Requires a VTAM session between SA z/OS and each OMEGAMON monitor
- Use OMEGAMON XE situations for *event-based* monitoring  
Passively monitor for OMEGAMON XE situation events
- Use MTR to influence the Health Status of resources
- Use Base libraries for more complex automation
- Use Tivoli Enterprise Portal workspaces
- Captured messages can be reported immediately as situations on the Tivoli Enterprise Portal (new in SA 3.5) using pure events

### What this lesson is about

This lesson provides an overview of OMEGAMON integration.

### Objectives

After completing this lesson, you should be able to describe how OMEGAMON is integrated into the System Automation environment.

### References

*SC34-2717 System Automation for z/OS Version 4 Release 1 Defining Automation Policy*  
*SC34-2647 System Automation for z/OS Version 4 Release 1 User's Guide*

SA z/OS provides policy definitions for managing the OMEGAMON (Classic and XE) Applications. The policy definitions are available in the \*ITM add-on policy. Additionally, SA z/OS can provide

automation based on OMEGAMON data. Exception-based monitoring is used to retrieve exceptions from the OMEGAMON Classic monitors. Event-based monitoring is used by situation events that are generated by OMEGAMON XE.

You can use the OMEGAMON data to set the SA z/OS Health Status and take corrective actions through automation. This unit contains information about several SA z/OS APIs, including one to issue OMEGAMON commands from NetView.

Captured messages can be reported immediately as situations on the Tivoli Enterprise Portal (new in SA 3.5) using pure events. A *Pure event* is displayed on TEP, if situation was defined and remains open until manually closed.

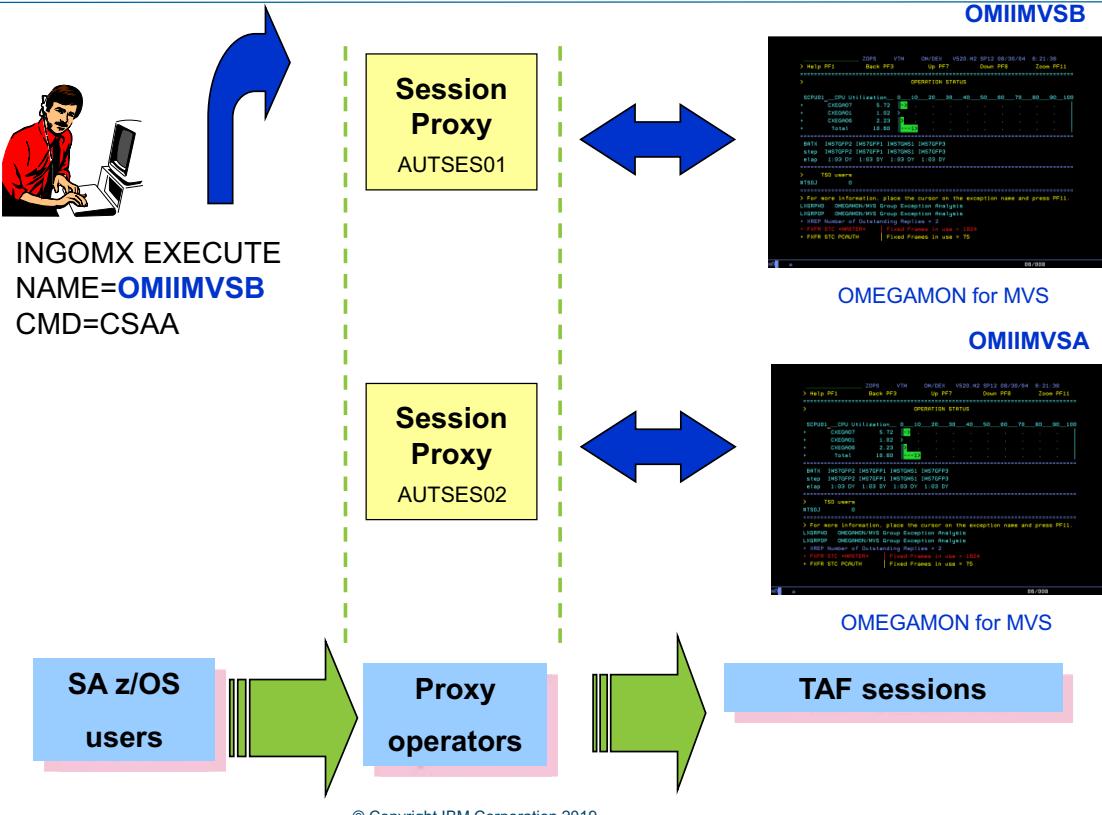
Event candidates are:

- Captured messages which are: UNUSUAL, IMPORTANT, CRITICAL
- Pre-defined alert points, like START\_FAILED or CS\_PROBLEM

Leverage existing INFORM LIST concept to tell SA for what APL / APG / MTR pure events for ITM should be generated

- INFORM LIST settings can be specified on class or instance level, defaults are supported as well, as today
- Only when keyword ITM is listed in the INFORM LIST, pure events will be generated

## Example command flow



Example command flow

When a user issues an INGOMX command, the session name is passed as a parameter, such as OMIIMVSB. SA z/OS uses the session name to determine the correct proxy operator to route the command to. The command is then routed from the proxy operator over a TAF session between the proxy operator and the OMEGAMON monitor.

For example, an **INGOMX run,NAME=OMIIMVSB,CMD=CSAA** command routes the OMEGAMON CSAA command over to the OMEGAMON for MVS monitor by using the OMIIMVSB session. The request flows over the TAF session between AUTSES01 and OMEGAMON for MVS on MVSB.

There are three basic components to a session between SA z/OS and an OMEGAMON monitor:

- **SA z/OS users:** Any NetView autotask or operator issuing the INGOMX or INGMTRAP commands

You can restrict access to the commands by using the NetView Command Authorization Table. Information about security is available in Lesson 5, “Security considerations,” on page 694.

- **Proxy operators:** NetView autotasks assigned to one or more OMEGAMON sessions

Proxy operators are defined by using the AOP policy definition. The session between SA z/OS and OMEGAMON is between the proxy operator and OMEGAMON. A proxy operator can connect to one or more OMEGAMON monitors. Each OMEGAMON monitor; for example, OMIIMVSB, can be connected to only one proxy operator. Commands are routed to a specific OMEGAMON monitor through a session; for example, OMIIMVSB. The session name is one of the command parameters.

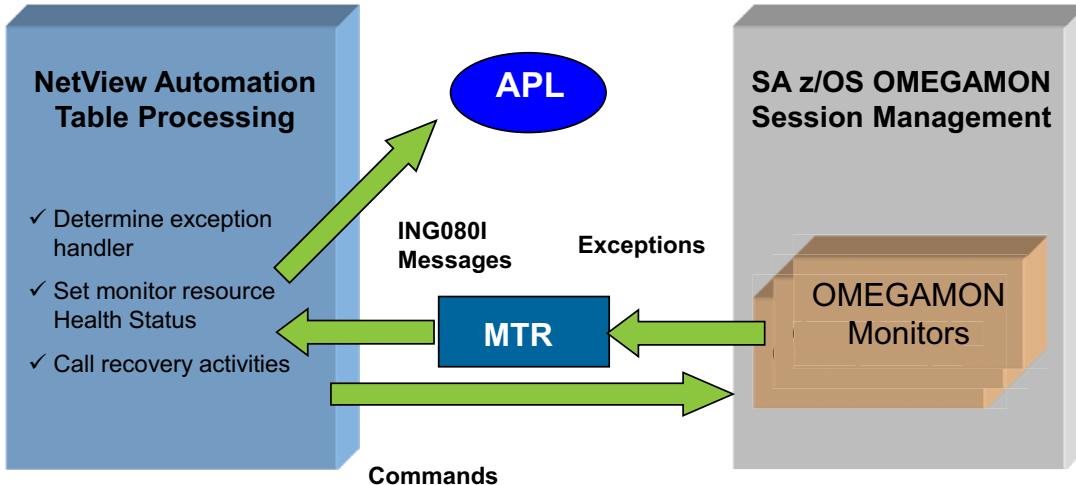
- **TAF sessions:** Actual connection to the OMEGAMON monitors from NetView

Each TAF session definition must have these items:

- Session name, such as OMIIMVSB (for OMEGAMON for MVS)
- TAF SRCLU name assigned to the session (for example, TFDA#000)
- Other parameters, such as the VTAM applid and OMEGAMON user ID

The proxy operators are defined in the AOP policy. Each proxy operator starts a Virtual Operator Station Task (VOST) and uses it for its TAF sessions to OMEGAMON. Because a VOST is used, the session must be a 3270 model 2 (screen size 24x80) connection. The OMEGAMON session information, such as the user ID and password, is defined in the NTW policy. The OMEGAMON monitors do not have to be running on the same system as the SA z/OS users. The SA z/OS users and proxy operators are required to be on the same system and running within the automation agent. The sessions are managed with the INGSESS command.

## Exception-based monitoring overview



- Use MTR to monitor for OMEGAMON exceptions
  - Notifies SA z/OS for each *interested* exception
- For Health Status processing and recovery, use the Automation Table that is created from the SA z/OS policy

© Copyright IBM Corporation 2019

10-5

### Exception-based monitoring overview

This slide illustrates an overview of exception-based automation. OMEGAMON Classic monitors collect data and set exceptions when a threshold is exceeded. Defining the data to collect and the thresholds for setting exceptions are OMEGAMON functions. SA z/OS uses monitor resources (MTRs) with a predefined monitor command (INGMTRAP) to request one or more exceptions from an OMEGAMON monitor. A TAF session is required between SA z/OS and each OMEGAMON monitor. This process is performed every time the SA z/OS monitor command is invoked, based on the monitor interval definition in the MTR policy.

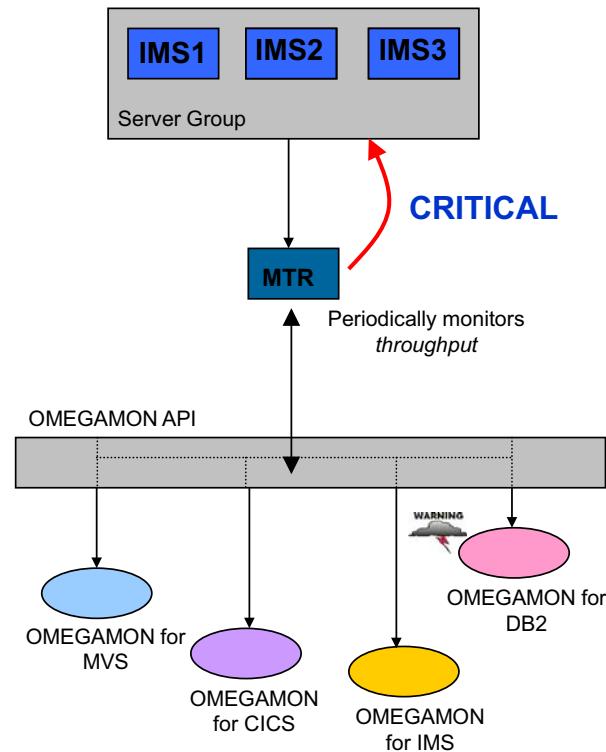
The OMEGAMON exceptions are interrogated by SA z/OS. The exceptions can affect the Health Status of the resource and cause additional automation. The mapping of an exception to a Health Status and the automation are defined in the MTR policy.

In this unit, you learn how to define the MTR policy and the sessions to connect SA z/OS to OMEGAMON. You also learn how to issue commands and manage the connections between SA z/OS and OMEGAMON.

## Example provisioning scenario

IMS Server Group with AVT=2  
Preference given to IMS1 and IMS2

Goal:  
Activate additional instance of IMS when throughput becomes CRITICAL  
Changing AVT to 3 starts IMS3



© Copyright IBM Corporation 2019

10-6

### Example provisioning scenario

In this simple scenario, a Server Group is defined with three IMS Applications (IMS1, IMS2, and IMS3). The Availability Target (AVT) for the group is 2. Therefore, IMS3 is not active.

This diagram shows a MTR to monitor for exceptions from OMEGAMON for IMS. SA z/OS establishes a session to the OMEGAMON for IMS monitor. When an exception occurs, the Health Status is set to CRITICAL. Automation starts when the Health Status is CRITICAL so that it can dynamically change the AVT to 3 to start another instance of the IMS Application. *Proactive automation* is provided to improve throughput for the IMS users.

To accomplish this scenario, these policy definitions are needed:

- The IMS Server Application Group.
- The session between SA z/OS and the OMEGAMON for IMS monitor.
- The MTR to retrieve IMS exceptions.
  - The MTR maps the exceptions to a CRITICAL Health Status.
  - The MTR uses automation to issue an INGGROUP command to change the AVT.

## Event-based monitoring overview

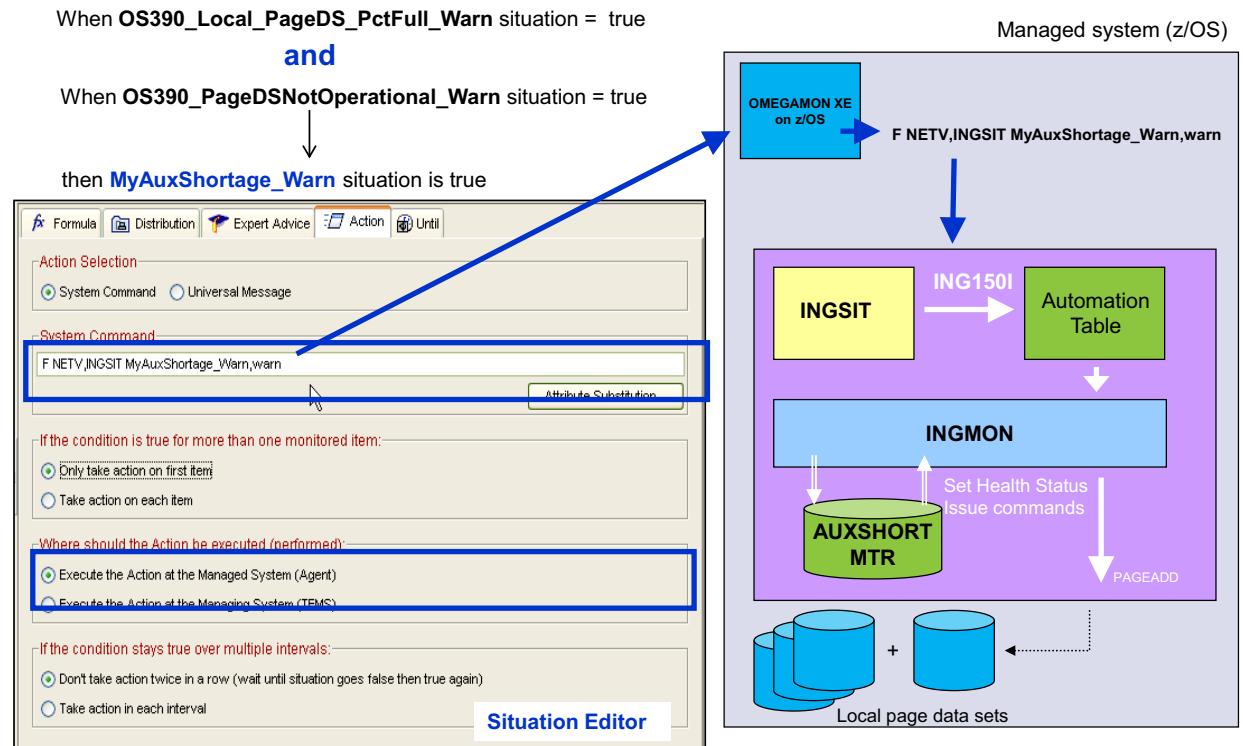
- Passive monitor resource (MTR)
  - The OMEGAMON XE monitors raise *situation events*
    - Also known as *reflex automation*; no monitor command needed
  - ING150I message automation
  - Action that raised the situation event runs on the managed system where the SA z/OS Automation Agent is running
- SOAP server connection to Tivoli Enterprise Monitoring Server
  - XML-based interface for commands and responses
  - Provides access to more monitoring data

### Event-based monitoring overview

You can use monitor resources to determine the overall health of a resource, such as an Application. You can set the Health Status of the Application to CRITICAL, for example, if there is a problem with the Application. The Health Status change causes the Compound Status to become DEGRADED, indicating that there is a problem with the Application. *Exception-based automation* uses monitor resources to retrieve exceptions from the OMEGAMON Classic monitors and set the Health Status of a resource based on the exception data.

*Event-based automation* introduces the concept of passive monitoring by using monitor resources that are used when situation events are generated from the OMEGAMON XE monitors. This concept is called *event-based monitoring*. Within the IBM Tivoli Monitoring architecture, this concept is also called *reflex automation*.

## Event-based automation example



© Copyright IBM Corporation 2019

10-8

### Event-based automation example

On this slide, you see an example of event-based automation. In this case, the situation name is MyAuxShortage\_Warn. The Formula tab, if selected, shows the OS390\_Local\_PageDS\_Pct\_Full\_Warn situation and the OS390\_PageDSNotOperational\_Warn situation evaluating as true. The Action tab shows the INGSIT command, which is used when the MyAuxShortage\_Warn situation generates a situation event. The command is issued the first time that a situation event is generated at the managed system (z/OS). The INGSIT command generates an ING150I message. The ING150I message starts the Automation Table and runs the INGMON command to set the Health Status of the AUXSHORT monitor resource (MTR). Commands that are defined for the MTR result in the issuing of a PAGEADD command.



**Hint:** Situations are defined with the Situation Editor component of the Tivoli Enterprise Portal. Some products provide situations, such as the OS390\_Local\_PageDS\_Pct\_Full\_Warn situation.



**Note:** Situations generate situation events based on the values of their attributes.

Attributes are characteristics of objects that are monitored by a monitoring agent. Attributes are queried to collect data for workspaces and situations. Related attributes can be organized into attribute groups. Situations test one or more conditions (values of attributes) and can trigger an event when conditions are met. Situation events indicate a problem or potential problem. Situation events can have one of the following severities associated with them:

- Critical
- Warning
- Informational

# Lesson 2. Implement \*ITM add-on policy

## Lesson 2: Implement \*ITM add-on policy

- On the Data Management panel, select **Import from add-on**
  - Import the \*ITM add-on policy
  - Customize policy; for example, modify job name for Applications that you want to automate
- Define sessions for exception-based monitoring, such as Network (NTW) policy
- Define automated operators
  - SESS\_AUTOOPS: OMEGAMON session operators
  - KAH\_AUTOOPS. SA z/OS monitoring agent operators

### What this lesson is about

This lesson describes how to implement the \*ITM add-on policy.

### Objectives

After completing this lesson, you should be able to implement the \*ITM add-on policy.

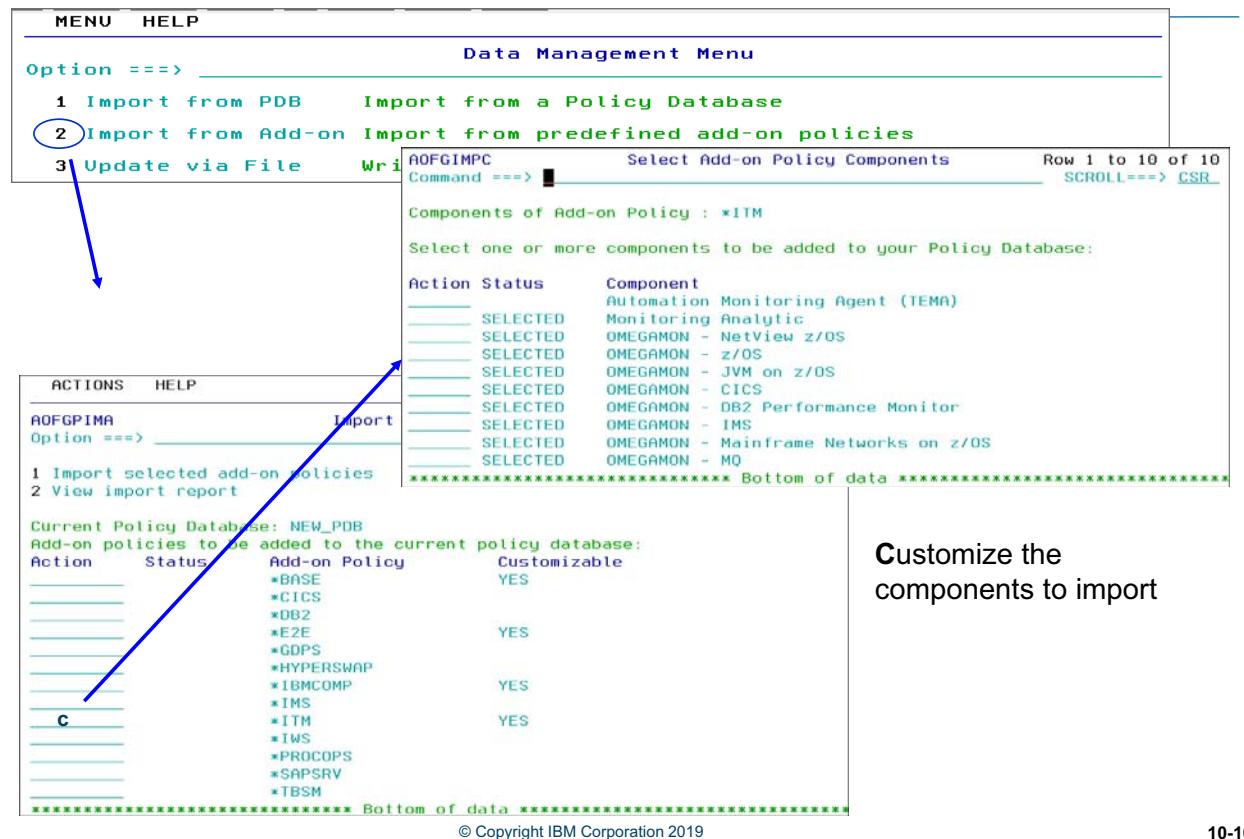
### References

*SC34-2717 System Automation for z/OS Version 4  
Release 1 Defining Automation Policy*

This lesson is about the policies that are required to implement exception-based and event-based automation: \*ITM add-on, NTW, and AOP. The \*ITM add-on policy contains policies for the OMEGAMON Applications. You can customize it to select a subset of the policies for the OMEGAMON II or OMEGAMON XE environments. After importing the add-on policy, you must

modify the definitions for your environment. For example, most likely your job names do not match the samples provided by the SA z/OS add-on.

# Import \*ITM add-on policy



Customize the components to import

## Import \*ITM add-on policy

To import the \*ITM add-on policy, access the Data Management Menu (panel AOFGIMPC). Select option **2**, Import from Add-on, to import \*ITM policy definitions. You see a list of the available add-on policies on panel AOFGPIMA. Scroll down by pressing PF8 to see more policies. To select a subset of the \*ITM policies, enter **C** in the Action column for the \*ITM add-on to customize the add-on.

The Select Add-on Policy Components panel (AOFGIMPC) is displayed with the components that you can import. In this case, all but the SA z/OS monitoring agent are selected. Press PF3 to end component selection and return to panel AOFGPIMA. Select the Import selected add-on policies option (**1**) and press Enter to display the policy definitions for the selected components.

## Import \*ITM add-on policy (continued)

| Entries of selected Add-on Policies |                  |              |   | Row 1 to 22 of 47<br>SCROLL==> CSR |
|-------------------------------------|------------------|--------------|---|------------------------------------|
| Option ==>                          |                  |              |   |                                    |
| 1 Modify data                       | 2 Check data     | 3 Run import | 4 View report                             |                                    |
| Action                              | Entry Name       | Type         | C D Short Description                     |                                    |
| m                                   | SYSPLEX1         | GRP          | Y Placeholder. Original in *BASE          |                                    |
| m                                   | SYS1             | SYS          | Y Placeholder. Original defined in *BASE  |                                    |
| m                                   | SYS2             | SYS          | Placeholder. Original defined in *BASE    |                                    |
| m                                   | SYS3             | SYS          | Y Placeholder. Original defined in *BASE  |                                    |
|                                     | BASE_ITM         | APG          | ITM Base Components                       |                                    |
|                                     | ING_ANALYTIC     | APG          | Automation Monitoring Analytics           |                                    |
|                                     | OM_CICS          | APG          | OMEGAMON - CICS                           |                                    |
|                                     | OM_DB2           | APG          | OMEGAMON - DB2 Performance Monitor        |                                    |
|                                     | OM_TMS           | APC          | OMEGAMON - TMS                            |                                    |
| Action                              | Entry Name       | Type         | C D Short Description                     |                                    |
| 0                                   | C_ITM            | APL          | * ITM Application Class                   |                                    |
| 0                                   | C_LOOPSUPP       | APL          | * Looping Address Space Suppression Class |                                    |
| 0                                   | IBMCN            | APL          | OMEGAMON Subsystem Interface              |                                    |
| 0                                   | IBMC5            | APL          | OMEGAMON CICS Agent                       |                                    |
| 0                                   | IBMD5            | APL          | ITM Monitoring Server (TEMS)              |                                    |
| 0                                   | IBMD5            | APL          | OMEGAMON DB2 Performance Monitor Agent    |                                    |
| 0                                   | IBMETE           | APL          | OMEGAMON End-To-End Response Monitor      |                                    |
| 0                                   | IBMGW            | APL          | OMEGAMON CICS TG on z/OS Agent            |                                    |
| 0                                   | IBMI2            | APL          | OMEGAMON IMS CUA Interface                |                                    |
| 0                                   | IBMIS            | APL          | OMEGAMON IMS Agent                        |                                    |
| 0                                   | IBMJJ            | APL          | OMEGAMON JVM Monitoring Agent             |                                    |
| 0                                   | IBMJT            | APL          | OMEGAMON JVM Monitoring Collector         |                                    |
| 0                                   | IBMMC            | APL          | OMEGAMON MQ Configuration Agent           |                                    |
| 0                                   | IBMMQ            | APL          | OMEGAMON MQ Agent                         |                                    |
| 0                                   | IBMM2            | APL          | OMEGAMON z/OS CUA Interface               |                                    |
| 0                                   | IBMM2CS          | APL          | OMEGAMON z/OS CSA Analyzer (Classic)      |                                    |
| 0                                   | IBMM2EZ          | APL          | OMEGAMON z/OS EPILOG Zoom (Classic)       |                                    |
| 0                                   | IBMM2HD          | APL          | OMEGAMON z/OS Hist. Data I/F (Classic)    |                                    |
| 0                                   | IBMM2HI          | APL          | OMEGAMON z/OS EPILOG Collector (Classic)  |                                    |
| 0                                   | IBMM2RC          | APL          | OMEGAMON z/OS Realtime Coll. (Classic)    |                                    |
| 0                                   | IBMNA            | APL          | OMEGAMON NetView z/OS Agent               |                                    |
| 0                                   | IBMN3            | APL          | OMEGAMON Mainframe Networks z/OS Agent    |                                    |
| 0                                   | IBMOCO           | APL          | OMEGAMON CICS Interface (Classic)         |                                    |
| 0                                   | IBMOIO           | APL          | OMEGAMON IMS Interface (Classic)          |                                    |
| 0                                   | IBMON            | APL          | OMEGAMON Mainframe Networks CUA I/F       |                                    |
| 0                                   | IBMO2            | APL          | OMEGAMON DB2 Perf. Monitor (Classic)      |                                    |
| 0                                   | IBMQI            | APL          | OMEGAMON Integration Bus Agent            |                                    |
| 0                                   | IBMTOM           | APL          | OMEGAMON Enhanced 3270 User Interface     |                                    |
| 0                                   | LOOPSUPP         | APL          | Looping Address Space Suppression         |                                    |
| 0                                   | ING_TEMA_AUTOOPS | AOP          | Automation Monitoring Agent Auto Ops      |                                    |
| 0                                   | SESS_AUTOOPS     | AOP          | OMEGAMON Session Operators                |                                    |
| 0                                   | SOAP_SERVERS     | NTW          | Network definitions for SOAP servers      |                                    |

Animated

© Copyright IBM Corporation 2019

10-11

### Import \*ITM add-on policy (continued)

The entries shown here are for these items:

- Applications, like OMEGAMONS
- Application classes
- Application Groups
- Automation Operators like SESS\_AUTOOPS
- Network definitions for SOAP

You rename the system entries or other entries by typing the new name in the Entry Name field. For example, press the Tab key to SYS2 and overtype MVSB.

SYS1 and SYS2 could be renamed to systems MVSA and MVSB. Renaming the system entries reduces the have to link the resources to the systems later.

Entries can be removed from the list. For example, the SAplex SYSPLEX1 already exists in the target PDB and can be removed. After customizing the list of policy definitions, press Enter to start the policy import. The Command Progress Display panel displays messages related to importing the policy definitions.



**Important:** Do not press PF3 to go back to the previous panel at this point. You lose all of your customization.

After the import is complete, you edit the policy for each Application, Application Group, and class to modify information such as the job name for each to suit your environment.

## Define session with OMEGAMON classic

| AOFGEPOL  |   | Policy Selection   | Row 1 to 10 of 10<br>Scroll ==> CSR |
|---|---|--|-------------------------------------|
| Command ==>                                       |   |  |                                     |
| Entry Type : Network<br>Entry Name : BASE_NETWORK |   | PolicyDB Name : NEW_PDB<br>Enterprise Name : SAPLEX  | NTW<br>(network)<br>policy          |
| Action  | Policy Name<br>DESCRIPTION<br>SDF FOCALPOINT<br>FULL SESSIONS<br>GATEWAY<br>s<br>OMEGAMON SESSIONS<br>AUTHENTICATION<br>SOAP SERVER | Policy Description<br>Enter description<br>Define Status Display Facility focal point<br>Define TAF sessions (Applications)<br>Define gateways<br>Define OMEGAMON sessions<br>Define authentication information (INGPW)<br>Define ITM SOAP-Servers |                                     |
| WHERE USED<br>COPY                                |   | List systems linked to this entry<br>Copy data from an existing entry  |                                     |

| AOFGOSOA  |  | OMEgamon Session Attributes                         |
|---|--|---|
| Command ==>                                       |  |   |
| Entry Type : Network<br>Entry Name : BASE_NETWORK |  | PolicyDB Name : NEW_PDB<br>Enterprise Name : SAPLEX |
| Session Name : OMIIMVSA                           |  | <b>OMEgamon Classic VTAM APPLID</b>                 |
| VTAM applid . . . A01M2RC                         |  | Name of OMEGAMON VTAM application                   |
| VTAM logmode . . .                                |  | Logmode for session setup                           |
| TAF source LU . . .                               |  | Name of TAF source LU or blank                      |
| OMEGAMON type . . . OMIIMVS                       |  | (OMIICICS OMIIDB2 OMIIMS OMIIMVS)                   |
| User ID . . . .                                   |  | User ID to log on to OMEGAMON                       |
| Password . . . .                                  |  | Password of the logon user or SAFPW                 |
| Timeout . . . .                                   |  | Time to wait for OMEGAMON response (1-999 sec)      |
| Session data . . .                                |  |   |

© Copyright IBM Corporation 2019

10-12

### Define session with OMEGAMON classic

The OMEGAMON sessions and security are defined by using the Network (NTW) as follows:

- Use the OMEGAMON SESSIONS policy item to define the session parameters, such as user ID or session name.
- Use the AUTHENTICATION policy item to define password security by using the **GETPW** command.

On this slide, you see the OMEGAMON Session Attributes panel for session OMIIMVSA. This panel defines how to connect to the OMEGAMON monitors. In this case, the connection is to the OMEGAMON II for MVS monitor (**Type** field is OMIIMVS). The session between SA z/OS and OMEGAMON must be a 3270 model 2 (24x80) connection.

The following panel fields are available:

- Session Name: The session name uniquely identifies an OMEGAMON session. You can define multiple sessions referring to the same OMEGAMON monitor.
- VTAM Applid: The VTAM Application ID of the OMEGAMON monitor as defined in VTAMLST. The applied VTAM might contain symbolic names that are resolved when the automation control file is loaded or refreshed.

The VTAM Applid is used in the APPLID= parameter when the BGNSESS command is issued to start the TAF session.

- Type: Identifies the type of session. Four choices correspond to each of the supported OMEGAMON monitors. OMIIIMVS identifies this type as a session to an OMEGAMON II for MVS monitor.
- User ID: The user ID that is used to establish the session with the OMEGAMON monitor. It can be a generic user on behalf of which all autotasks or operators can log on to the OMEGAMON monitor. The user ID might contain symbolic names that are resolved when the automation control file is loaded or refreshed.

If you do not have operator security set up for OMEGAMON, you can leave the User ID field blank.

- Password: Contains the session password (not encrypted) or SAFPW. You can use SAFPW to have SA z/OS dynamically retrieve a session password from the NetView password data set. If you do not use SAFPW, then anyone who has access to the customization dialog or the **INGSESS** command can see the password.
- Timeout: Specifies the maximum time that a request to OMEGAMON can take. When the timeout expires, the current request is canceled. The default timeout is set from WAITTIME as specified in CNMSTYLE.
- Session Data: Additional session attributes are passed to the session upon opening the communication, like the following example:
  - To monitor a particular DB2 subsystem, you must define *DB2=ssname* to establish a connection with the correct DB2.
  - To monitor a particular CICS region, you must define *C/CS=jobname* to establish a connection with the correct CICS.

Refer to the OMEGAMON publications for additional details.

The connection between SA z/OS and OMEGAMON is a TAF full screen session. TAF SRCLUs are defined in VTAMLST. Use model APPL statements such as this one:

```
TFaa#*      APPL MODETAB=AMODETAB,EAS=9,          X
                  DLOGMOD=M2SDLNCQ
```

The SRCLU is an LU name based on the naming convention of **TFaa#nnn** where *aa* are the last two characters of the NetView domain ID and *nnn* is a decimal number in the range 000-999. When SA z/OS starts the TAF session, the SRCLU name is determined dynamically. SA z/OS issues an **ATTACH BGNSESS FLSCN APPLID=A01M2RC SRCLU=\*** command to create the VOST and start the TAF full-screen session to the OMEGAMON for MVS monitor.

## Define automation operators

| AOFGENAM     |                      | Entry Name Selection                 |         | Row 1 to 5 of 5   |
|--------------|----------------------|--------------------------------------|---------|---|
| Command ==>  |                      |                                      |         | Scroll ==> CSR  |
| Entry Type : | Automation Operators | PolicyDB Name :                      | NEW_PDB | On the AOP policy definition panel, select SESS_AUTOOPS and press Enter |
|              |                      | Enterprise Name :                    | SAPLEX  |   |
| Action       | Entry Name           | Short Description                    |         |   |
|              | BASE_AUTOOPS         | Default Automation Operator          |         |   |
|              | GATEWAY_AUTOOPS      | Gateway Automation Operator          |         |   |
|              | ING_TEMA_AUTOOPS     | Automation Monitoring Agent Auto Ops |         |   |
|              | SESS_AUTOOPS         | OMEGAMON Session Operators           |         |   |
|              | WORK_AUTOOPS         | Automation Work Operators            |         |   |

| AOFGDYNO       |   | Automation Operator Definitions    |                 | Line 00000001 Col 001 075  |
|----------------|---|------------------------------------|-----------------|----------------------------|
| Command ==>    |   |                                    |                 | Scroll ==> CSR             |
| Entry Type :   | Automation Operators                                    | PolicyDB Name :                    | NEW_PDB         |                            |
| Entry Name :   | SESS_AUTOOPS  | Enterprise Name :                  | SAPLEX          |                            |
| Line Commands: | S (Select), M (Move), B (Before), A (After), R (Repeat) | I (Insert lines), D (Delete lines) |                 |                            |
| Cmd            | AutoFunc  | Primary Operator                   | Backup Operator | Messages for this Operator |
|                | AOFSES01  | AUTSES01                           |                 |                            |
|                | AOFSES02  | AUTSES02                           |                 |                            |
|                | AOFSES03  | AUTSES03                           |                 |                            |

SA z/OS provides definitions for AUTSES01 - AUTSES03

### Define automation operators

Use the AOP policy to define your SESS\_AUTOOPS for OMEGAMON SESSIONS and ING\_TEMA\_AUTOOPS for the SA z/OS monitoring agent. The SESS\_AUTOOPS policy defines the autotasks (*proxy operators*) for connecting to the OMEGAMON monitors. The sessions are distributed among the autotasks by using a *round-robin* methodology. SA z/OS includes three defined autotasks. You can add more if needed.

You do not have to define the SA z/OS users who issue the **INGOMX** and **INGMTRAP** commands. This definition is for defining the proxy operators only. Any SA z/OS user who has the appropriate authority can issue OMEGAMON commands.



**Important:** Do not specify a backup automation operator for the OMEGAMON sessions. If one is specified, it is ignored.

The ING\_TEMA\_AUTOOPS definitions are like SESS\_AUTOOPS. AUTKAH01 - AUTKAH03 are provided by SA z/OS. The ING\_TEMA\_AUTOOPS are used with the SA z/OS monitoring agent and Tivoli Enterprise Portal workspace data.

# Lesson 3. Commands and programming interfaces

## Lesson 3: Commands and programming interfaces

- INGSESS
  - Display one or all OMEGAMON sessions
  - Start or stop an OMEGAMON session
  - Display detailed data for an OMEGAMON session
- INGOMX
  - Issue OMEGAMON commands, such as CSAA
  - Retrieve OMEGAMON exceptions
  - Send XML request to SOAP server
- INGMTRAP: Predefined monitor command for exception-based monitoring
- INGMON: Generic routine to set the status for a monitor resource and to issue commands in response to a message or an OMEGAMON exception
- INGSIT: Generic routine to generate an ING150I message from an OMEGAMON situation event. The situation event must be associated with one monitor resource

### What this lesson is about

In this lesson we review the commands and programming interfaces associated with OMEGAMON.

### Objectives

After completing this lesson, you should be able to describe the commands and programming interfaces associated with OMEGAMON.

## References

- SC34-2647 System Automation for z/OS Version 4 Release 1 User's Guide*
- SC34-2649 System Automation for z/OS Version 4 Release 1 Operator Commands*
- SC34-2650 System Automation for z/OS Version 4 Release 1 Programmer's Reference*

You can use the **INGSESS** command to manage the sessions with the OMEGAMON monitors; for example, start and stop, and display session attributes and statistics. **INGSESS** with no parameters displays all defined sessions. **INGSESS OMII\*** displays all defined sessions that begin with OMII.

**INGOMX** is available for all operators to issue OMEGAMON commands and retrieve exceptions. **INGMTRAP** is intended as a monitor command to monitor for OMEGAMON exceptions. Specify INGMTRAP as the monitor command for active monitors. INGOMX and INGMTRAP require a connection with one or more OMEGAMON Classic monitors. INGMTRAP, INGMON, and INGSIT are used by your system administrator as part of a monitor resource policy or situation definition.

## INGSESS example

```
INGKYSSO          SA z/OS - Command Dialogs      Line  1   of 2
Domain ID = AOFDA  INGSESS                         Date = 12/11/12
Operator ID = SASTUD1    System = MVSA             Time = 23:34:47
CMD: B Start session  C Stop session  D Details
CMD Session      System     Type      Status      Appl-id  User id  SessOper
---  ---  ---  ---  ---  ---  ---
---  OMIIIMVSA    MVSA      OMIIIMVS  ACTIVE     A01M2RC  AOFSES01
---  OMIIIMVSB    MVSA      OMIIIMVS  INACTIVE   A02M2RC  AOFSES02

In this example of INGSESS REQ=DISPLAY, two sessions are defined: OMIIIMVSA and OMIIIMVSB
- OMIIIMVSB is on a different system and is not started
- The OMIIIMVSA session is started

To see session details, type a D in the CMD column and press Enter

Command ==> PF1=Help  PF2=End  PF3=Return  PF9=Refresh  PF6=Roll  PF12=Retrieve
```

© Copyright IBM Corporation 2019

10-15

### INGSESS example

On the slide is an example of **INGSESS**; in this case to display all sessions that are defined for the MVSA system. There are two sessions defined: OMIIIMVSA and OMIIIMVSB. OMIIIMVSA is on MVSA and active. OMIIIMVSB is on MVSB and inactive.

A brief explanation of the columns is in the following list:

- **Session:** Displays the name of the session as defined in the policy such as OMIIIMVSB.
- **System:** Displays the name of the system that is connected to the OMEGAMON monitor.
- **Type:** Displays the monitor type of OMEGAMON monitor, such as OMIIIMVS.
- **Status:** Displays the current status of the session.
- **Appl-id:** Displays the target VTAM APPL ID for the session.
- **User id:** Displays the user that is defined to log on to the OMEGAMON Application. The user ID is needed to control access to the OMEGAMON Application.

This field can be blank if product-level security is not implemented for this OMEGAMON Application.

- **SessOper:** Displays the name of the autotask being used as the proxy operator for the session.

You can start, stop, or display details for a specific session by selecting one of the options. You can start sessions only when their status is MAINT or INACTIVE. Stopping a session sets its status to MAINT.

## INGSESS details

|                            |                 |                     |                 |
|----------------------------|-----------------|---------------------|-----------------|
| INGKYSS1                   | SA z/OS         | - Command Dialogs   | Line 1 of 24    |
| Domain ID = A0FDA          | -----           | INGSESS             | Date = 12/14/12 |
| Operator ID = SASTUD1      | -----           | System = MVSA       | Time = 12:56:27 |
| Session                    | : OMIIMVSA      | in Sysplex : PLEX12 |                 |
| System                     | : MVSA          |                     |                 |
| Type                       | : OMIIMVS       |                     |                 |
| Description                | : MVSA OMEGAMON |                     |                 |
| Status                     | : ACTIVE        |                     |                 |
| Session Operator           | : A0FSES01      |                     |                 |
| Actual LU                  | : TFDA#000      |                     |                 |
| Application id             | : A01M2RC       |                     |                 |
| User id                    | :               |                     |                 |
| Password                   | : *****         |                     |                 |
| Timeout                    | : 29            |                     |                 |
| Logon data                 | :               |                     |                 |
| Version.Profile            | : V420./C       |                     |                 |
| System                     | : MVSA          |                     |                 |
| LogMode                    | :               |                     |                 |
| Users                      | : SASTUD1       |                     |                 |
| Statistics...              |                 |                     |                 |
| Total # Commands           | :               | 4                   |                 |
| Total # exception analysis | :               | 2                   |                 |
| Total # exceptions tripped | :               | 2                   |                 |

This panel shows the session information, including the proxy operator. In this case, the User ID field is null because OMEGAMON security is disabled.

Defined LU : \*

**Total # commands:** Number of INGOMX commands.

**Total # exception analysis:** Number of INGOMX commands that requested exception data.

**Total # exceptions tripped:** Number of ING080I messages generated from exception analysis.

### INGSESS details

On the slide is an example of **INGSESS REQ=DETAILS** for the OMIIMVSA session. You can look at the data displayed on this panel in four sections.

- The first section of the panel has general session information.
- The second section of the panel has status information, including the proxy operator and SRCLU being used.
- The third section of the panel has the OMEGAMON session policy definitions, including the VTAM APPL ID, user ID, password, and users of the session. The Users field can contain the names of operators and autotasks that have used this session. The Password field contains SAFPW, or asterisks if a password is coded in the policy definition.

This information is defined in the OMEGAMON SESSIONS policy item of the NTW policy. The User id field is blank. In this example, there is no user ID defined for the session. You can connect to OMEGAMON for MVS without a user ID with minimal functions.

- The fourth section, statistics, contains these fields:
  - Total # Commands: The number of commands sent to the OMEGAMON monitor from all tasks by using **INGOMX run CMD=**.

- Total # exception analysis: The number of times exception analysis was requested by using either **INGOMX TRAP XTYPE=** or **INGMTRAP** commands.
- Total # exceptions tripped: The number of times exception analysis returned an exception message back to the requester, such as **INGMTRAP**.

The slide example shows four operator commands for the session. Exception analysis was invoked two times, and generated two exception messages.

Another example of displayed statistic lines only is like the following one:

|                              |   |    |
|------------------------------|---|----|
| Total # Commands             | : | 2  |
| Total # exception analysis : |   | 12 |
| Total # exceptions tripped : |   | 17 |

In this example, two operator commands were sent to the session. Exception analysis was invoked 12 times, and generated 17 exception messages. You can monitor for conditions such as this one by issuing a command like the following one:

```
PIPE NETV INGSESS OMIIIMVSA REQ=DETAIL OUTMODE=LINE | SEP | TAKE LAST 5 | TAKE  
FIRST 3 | CONS
```

Sometimes, the exceptions tripped might be greater than the exception analysis. These instances indicate a condition in which an exception analysis request resulted in more than one exception. Each of these counters is reset to zero (0) when the session is started.

## Recovering a failed session

To recover a failed session with status AUTHFAIL or SESSFAIL, follow these steps:

1. Put session in maintenance mode:

**INGESS omiimvsb req=stop outmode=line**

2. Fix the problem

3. Restart the session:

**INGESS omiimvsb req=start outmode=line**

Note:

You can use the INGESS panel instead of line-mode commands

A failed session can have a status of AUTHFAIL when the connection request failed because of invalid password, or SESSFAIL for all other possibilities. Before you can restart a failed session, the session must be put into maintenance mode by stopping it. This action changes the state to MAINT.



**Note:** You can start a session only with a status of INACTIVE (never started) or MAINT.

## INGOMX overview

With INGOMX, you can perform these tasks:

- Issue OMEGAMON **major**, **minor**, and **immediate** commands
  - Provides an interface between operators and any of the classic OMEGAMON monitors for CICS, DB2, IMS, and MVS
  - Includes automation operators
- Retrieve one or more exceptions from OMEGAMON
  - Exceptions are returned to operator in command response
- Send XML requests to a SOAP server

### INGOMX overview

Operators can use **INGOMX** to perform the following tasks:

- Issue OMEGAMON commands. Specify the run CMD= parameters.
  - Major commands select general categories for display, such as system information, resource utilization, or storage utilization.
  - Minor commands display detailed information about the major category and must be preceded by the major command.
  - Immediate commands serve various functions. Some are system monitoring commands, while others provide information about your session or about OMEGAMON.
- Retrieve OMEGAMON exception. Specify the TRAP XTYPE= parameters.
- Send XML requests to a SOAP server. Specify the SOAPREQ SERVER= parameters. The server must be defined with the SOAP SERVER policy item of the NTW policy.

Users of the INGOMX command do not have to perform session management. Instead, if the session is not active when a user issues an INGOMX request, SA z/OS automatically attempts to start the session on behalf of the user.

Access to the OMEGAMON commands and their parameters is controlled with the NetView Command Authorization Table (CAT) or SAF products. More information about command security is in [“Security considerations”](#) on page 694.

The output from an INGOMX command is displayed to the terminal but is not logged or exposed to automation. You can use the NetView **PIPE** command to issue your INGOMX commands and format the data so that it is more easily read and logged.

If the request is to retrieve one or more OMEGAMON exceptions, specify the TRAP parameter with the XTYPE parameter to define the exceptions that you want to retrieve. See the *IBM Tivoli OMEGAMON for MVS Reference Manual* for more details about OMEGAMON exceptions.

Use CMD=\* only when you are calling INGOMX from a program such as a REXX routine. In that case, you define each command as an array item and pass the array as input to the INGOMX command.

## OMEGAMON command examples

- Determine common storage use by using the OMEGAMON for MVS CSAA command: `INGOMX EXECUTE,NAME=OMIIMVSB,CMD=CSAA`

| CSAA SUMMARY |         |         |          |        |        |              |
|--------------|---------|---------|----------|--------|--------|--------------|
|              |         | System  |          |        |        |              |
|              |         | Maximum | Pre-CSAA | Orphan | Usage  |              |
|              |         | -----   | -----    | -----  | -----  |              |
| 0            | 2       | 4       | 6        | 8      | 100    |              |
| + CSA        | 4796K   |         | 0        | 0      | 246K   | 5.1% >       |
| + ECSA       | 123812K |         | 0        | 79K    | 23532K | 19.0% -->    |
| + SQA        | 2460K   |         | 0        | 128    | 531K   | 21.6% --->   |
| + ESQA       | 12516K  |         | 0        | 1K     | 9425K  | 75.3% -----> |

- Some OMEGAMON commands require additional parameters. Use **PARM=** keyword of the INGOMX command

```
INGOMX EX CMD=SVOL PARM=tved22 NAME=omiimvsb
```

```
>SVOL TVED22 0501 STR/RSNDT FREE(01149,00005) AREAS=0002  
MAX_CNTG(01149,00000)
```

© Copyright IBM Corporation 2019

10-19

### OMEGAMON command examples

On this slide, you see two examples of issuing OMEGAMON for MVS commands: CSAA and SVOL. For readability purposes, the CSAA command is issued from within a NetView PIPE, as shown in the following example. This approach changes the response from many single lines into one multi-line response.

```
pipe netv INGOMX run NAME=OMIIMVSB CMD=CSAA | coll | cons
```

You can also use the NetView WINDOW command to display the output in a readable format. The SVOL command example shows how you can pass command parameters to OMEGAMON by using the **PARM=** keyword of the INGOMX command.

## INGOMX TRAP XTYPE examples

### INGOMX TRAP XTYPE=XREP NAME=omiimvsb

+ XREP Number of Outstanding Replies = 6

### INGOMX TRAP XTYPE=SWPC NAME=omiimvsb

|                     |                               |
|---------------------|-------------------------------|
| + SWPC STC INETD4   | Excessive Swap counts = 101   |
| + SWPC STC FTPD1    | Excessive Swap counts = 101   |
| + SWPC STC BPXOINIT | Excessive Swap counts = 36341 |
| + SWPC STC RV03     | Excessive Swap counts = 101   |
| + SWPC STC RXSERVE  | Excessive Swap counts = 101   |
| + SWPC STC OSNMPD   | Excessive Swap counts = 103   |
| + SWPC STC SNMPQE   | Excessive Swap counts = 2715  |
| + SWPC STC RV02     | Excessive Swap counts = 101   |
| + SWPC STC RV04     | Excessive Swap counts = 101   |
| + SWPC STC RV05     | Excessive Swap counts = 101   |
| + SWPC STC RV01     | Excessive Swap counts = 101   |
| + SWPC STC CANSM2RC | Excessive Swap counts = 9     |
| + SWPC STC CANSM2HD | Excessive Swap counts = 100   |

# Lesson 4. MTR policy definitions

## Lesson 4: MTR policy definitions

- MONITOR INFO
  - Define INGMTRAP as the monitor command for exception-based monitoring
  - Identify the exceptions that you want the monitor to retrieve
- MESSAGES/USER DATA
  - Exception-based monitoring
    - Map exceptions to SA z/OS Health Status
    - Issue commands by using PASSes or CODEs
    - Disable exception during recovery
  - Event-based monitoring
    - Define *monitored object* name. Default setting is MTR name
    - Correlate with OMEGAMON XE situation name

### What this lesson is about

In this lesson we will review how to create MTR policy definitions.

### Objectives

After completing this lesson, you should be able to create MTR policy definitions.

### References

*SC34-2717 System Automation for z/OS Version 4  
Release 1 Defining Automation Policy*

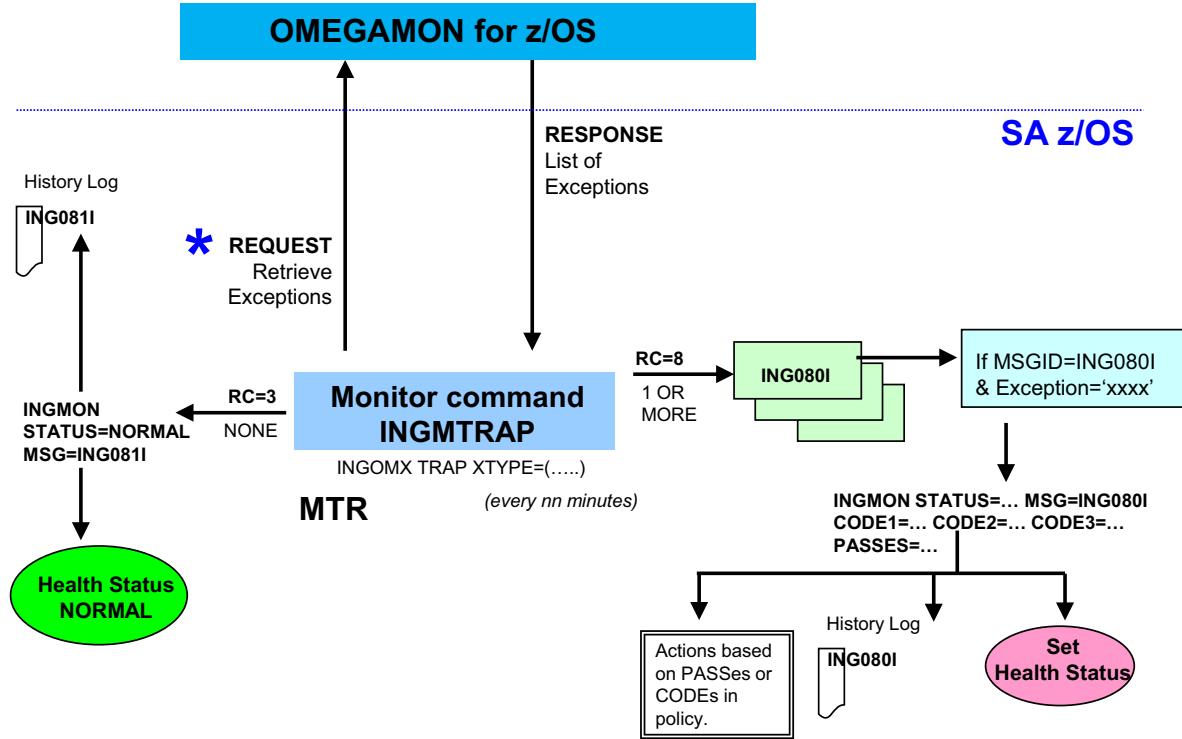
This lesson is about defining monitor resources (MTRs) to monitor for OMEGAMON exceptions. The MESSAGES/USER DATA policy item of the monitor resource (MTR) policy definitions is available so that you can define automation for OMEGAMON exceptions. It provides the same basic capabilities as the MESSAGES/USER DATA policy for Applications (APL).

MESSAGES/USER DATA policy is used for these purposes:

- Set Health Status based on an OMEGAMON exception.
- Define commands to handle the exception.
- Define commands for PASSes **or** CODEs for escalation or selective recovery.
- Temporarily disable exception handling.

If you previously defined an MTR with message overrides for passive monitoring of an Application, then migrate those definitions from the Application policy to the MTR policy.

# Exceptions and Health Status



© Copyright IBM Corporation 2019

10-22

## Exceptions and Health Status

The system administrator defines monitor resources (MTRs) to check for specific exceptions from an OMEGAMON monitor. INGMTRAP is the monitor command and is scheduled periodically, based on the monitor interval, to retrieve exceptions from an OMEGAMON monitor. The exceptions are generated when OMEGAMON thresholds are exceeded, as defined in the OMEGAMON monitor.

The process starts with the monitor command, INGMTRAP, requesting one or more exceptions from the OMEGAMON monitor, as indicated by the asterisk (\*) on the slide. If one or more exceptions are found, you get the following results:

- INGMTRAP issues an ING080I message for each exception and then ends with a DEFER return code (8). The monitor scheduler (AOFRSMTR) accepts the DEFER return code but does not update the Health Status. The Health Status is determined by the MTR policy.
- The Automation Table contains actions to take for the ING080I messages. You can refine the actions by defining PASS counts **or** CODE values in the MTR MESSAGES/USER DATA policy.

Each ING080I message initiates the INGMON command to set the Health Status and issue commands, if defined, to take corrective actions.

If there are no exceptions, message ING081I is issued, and INGMTRAP ends with a NORMAL return code (3). The ING081I message is only written to the monitor resource history log. INGMON is used to set the Health Status to NORMAL. INGMTRAP is rescheduled, based on the monitor interval for the MTR. This process repeats itself at the next time interval.

## MTR policy definitions: Overview

- Use the monitor command INGMTRAP to request list of exceptions by using a session with OMEGAMON
  - If exceptions exist, you issue message ING080I for each or issue message ING081I to indicate no exceptions
- Use policy definitions to create Automation Table statements
  - INGMON command for each ING080I message
- Specify commands in the MESSAGES/USER DATA policy item
  - Invoked by INGMON using PASSes or CODEs definitions
  - Can also be used by the healthstate processing
- Start your own automation routines

### *MTR policy definitions: Overview*

**INGMTRAP** is a specialized command: a monitor command to retrieve one or more exceptions from an OMEGAMON monitor. For each exception, an ING080I message is issued. The ING080I is trapped in the Automation Table and results in the SA z/OS INGMON command to update the Health Status of the resource. You can also define PASSes or CODEs for MTRs to invoke automation routines. Additionally, you can use user commands to take corrective actions.

## INGMTRAP as a monitor command

```

AOFGMTR          Monitor Resource Information
Command ==> _____
Entry Type : Monitor Resource      PolicyDB Name : NEW_PDB
Entry Name  : XREPMONA            Enterprise Name : SAPLEX
Monitored Object . . . . . XREPMONA
Monitored Jobname : . . . . .
Activate Command : . . . . .
Deactivate Command : . . . .
Monitor Command. . . . .
INGMTRAP NAME=OMIIMVSB XTYPE=XREP
Monitoring Interval . . . . . 00:15      (hh:mm)
Captured Messages Limit : . 25        (0 to 999)
Desired Available . . . . .
Monitor for IPL complete. . .
Inform List . . . . . (SDF EIF E2E IOM ITM SMF TTT USR NONE)
Owner . . . . . .
Info Link : . . . . .
Runtokens . . . . .

```

Define INGMTRAP to retrieve OMEGAMON **XREP** exceptions from system MVSB

### *INGMTRAP as a monitor command*

On the slide is an example of an MTR policy definition for an OMEGAMON for MVS monitor. The monitor command, **INGMTRAP NAME=OMIIMVSB XTYPE=XREP**, monitors the OMEGAMON exception for the number of outstanding replies (XREP). INGMTRAP sends a request to the OMEGAMON for MVS monitor each time it is called.

If you want more complex automation, you can write the code for your own monitor command to issue one or more INGOMX TRAP commands and set the Health Status based on the output to the commands. In that case, you define the name of your monitor command on this panel instead of the INGMTRAP command. The MTR name is XREPMONB in this example. XREPMONB is also the default Monitored Object name. The Monitored Object name is used during event-based monitoring.

## INGMTRAP

- Customized interface to retrieve one or more exceptions from OMEGAMON
- Intended for use as a monitor command
  - Retrieve OMEGAMON exceptions
  - If there are no exceptions, Health Status is NORMAL (RC=3) and message ING081I is issued
  - If there are exceptions, Health Status is deferred (RC=8), and message ING080I is issued
- Issue commands and set Health Status as defined in the MTR policy

### INGMTRAP

INGMTRAP is a specialized interface that you can use to retrieve exceptions from an OMEGAMON monitor. INGMTRAP is available as a monitor command, retrieving exceptions and issuing messages for more automation.

INGMTRAP returns a NORMAL or DEFER return code value. If there are any exceptions, then you can use DEFER for other processing that is defined in the MTR policy to set the Health Status of the monitor resource. For example, you can map an exception to a particular Health Status.

INGMTRAP can also use return codes to set a Health Status of BROKEN or FAILED if it encounters errors. You do not typically see those status values. If INGMTRAP detects that no exceptions occurred, an ING081I message is issued. The ING080I is not impacted by automation.

If INGMTRAP detects that exceptions occurred, it issues an ING080I message for each exception. The text of the OMEGAMON exception message is included in the ING080I message. The ING080I messages are used to start automated actions. Both ING080I and ING081I messages are written to the monitor history. You can view them from DISPMTR.

You might see instances in which multiple ING080I messages are generated for an exception. For example, a request to retrieve the OMEGAMON for MVS SWPC (swap count) exception can result in several exceptions:

|                     |                               |
|---------------------|-------------------------------|
| + SWPC STC INETD4   | Excessive Swap counts = 102   |
| + SWPC STC FTPD1    | Excessive Swap counts = 102   |
| + SWPC STC BPXOINIT | Excessive Swap counts = 36842 |
| + SWPC STC RV03     | Excessive Swap counts = 102   |
| + SWPC STC RXSERVE  | Excessive Swap counts = 102   |
| + SWPC STC OSNMPD   | Excessive Swap counts = 104   |
| + SWPC STC SNMPQE   | Excessive Swap counts = 2751  |
| + SWPC STC RV02     | Excessive Swap counts = 102   |
| + SWPC STC RV04     | Excessive Swap counts = 102   |
| + SWPC STC RV05     | Excessive Swap counts = 102   |
| + SWPC STC RV01     | Excessive Swap counts = 102   |
| + SWPC STC CANSM2RC | Excessive Swap counts = 9     |
| + SWPC STC CANSM2HD | Excessive Swap counts = 101   |

In this case, there are 13 lines in the response to the INGMTRAP command. Each one results in an ING080I message. There are 13 ING080I messages issued, one for each exception:

|   |           |
|---|-----------|
| ING080I SWPCMON/MTR/TIVED2 OMIIMVSB OMIIMVS SWPC STC INETD4 | Excessive |
| Swap counts = 102   |           |

In the following list is a detailed explanation of the ING080I message. If you write automation for this message, you must understand the message tokens.

- First token: ING080I.
- Second token: Full monitor resource name, such as XREPMON/MTR/TIVED2.
- Third token: Session\_Name, such as OMIIMVSB.
- Fourth token: The monitor type.

This class uses the OMEGAMON II for MVS monitor. The monitor type is OMIIMVS.

Other possible monitor types are: OMIICICS, OMIIIMS, and OMIDB2. The monitor type is part of the OMEGAMON session attributes that you define when you define the session between SA z/OS and each OMEGAMON monitor.

- Fifth token: Four-character OMEGAMON exception, such as DRDY or XREP.
- Remaining tokens: Exception message from the OMEGAMON monitor.

## Exception-based automation

Use MTR MESSAGES/USER DATA policy for these tasks:

- Map an exception to a Health Status change
- Issue a command based on the exception setting the Health Status
- Define PASSes to issue commands when the Health Status changes
- Define CODEs to issue commands based on the content of the exception message
- Define your own routines to run

### *Exception-based automation*

This slide provides an overview of the possible automation available to you, as described in the following list

- You can map an exception condition to a Health Status. For example, if an XREP exception is detected, then change the Health Status to WARNING.
- You can define a command to be run every time the exception is detected.
- You can define commands to be run based on PASSes, which provides an escalation vehicle for your commands.
- You can define CODEs and commands, based on those CODEs. You can run commands based on the content of the exception message.
- You can define Automation Table overrides to call your own automation routines.

## MESSAGES/USER DATA example

| AOFGDYNM  | Message Processing       | Line 00000001 Col 001 075 |
|---|--------------------------|---------------------------|
| Command ==>   |                          | Scroll ==> CSR            |
| Entry Type : Monitor Resource   | PolicyDB Name : NEW_PDB  |                           |
| Entry Name : XREPMONA   | Enterprise Name : SAPLEX |                           |
| Line Commands: S/C (Cmd), R (Rep), K (Cod), U (Usr), A,M (AT, MRT, MPF) |                          |                           |
| I, D (insert or delete lines)   |                          |                           |
| Message ID field length. . 16 (1 - 32)                                  |                          |                           |
| Cmd Message ID  | Description              | C R K U A M               |
| a + XREP  | Warning                  | *                         |

- Message ID: Six characters  
First character must be a plus sign (+) followed by a blank, and the four-character exception ID
- A (AT)
  - AS: Map exception to Health Status
  - AO: Modify Automation Table statements. If you use AO, then you must set the Health Status by using INGMON called from the Automation Table
- C (CMD): Define commands to issue when Health Status changes and for PASSes or CODEs
- K (COD): Map CODEs to values
- U (USR): Define DISABLETIME value

### MESSAGES/USER DATA example

On this slide, you see an overview of the functions that are available for MTR resources by using the MESSAGES/USER DATA panel. These functions are described in the following list:

- **Message ID:** Used to define the four-character exception ID from OMEGAMON, preceded by a plus sign (+) and a blank. For example, + XREP. The blank is used by SA z/OS to distinguish between messages and exceptions.

Use this format to define automation for OMEGAMON exceptions. Do not define an ING080I message for automation. ING080I is used internally by SA z/OS processing.

- **AT:** Maps an exception to a Health Status and supports user-defined Automation Table overrides.
- **CMD:** Maps an exception to a Health Status and includes a definition to run one or more commands.
- **CODE:** Provides support for defining codes within the message that can drive automation.
- **USER:** Provides support for you to define user keywords and values, such as DISABLETIME.

## Map OMEGAMON exception to Health Status

```

AOFGMAT2          AT Status Specification
Command ==> _____
View or change specifications for message: + XREP
Application / MVC Component status      Monitor status
  ACTIVE   Starting    NORMAL  Resource shows good results
  UP       Available   WARNING Resource shows degradation
  HALTED  Degraded    MINOR   More severe than WARNING
  Terminating Received STOP command   CRITICAL More severe than MINOR
  Terminated Terminated normally     FATAL    More severe than CRITICAL
  ABENDING Going to end abnormally
  Abended  Ended abnormally
  BREAKING Non-recoverable abending
  BROKEN   Non-recoverable abend

Capture specification
  Capture  Capture Message
  DOM . . .

```

Set the Health Status  
to WARNING when an  
XREP exception is  
detected

### Map OMEGAMON exception to Health Status

Use the AT action to map an exception to a Health Status change. This step is the first one in defining an Automation Table action for whenever the monitor detects an XREP exception. The XREP exception creates an ING080I message, which initiates the Automation Table to call INGMON to set the Health Status. Select the appropriate status, such as WARNING, and press PF3 to set the Health Status to WARNING every time an XREP exception is detected.

## Issue a command based on an exception

```
AOFGDYN7      Command Processing : line 1
Command ===> █                               Scroll ===> CSR
This panel displays the complete input fields of a single data line.
Pass/Selection . . . . . 1
AutoFn/* . . . . . . . .
Command Text . . . . .
msg all,XREPMON monitor status is WARNING. Check number of outstanding replies
```

When an XREP exception is detected, the following actions occur:

1. XREPMON Health Status is set to WARNING (from previous slide)
2. MSG ALL command is run each time because no PASS is defined

### *Issue a command based on an exception*

In addition to mapping exceptions to Health Status changes, you can also define one or more commands. In this example, an MSG ALL command is issued every time an XREP exception is detected.

## Issue a command based on an health state

```
AOFGDYNC      Command Processing : HEALTH_RECOVERY      Line 00000001 Col 001 075
Command ===> _____ Scroll ===> CSR

Mixed case . . . YES (YES NO)

Cmd State      AutoFn/* Command Text
WARNING        ALRT monitor &EHKVAR1 changed from &EHKVAR3 to &EHKVAR2
```

Alternate option to issue commands:

When the XREPMON Health Status is set to WARNING, the ALRT REXX is called:

- ALRT monitor &EHKVAR1 changed from &EHKVAR3 to &EHKVAR2

Note the usage of &EHKVARx variables

### *Issue a command based on an health state*

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. It is an alternate option to issue commands:

- When the XREPMON Health Status is set to WARNING, the ALRT REXX is called:
- ALRT monitor &EHKVAR1 changed from &EHKVAR3 to &EHKVAR2

Note the usage of &EHKVARx variables:

- &EHKVAR1 Contains the monitor name
- &EHKVAR2 Contains the current health status
- &EHKVAR3 Contains the old health status
- &EHKVAR4 Contains the message that is associated with the health status
- &EHKVAR5 Contains the object name of the monitor
- &EHKVAR6 Contains the job name
- &SUBSAPPL Contains the monitor name
- &SUBSTYPE Contains the string MONITOR

# Using PASS counts for escalation

```
A0FGDYNC      Command Processing : + XREP          Line 00000001 Col 001 075
Command ===> _____ Scroll ===> CSR
Mixed case . . . YES (YES NO)
Cmd Ps/Select AutoFn/* Command Text
____ 1 _____ msg_all,XREPMON monitor status is WARNING. Check number
____ 3 _____ msg_netop1,XREPMON monitor escalation. WTOR count still
***** Bottom of data *****
```

- Three PASSes are defined with two commands
- When an XREP exception is detected, the following actions occur:
  - XREPMON Health Status is set to WARNING
  - On PASS1, the MSG ALL command is run
  - On PASS2, no action is taken
  - On PASS3, the MSG NETOP1 command is run

## Using PASS counts for escalation

If you require more granularity, you can define PASSes to issue different commands or no commands. PASSes are typically used to escalate a potential problem. In this example, the first time the XREP exception is detected, an **MSG ALL** command is issued. The second time an XREP exception is detected, the Health Status is set and no action is taken. The third time an XREP exception is detected, an **MSG NETOP1** command is issued. Using this simple example, you can define a command for the third pass to perform an action such as escalating the exception to a management team member.

## Suspending exception analysis

```
A0FGDYN8      User Data Processing : + XREP          Line 00000001 Col 001 075
Command ===> _____           Scroll ===> CSR
Mixed case . . . YES (YES NO)          Keyword length. . . 20 (1-64)
Cmd  Keyword          Data
____ DISABLETIME       00:15
_____
_____
_____
_____
_____
```

- While automation occurs, you might have to suspend further invocations of the monitor command  
Prevents further exceptions from starting automation while the automation is processing an exception
- Define a USER action for the DISABLETIME keyword in MESSAGES/USER DATA policy item

### *Suspending exception analysis*

Depending on your monitor interval, you might have to suspend monitoring to have sufficient time for your commands to complete before querying for exceptions again. To suspend monitoring, define a time interval for the predefined keyword DISABLETIME. In this example, automation is paused for 15 minutes. The monitor command is rescheduled when the 15-minute interval expires.

Using DISABLETIME is helpful when the actions from a command might take longer to complete. By defining DISABLETIME you prevent other automation such as escalation processing from occurring. Then, sufficient time is available for the command to complete. The DISABLETIME keyword is defined on the User Defined Data panel. You can access this panel with the **user** action on the Message Processing panel in the monitor definition.

# MESSAGES/USER DATA summary of PASSes

```

A0FGDYNM          Message Processing      Line 00000001 Col 001 075
Command ==>          PolicyDB Name : NEW_PDB
Entry Type : Monitor Resource   Enterprise Name : SAPLEX
Entry Name : XREPMONA
Line Commands: S/C (Cmd), R (Rep), K (Cod), U (Usr), A,M (AT, MRT, MPF)
              I, D (insert or delete lines)
Message ID field length. . 16 (1 - 32)

Cmd Message ID      Description          C R K U A M F
+ XREP             Warning            2 1 * ****
***** Bottom of data *****
```

For XREP exceptions involving this monitor, you have these items:

- One AT entry to set the Health Status
- Two commands defined for PASSes 1 and
- One user entry defined to set the DISABLETIME

## MESSAGES/USER DATA summary of PASSes

This example illustrates the way that you define a monitor resource, as follows:

- Monitor for XREP exception and set the Health Status to WARNING when an exception is detected.
- Issue an MSG ALL for the first exception.
- Take no command action for the second exception.
- Issue an MSG NETOP1 for the third exception, for escalation.
- Set the DISABLETIME to 15 minutes.

## Automation Table fragment for ING080I

| AOFGMADV  | Message Automation Overview | Scroll ==> CSR             |
|---|-----------------------------|----------------------------|
| <b>Command ==&gt;</b>   |                             |                            |
| AC AT condition   | MS MRT action selection     | MF MPFLSTxx specifications |
| AS AT status  | MO MRT override             | IG Ignore for AT, MRT, MPF |
| AO AT override  |                             |                            |
| <b>Message ID :</b> + XREP  |                             |                            |
| <pre>Generated AT entry : IF TOKEN(5) = 'XREP' THEN   EXEC(CMD('INGMON 'MTRNM', STATUS=WARNING') ROUTE(ONE %AOFGOPGSSOPER%));</pre> |                             |                            |
| <pre>Generated MRT entry : UPON ( MSGID = 'ING080I' ) REVISE('Y' AUTOMATE)</pre>  |                             |                            |
| <pre>Generated MPF entry : + XREP</pre>   |                             |                            |

AT fragment is generated for XREP exceptions; Health Status is set to WARNING.

### Automation Table fragment for ING080I

This Automation Table fragment was created for the XREP monitor. When message ING080I is generated for XREP exceptions, INGMON is called to set the Health Status to WARNING.

## CODEs for selective processing

Selective processing involves the following three steps:

1. Define CODEs and their values
2. Define commands to issue, based on the CODEs values
3. Define Automation Table override to call INGMON command with parameters
  - MSGTYPE=exception (token 5)
  - CODE1, CODE2, CODE3

### CODEs for selective processing

You can use CODEs definitions to define more granular automation to take place. For example, you can define CODEs for anticipated values of an insert within an exception message and then define commands to run for each value.



**Note:** CODEs and PASSes are mutually exclusive.

## Example: ING080I tokens for XREP

| Token # | Value              |
|---------|--------------------|
| 1       | ING080I            |
| 2       | XREPMON/MTR/TIVED2 |
| 3       | OMIIMVSB           |
| 4       | OMIIMVS            |
| 5       | XREP               |
| 6       | Number             |
| 7       | of                 |
| 8       | Outstanding        |
| 9       | Replies            |
| 10      | =                  |
| 11      | 6                  |

Pass the number of replies as **CODE1**

### Example: ING080I tokens for XREP

Using the XREP exception as shown in this example, you can see a complete ING080I message and each token in the message, as shown on this slide. The fifth token contains the exception ID (XREP). The eleventh token is highlighted because it contains the current number of outstanding WTORs. In this case, there are **6** WTORs. You can define CODEs to automate based on the number of WTORs. This process involves defining an Automation Table override, which is the reason that you must understand the message tokens.

## Example: Define CODEs for XREP

| COMMANDS                   |        | HELP                     |   |                |
|----------------------------|--------|--------------------------|---|----------------|
| AOFGDYN9                   |        | Code Processing : + XREP | Line 00000001 Col 001 075<br>Scroll ==> CSR |                |
| Command                    | =      |                          |   |                |
| Cmd                        | Code 1 | Code 2                   | Code 3                                      | Value Returned |
| 0                          |        |                          |   | OK             |
| 1                          |        |                          |   | ONE            |
| 2                          |        |                          |   | TWO            |
| 3                          |        |                          |   | THREE          |
| 4                          |        |                          |   | FOUR           |
| 5                          |        |                          |   | FIVE           |
| *                          |        |                          |   | TOOMANY        |
| ***** Bottom of data ***** |        |                          |   |                |

- Code 1 = ING080I token 11 = number of WTORs
- This example defines a threshold for WTORs
  - Status is *good* if less than or equal to 5 WTORs  
Returning value from OK to FIVE
  - Status is *bad* if more than 5 WTORs (value returned is TOOMANY)

© Copyright IBM Corporation 2019

10-37

### Example: Define CODEs for XREP

You can use the Code Processing panel to define values to be set based on the tokens within the ING080I message. In this example, there are seven values for Code 1 defined. The last one is an asterisk (\*), which you use as a default value. The asterisk functions as an upper threshold and returns the value TOOMANY when there are six or more outstanding WTORs. In this example, you see how to automate when more than five WTORs are detected by the exception.

This example generates these policy statements for the XREPMON monitor:

```
CODE=(0,,, "OK")
CODE=(1,,, "ONE")
CODE=(2,,, "TWO")
CODE=(3,,, "THREE")
CODE=(4,,, "FOUR")
CODE=(5,,, "FIVE")
CODE=(*,,, "TOOMANY")
```

You press PF3 to save the code definitions.

## Example: Define commands for XREP

|                                     |  |                           |
|-------------------------------------|--|---------------------------|
| AOFGDYN9                            | Command Processing : + XREP                            | Line 00000001 Col 001 075 |
| Command ==>                         |  | Scroll ==> PAGE           |
| <hr/>                               |  |                           |
| Cmd Ps>Select AutoFn/* Command Text |  |                           |
| <hr/>                               |  |                           |
| 1                                   | MSG ALL,XREPMON STATUS IS WARNING. CHECK NUMBER        |                           |
| 3                                   | MSG NETOP1, XREPMON MONITOR ESCALATION - WTOR COUNT IS |                           |
| FIVE                                | MSG NETOP1, XREPMON DETECTED 5 WTORS                   |                           |
| FOUR                                | MSG NETOP1, XREPMON DETECTED 4 WTORS                   |                           |
| THREE                               | MSG NETOP1, XREPMON DETECTED 3 WTORS                   |                           |
| TWO                                 | MSG NETOP1, XREPMON DETECTED 2 WTORS                   |                           |
| ONE                                 | MSG NETOP1, XREPMON DETECTED 1 WTORS                   |                           |
| OK                                  | MSG NETOP1, XREPMON IS LOOKING GOOD                    |                           |
| TOOMANY                             | MSG NETOP1, XREPMON DETECTED TOOMANY WTORS*WARNING*    |                           |
| -                                   | MSG ALL, XREPMON MONITOR STATUS IS WARNING. CHECK      |                           |
| <hr/>                               |  |                           |

- Based on the defined CODEs and values that are returned
- Based on the number of WTORS when an XREP exception is detected
  - 0: Command defined for OK is run
  - 1: Command defined for ONE is run for each code value
  - 6 or more: Command defined for TOOMANY is run
  - The last blank Select command is the default command

© Copyright IBM Corporation 2019

10-38

### Example: Define commands for XREP

After you define the codes and the values that each code returns you use the Command Processing panel to define a command to run, based on the CODE value returned. On this slide you see commands that are defined for each CODE1 value. A default command (MSG ALL) is issued every time an XREP exception is detected.

The complete set of definitions is like the following example:

```
CMD= (OK,, 'MSG NETOP1,XREPMON STATUS IS OK.')
CMD= (ONE,, 'MSG NETOP1,XREPMON DETECTED 1 WTOR')
CMD= (TWO,, 'MSG NETOP1,XREPMON DETECTED 2 WTORS')
CMD= (THREE,, 'MSG NETOP1,XREPMON DETECTED 3 WTORS')
CMD= (FOUR,, 'MSG NETOP1,XREPMON DETECTED 4 WTORS')
CMD= (FIVE,, 'MSG NETOP1,XREPMON DETECTED 5 WTORS')
CMD= (TOOMANY,, 'MSG NETOP1,*** ATTENTION *** XREPMON WTOR THRESHOLD HAS BEEN
EXCEEDED!')
CMD= (,, 'MSG ALL,XREPMON MONITOR STATUS IS WARNING. CHECK NUMBER OF OUTSTANDING
REPLIES.')
```

Notice that the last command does not match a code value returned.

## Example: AT overrides for XREP

**Message Automation Overview**

**AOFGMADV**  
Command ==>

**AC AT condition**  
**AS AT status**  
**AO AT override**

**Message ID : + XREP**

**AT override :**  
**IF TOKEN(5) = 'XREP' & TOKEN(11) = NUMREPLIES THEN**  
**EXEC(CMD('INGMON MTRNM', STATUS=WARNING, MSGTYPE=XREP, CODE1=NUMREPLIES'))**  
**ROUTE(ONE %AOFOPGSSOPER%)**  
**EXEC(CMD('MSG NETOP1,USER AUTOMATION GOES HERE'))**  
**ROUTE(ONE %USROPER1%));**

**Generated MRT entry :**  
**UPON ( MSGID = 'ING080I' )**  
**REVISE('Y' AUTOMATE)**

**Token 11 = Number of WTORS**  
**MS MRT action selection**  
**MO MRT object ID**  
**Parse in AT condition**  
**Add as CODE1 value for INGMON**

**MF MPFLSTxx specifications**  
**IG Ignore for AT, MRT, MPF**

**Use your own autotask**

**Define additional AT action**

**Token 11** is the number of replies  
Pass it to INGMON as **CODE1**  
Additionally, define the second AT action for user automation.

© Copyright IBM Corporation 2019

10-39

*Example: AT overrides for XREP*

You can use the Automation Table (AT) override action (AO) to modify the default Automation Table statement. Based on the examples used in this lesson, automation is defined for an XREP exception from OMEGAMON to use the INGMON command. Because CODEs are also defined, you must modify the INGMON command as follows:

- On the first NetView AT action:
    - Pass the MSGTYPE=XREP and CODE1=*numreplies* parameters to tell INGMON to check the CODE definitions and issue the appropriate commands.
    - STATUS=WARNING must also be passed to INGMON to set the Health Status because you cannot use OVR and AUTO definitions at the same time.
  - (Optional) Define another Automation Table action, by using the MSG NETOP1 command.

# MESSAGES/USER DATA summary of CODEs

| AOFGDYNY<br>Command ==>  | Message Processing                                  | Line 00000001 Col 001 075<br>Scroll ==> CSR   |   |   |   |   |   |   |   |    |   |   |   |  |  |  |
|--|---|---|---|---|---|---|---|---|---|----|---|---|---|--|--|--|
| Entry Type : Monitor Resource<br>Entry Name : XREPMONA   | PolicyDB Name : NEW_PDB<br>Enterprise Name : SAPLEX |   |   |   |   |   |   |   |   |    |   |   |   |  |  |  |
| Line Commands: S/C (Cmd), R (Rep), K (Cod), U (Usr), A,M (AT, MRT, MPF)<br>I, D (insert or delete lines) |   |   |   |   |   |   |   |   |   |    |   |   |   |  |  |  |
| Message ID field length . . 16 (1 - 32)  |   |   |   |   |   |   |   |   |   |    |   |   |   |  |  |  |
| Cmd Message ID<br>+ XREP   | Description<br>Warning                              | <table border="1"> <tr> <td>C</td><td>R</td><td>K</td><td>U</td><td>A</td><td>M</td><td>F</td></tr> <tr> <td>10</td><td>7</td><td>1</td><td>*</td><td></td><td></td><td></td></tr> </table> | C | R | K | U | A | M | F | 10 | 7 | 1 | * |  |  |  |
| C  | R   | K   | U | A | M | F |   |   |   |    |   |   |   |  |  |  |
| 10   | 7   | 1   | * |   |   |   |   |   |   |    |   |   |   |  |  |  |
| ***** Bottom of data *****   |   |   |   |   |   |   |   |   |   |    |   |   |   |  |  |  |

For XREP exceptions involving this monitor, you have these items:

- Seven CODEs defined, each with a command plus a default select command for a total of eight
- Two additional first and third pass commands for a total of **10**
- One user entry defined to set the DISABLETIME
- Automation Table definitions: One override plus set Health Status
- One MRT entry to automate the + XREP message

## MESSAGES/USER DATA summary of CODEs

In summary, you use these steps to define a monitor resource as follows:

- Monitor for XREP exceptions with CODEs values for the number of WTORS that are currently outstanding. Based on the XREP exception message in the ING080I message, you use token 11.
- Issue two AT actions for every XREP exception that is detected:
  - Issue **INGMON XREPMON STATUS=WARNING,MSGTYPE=XREP,CODE1=numreplies** to set the Health Status and use the CODEs processing.
  - Issue an **MSG NETOP1** action to simulate a user-defined command to be run for more automated actions.
- Issue one of seven commands based on the CODEs definitions.
- Issue an MSG ALL (the eighth command) every time.
- Issue commands for pass1 and pass3.
- Set the DISABLETIME to 15 minutes. (This step was not shown in this sequence, but it is important to remember when you define commands.)

## Example: XREP automation

When an XREP exception occurs

**ING080I** XREPMON/MTR/TIVED2 OMIIMVS OMIIMVS XREP Number of Outstanding Replies = **6**

The messages to NETOP1 are

**DSI039I** MSG FROM AUTWRK04 : \*\*\* ATTENTION \*\*\* XREPMON WTOR THRESHOLD HAS BEEN EXCEEDED!

[From INGMON CODE1=6 \(TOOMANY\) processing.](#)

**DSI039I** MSG FROM AUTWRK04 : XREPMON MONITOR STATUS IS WARNING. CHECK NUMBER OF OUTSTANDING REPLIES.

[From the command definitions, as a default to be run every time.](#)

**DSI039I** MSG FROM AUTWRK04 : \*\*\*User automation would go here

[From the second AT action, called \*user automation\*.](#)

### Example: XREP automation

This slide illustrates an example of an XREP exception that is detected by the defined XREPMON monitor. When the exception is detected, message ING080I is generated and invokes the following three commands:

- INGMON XREPMON,STATUS=WARNING,MSGTYPE=XREP,CODE1=6

A result of the AT override that was coded.

- MSG ALL,XREPMON MONITOR STATUS IS WARNING. CHECK NUMBER OF OUTSTANDING REPLIES.

A result of the command that was coded for every XREP exception that was detected.

- MSG NETOP1,\*\*\*User automation would go here.

A result of the second AT override that was coded to simulate calling a routine to provide your own automated actions.

## Define relationships

- Define HasParent relationship between the monitor resource and the OMEGAMON Application; for example, XREPMON and CANSM2RC

```

AOFGXRE0                               Define Relationship
Command ==> _____
Entry Type : Monitor Resource          PolicyDB Name   : NEW_PDB
Entry Name  : XREPMONA                 Enterprise Name: SAPLEX
Monitor Name      : XREPMONA
Description. . . .
Relationship Type. . HASPARENT          MAKEAVAILABLE MAKEUNAVAILABLE
                                         PREPAVAILABLE PREPUNAVAILABLE
                                         HASPARENT HASPASSIVEPARENT
                                         HASMONITOR PEEROF
                                         FORCEDOWN EXTERNALLY
Supporting Resource  CANAM2RC/APL/=    Resource Name
Sequence Number. . . .                  Sequence Number (1-99,blank)
Automation . . . . .                   ACTIVE PASSIVE
Chaining . . . . .                    STRONG WEAK
Condition . . . . .                   Satisfy condition
                                         (? for list of possible values)

```

- Define relationships to ensure that the monitor resource is started and stopped when the OMEGAMON application starts and stops

### Define relationships

Consider defining relationships between your SA z/OS monitor resources (for example, XREPMON) and the OMEGAMON monitor Applications (for example, CANAM2RC). The relationships ensure that the SA z/OS monitor resource starts after the OMEGAMON monitor Application is active and ends when the OMEGAMON monitor Application ends.

Otherwise, you might encounter situations in which the SA z/OS monitor resource attempts to start the session to the OMEGAMON monitor Application before the OMEGAMON monitor Application is active. This situation sets the session status to **SESSFAIL**, and an operator must manually restart it.

## Clear monitor status

```
INGMON XREPMON MSGTYPE=XREP CLEARING=YES
STATUS=NORMAL MSG='XREP exception corrected'
```

If automation is successful, use **INGMON** to generate a *clearing event*

- Reset the PASS count (if used)
- Cancel DISABLETIME processing
- Reset the Health Status to NORMAL
- Define message text for operator: *XREP exception corrected*

### *Clear monitor status*

When the condition is resolved, you can reset all automation by issuing an INGMON command for the exception (XREP) with the CLEARING=YES parameter. MSGTYPE identifies the OMEGAMON exception.

#### **CLEARING=YES**

- Resets the PASS count.
- Cancels any DISABLETIME processing.
- Resumes monitoring of the resource again, if monitoring was suspended.

#### **STATUS=NORMAL**

- Resets the Health Status to NORMAL.

#### **MSG=**

- Provides a descriptive message that can be seen on the DISPMTR panel.

# Lesson 5. Security considerations

## Lesson 5: Security considerations

- OMEGAMON security
- NetView command authorization
- Session password management

### What this lesson is about

This lesson provides an overview of security considerations for OMEGAMON.

### Objectives

After completing this lesson, you should be able to describe several security related considerations for OMEGAMON and System Automation.

### References

*SC34-2647 System Automation for z/OS Version 4 Release 1 User's Guide*  
*SC34-2716 System Automation for z/OS Version 4 Release 1 Planning and Installation*

You can secure the access to the OMEGAMON sessions and the commands that you issue. In this lesson, you learn about the tasks that are required for this security.

## OMEGAMON security and SA z/OS

- The user ID for the SA z/OS session
  - Define it internally in the OMEGAMON security table or externally to a SAF product
  - Ensure that someone can access an INITIALn profile to connect and issue commands
  - Grant the highest level of security defined for the installation
- Use NetView command authorization to control access to the session and access to the commands

© Copyright IBM Corporation 2019

10-45

### OMEGAMON security and SA z/OS

You must define the session user ID to OMEGAMON security to control access to the OMEGAMON monitors. You can use NetView command authorization to control who has access to the session and to the commands that someone can issue.

OMEGAMON supports two types of security:

- Product-level security is applied when users log on to OMEGAMON.
- Command-level security is applied when users issue commands.

You can define each type in the following ways:

- Internally by using an OMEGAMON security table. The security table provides four levels of security (0, 1, 2, and 3).
  - Assign commands to a level.
  - Grant the User ID access to a profile, INITIALn, where n is one of the four security levels.

You do not grant a User ID access to the INITIAL profile. If the security level of the user ID is less than the command security level, the command is rejected.

- Externally by using a SAF product.

- The user ID and commands must be secure.
- You can secure the commands internally, externally, or both.

Refer to the OMEGAMON publications for more details.

## NetView command authorization

- You can use the NetView Command Authorization Table (CAT) to control access to these items:
  - INGSESS to manage OMEGAMON sessions
  - INGOMX for OMEGAMON commands and sessions
- You can try this approach:
  - Protect access to all sessions and all commands
  - Permit access to sessions and commands only as needed

### NetView command authorization

NetView provides command and parameter security through the use of the NetView Command Authorization Table (CAT). For SA z/OS, you can control operator access to items such as starting sessions, stopping sessions, and issuing the OMEGAMON commands. Alternatively, you can use an SAF product instead of the CAT table.

By using the CAT table for security, you restrict access to all sessions and all commands. You can accomplish this task by using a **PROTECT** statement with a syntax like the following one:

```
netid.luname.clist_name.parm.parm_value
```

For example:

```
PROTECT *.AOFDA.INGOMX.*.* will restrict access to all commands.
```

```
PROTECT *.AOFDA.INGSESS.*.* will restrict access to the starting and stopping of  
the OMEGAMON sessions.
```

Then, you can grant access to the functions by using a **PERMIT** statement. You might want to define your operators to a group by using the GROUP statement and then grant access to the group of operators. This method is helpful if you have a large number of operators to define.

Administrators need the following access:

```
*.*.INGROMX0.NAME.session  
*.*.INGSESS.REQ.START  
*.*.INGSESS.REQ.STOP
```

Monitor tasks need the following access:

```
*.*.INGROMX0.NAME.session  
*.*.INGROMX0.CMD.EXSY for MVS, CICS, and DB2  
*.*.INGROMX0.CMD.XIMS for IMS
```

Some OMEGAMON commands contain a period (.) as a character in the command name; for example,.RMF. You cannot use periods in the CAT table. Instead, use an at symbol (@) as shown in the following example:

```
PROTECT *.*.INGROMX0.CMD.@RMF
```

## CAT example

- First, protect access to the session
  - PROTECT \*.\*.INGROMX0.NAME.OMIIMVSB
- Second, allow group OMOPERS to access the session
  - GROUP OMOPERS OPER1,OPER2
  - PERMIT OMOPERS \*.\*.INGROMX0.NAME.OMIIMVSB
- Third, allow access to commands
  - PERMIT OMOPERS \*.\*.INGROMX0.CMD.KILL

### *CAT example*

On this slide, you see an example of several CAT entries that you can use for these purposes:

- Restrict access to the session with OMEGAMON for MVS (OMIIMVSB).
- Define two operators (OPER1 and OPER2) to the OMOPERS group.
- Grant the OMOPERS group access to the session, which grants access to the session for OPER1 and OPER2.
- Grant the OMOPERS group access to the OMEGAMON KILL command. The OPER1 and OPER2 issue the KILL command.

You must grant the operators in OMOPERS access to several sessions and to the OMEGAMON commands.

# Session passwords

- Session passwords are defined in the AUTHENTICATION item of the Network (NTW) policy
- The passwords can be stored in these locations:
  - SA z/OS policy
    - As plain text, not encrypted
  - NetView password data set, EZLPSWD
    - In an encrypted format
    - Managed with GETPW command
- GETPW command is used to access the password data set to set or read the password
  - Session user IDs must also have authority to issue GETPW

|                           |          |
|---------------------------|----------|
| AOFGOSOA                  | OMEGAMON |
| Command ==>               | _____    |
| Entry Type : Network      | P        |
| Entry Name : BASE_NETWORK | E        |
| Session Name : OMIIMVSB   |          |
| VTAM Applid . . . A02M2RC | Name o   |
| VTAM Logmode . . .        | Logmod   |
| TAF SRCLU . . .           | Name o   |
| Type . . . . . OMIIMVS    | (OMIIC   |
| User ID. . . . .          | User I   |
| Password . . . . . SAFPW  | Passwo   |
| Timeout. . . . .          | Time t   |
| Session Data . . . . .    |          |

## Session passwords

The passwords used by the session user ID to connect from SA z/OS to the OMEGAMON monitors are defined in the Network (NTW) policy. If you specify SAFPW in the policy, then SA z/OS uses the NetView password data set, EZLPSWD, and the GETPW command to create encrypted passwords that are used for the OMEGAMON sessions.

If access to the GETPW command is restricted, you must permit the proxy operators access to it. If you do not, the session starts fail and the session has an AUTHFAIL status. The syntax of the GETPW command is as follows:

```
GETPW user_ID owner group_name READ|INIT|DELETE|UPDATE
```

where

- User\_ID is the OMEGAMON user name, defined in the User ID field of the OMEGAMON Session Attributes panel when defining the Network policy.
- Owner is up to five characters that identify the VTAM APPL ID. By default, these characters are the leftmost five characters of the *VTAM Applid* defined on the OMEGAMON Session Attributes panel. If no owner is defined for an Application ID, the five leftmost characters of the Application

ID are used by default. This ID can also be your NetView domain name. Passwords are stored and retrieved by user ID and owner.

# Lesson 6. SA z/OS monitoring agent

## Lesson 6: SA z/OS monitoring agent

- Monitoring agent collects system and sysplex data for display in workspaces on the Tivoli Enterprise Portal
- Automation Manager workspaces
  - Resource overview and detail
  - Resource request information
  - Resource Topology (new in System Automation for z/OS 3.4)
  - Automation environment information
  - User-defined status items
  - Monitor resources
  - Outstanding WTORs
- Automation Agent workspaces
  - Automation Agent details
  - Automation statistics
  - OMEGAMON sessions
  - Gateway connections
  - Critical messages

**Note:** The announcement of IBM Service Management Suite for z/OS, V1.5.0 contains a statement of general direction that IBM intends to discontinue System Automation for z/OS support for collecting automation data using the Monitoring Agent and feeding that data into the Tivoli Enterprise Portal (TEP).

### What this lesson is about

This lesson reviews the use of the System Automation for z/OS monitoring agent.

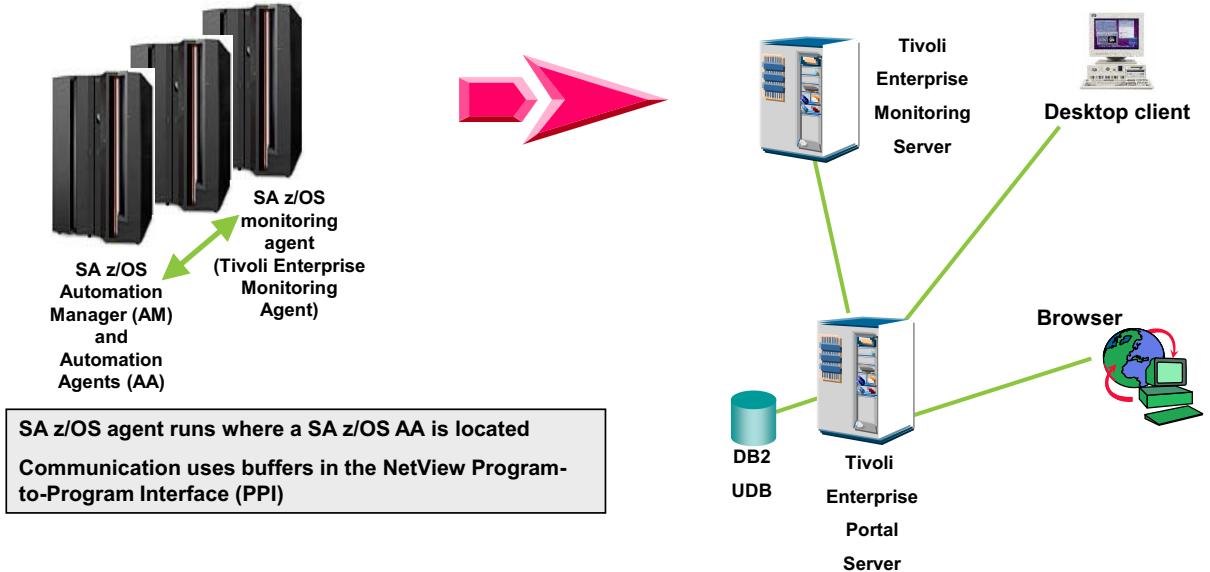
### Objectives

After completing this lesson, you should be able to implement and use the System Automation for z/OS OMEGAMON monitoring agent.

### References

*SC34-2716 System Automation for z/OS Version 4 Release 1 Installation and Planning*  
*SC24-2653 IBM System Automation for z/OS: Monitoring Agent Configuration and User's Guide*

# Tivoli Enterprise Portal architecture



**Tivoli Enterprise Monitoring Server can run on z/OS or distributed platforms**

**Tivoli Enterprise Portal Server and Tivoli Enterprise Monitoring Server can be in the same machine, if on a distributed platform**

**Remote Tivoli Enterprise Monitoring Server connected to a hub Tivoli Enterprise Monitoring Server is also supported**

© Copyright IBM Corporation 2019

10-50

*Tivoli Enterprise Portal architecture*

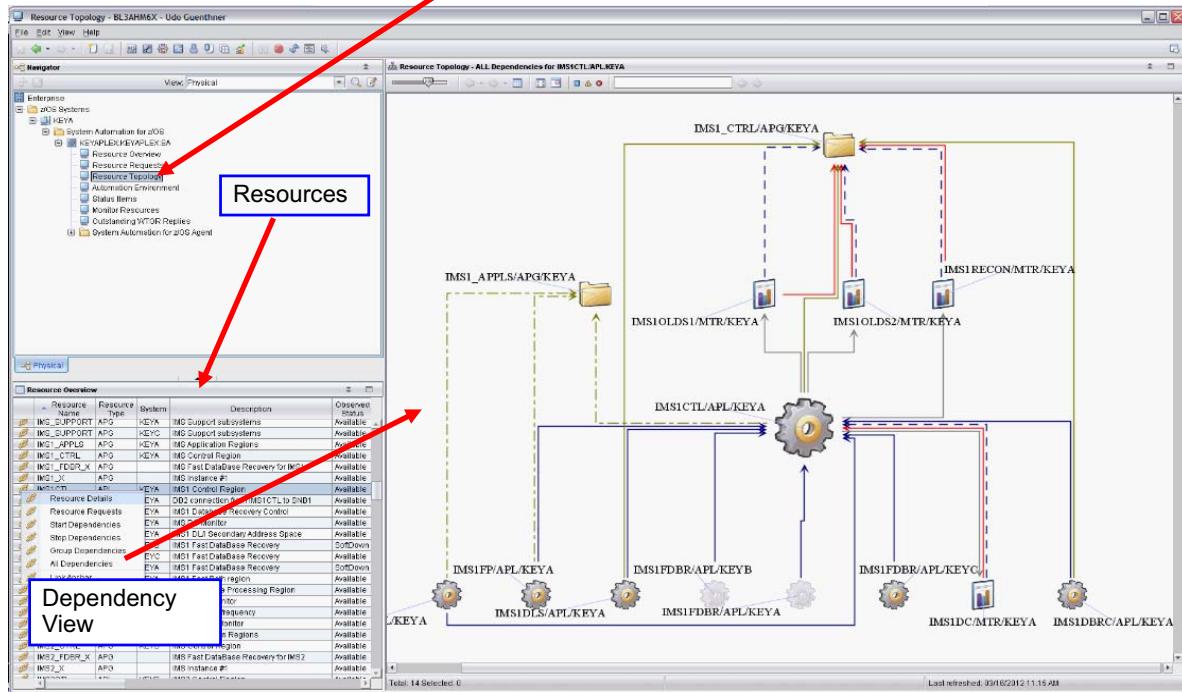
The SA z/OS monitoring agent is a Tivoli Enterprise Monitoring Agent (TEMA).



**Note:** The announcement of IBM Service Management Suite for z/OS, V1.5.0 contains a statement of general direction that IBM intends to discontinue System Automation for z/OS support for collecting automation data using the Monitoring Agent and feeding that data into the Tivoli Enterprise Portal (TEP).

# Tivoli Enterprise Portal topology view

This Tivoli Enterprise Portal workspace is called **Resource Topology**



© Copyright IBM Corporation 2019

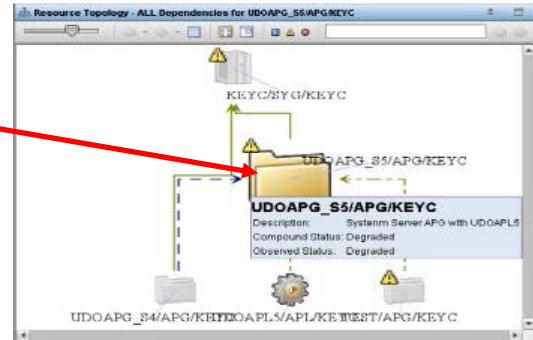
10-51

## Tivoli Enterprise Portal topology view

This slide shows the Tivoli Enterprise Portal topology workspace. The workspace is called **Resource Topology**, as illustrated on the slide. In the lower left pane is a table of the resources. You can right-click a resource entry to get a list of possible dependency views. An example is **All Dependencies**. From that selection, you can see a detailed view of the dependencies in the right pane.

# Tivoli Enterprise Portal topology view nodes and connections

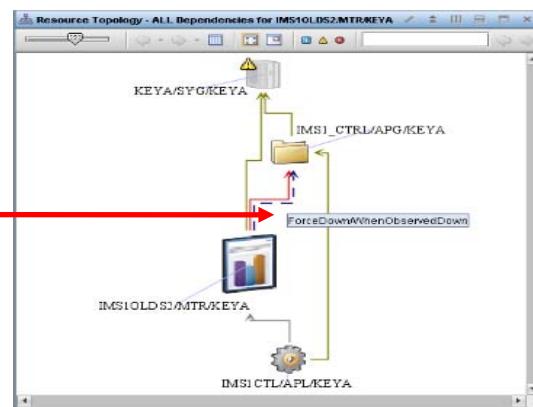
- Nodes
  - Each resource is a node
    - Name in AM notation
    - Has status decorator
    - Can be greyed-out
    - Has flyover text
  - One resource is in focus



- Connections

Each relationship is a connection

- Has a label (initially hidden) and flyover
- Color and stipple
- Membership as Is...Member
  - Basic
  - Move
  - Server



© Copyright IBM Corporation 2019

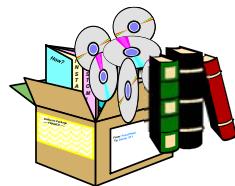
10-52

*Tivoli Enterprise Portal topology view nodes and connections*

The detailed view of a resource dependency is shown as nodes that represent the involved resources. The target resource is in focus and the dependency resources are shown as connected resources. The nodes and connections have various attributes as shown on this slide.

# Installation and customization overview

- Prerequisites
  - IBM Tivoli Monitoring Services (ITMS, 5698-A79) V6.3.0 or later
    - Tivoli Enterprise Portal Server and desktop client
    - Tivoli Enterprise Monitoring Services
- Deliverables
  - SA z/OS monitoring agent (Tivoli Enterprise Monitoring Server)
    - Tivoli Enterprise Portal data files, for example, workspaces, situations, online help
    - Traditional z/OS libraries that use SMP/E
  - Customize SA z/OS Automation Agents (NetView)
- The SA z/OS monitoring agent supports the IBM Tivoli Monitoring self-describing-agent feature that allows to automatically install all support files on the monitoring server, the portal server and the portal itself



© Copyright IBM Corporation 2019

10-53

Installation and customization overview

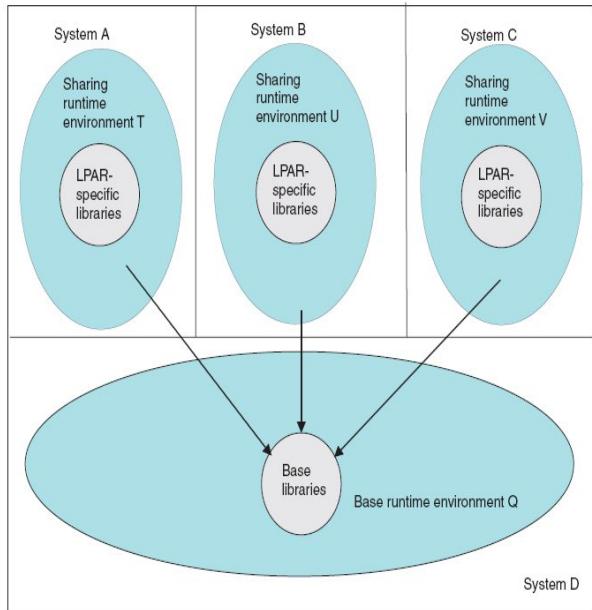
## Supported platforms

- SA z/OS monitoring agent (TEMA) runs on z/OS *only*
- One z/OS system for the entire sysplex
  - Can have more than one if you need a backup
- A runtime environment (RTE) is required for *each z/OS system where the monitoring agent runs*
  - RTE is a logical grouping of runtime libraries that are referenced by tasks which run on the z/OS system
  - RTE is defined with
    - the Installation and Configuration Assistance Tool (ICAT)
    - PARMGEN

Customization of the SA z/OS monitoring agent begins with the Installation and Configuration Assistance Tool (ICAT) to define a runtime environment (RTE).

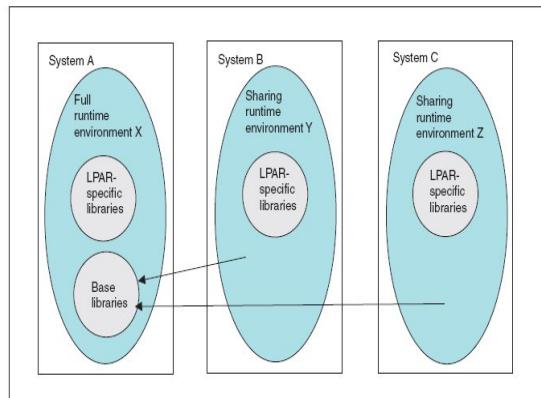
# Runtime environment example

**Shared with base**



Four systems with *base RTE* on one system,  
referenced from other three systems

**Shared with full**



Three systems with *base RTE* part of a full RTE  
on one system, referenced from other two  
systems

© Copyright IBM Corporation 2019

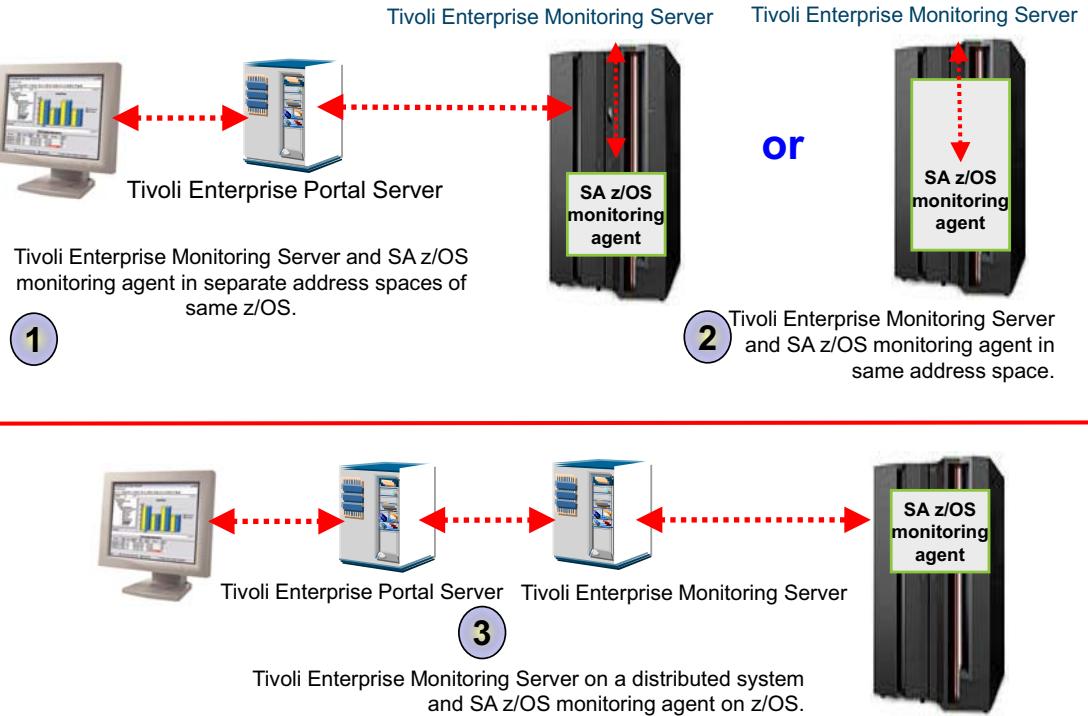
10-55

## Runtime environment examples

The types of RTEs that you can define are shown on the slide and described in the following list:

- **Full:** Self-contained; contains a full set of dedicated LPAR-specific and base runtime libraries
- **Base:** Contains a subset of libraries for the Tivoli monitoring products; must be shared by another RTE
- **Shared with base:** Contains LPAR-specific libraries with references to base libraries in a base RTE
  - Reduces disk space requirements
  - Simplifies maintenance
- **Shared with full:** Contains LPAR-specific libraries with references to base libraries in a full RTE
- **Shared with SMP/E:** Contains LPAR-specific libraries referencing libraries managed by SMP/E.

## Choose a configuration



See *IBM Tivoli System Automation for z/OS: Monitoring Agent Configuration and User's Guide*

© Copyright IBM Corporation 2019

10-56

### Choose a configuration

The Tivoli Enterprise Monitoring Server can run on z/OS or a distributed platform. If the Tivoli Enterprise Monitoring Server is on z/OS, you can run the SA z/OS monitoring agent in the same, or a different, address space as the Tivoli Enterprise Monitoring Server. This slide illustrates the available configurations:

- Option 1: z/OS Tivoli Enterprise Monitoring Server with the SA z/OS monitoring agent in a separate address space. By default, CANSDSST is the z/OS Tivoli Enterprise Monitoring Server JCL and CANSAH is the SA z/OS monitoring agent JCL.
- Option 2: z/OS Tivoli Enterprise Monitoring Server with SA z/OS monitoring agent running in the same address space, CANSDSST.
- Option 3: Distributed Tivoli Enterprise Monitoring Server, which can run on the same, or a different, workstation as the Tivoli Enterprise Portal Server.

Before you configure the SA z/OS monitoring agent, you must configure a Tivoli Enterprise Monitoring Server in the same RTE, in a different RTE, or on another platform. The SA z/OS

monitoring agent must support at least one communication protocol that is also supported by the Tivoli Enterprise Monitoring Server that it connects to.

# Installing with z/OS Tivoli Enterprise Monitoring Server: Overview

- Install the prerequisites
  - IBM Tivoli Monitoring Services components
  - DB2 Universal Database Workgroup Server Edition
  - Tivoli Enterprise Monitoring Server
  - Tivoli Enterprise Portal Server
  - Tivoli Enterprise Portal desktop client
- Define runtime environment
- Build runtime environment
- Configure hub Tivoli Enterprise Monitoring Server and the portal server using self-describing-agent feature
- Perform initial configuration of SA z/OS monitoring agent
- Load runtime libraries
- Perform final configuration of SA z/OS monitoring agent
- Configure SA z/OS and NetView for z/OS
- Install SA z/OS application support
- Verify and test

## Installing with z/OS Tivoli Enterprise Monitoring Server: Overview

This slide illustrates the steps to configure the SA z/OS monitoring agent. Most of these steps involve ICAT. All steps are documented in the *IBM System Automation for z/OS: Monitoring Agent Configuration and User's Guide*. The examples shown in this lesson define a z/OS Tivoli Enterprise Monitoring Server and the SA z/OS monitoring agent. Each one runs in a separate address space.



**Note:** If you see a KFWITM215E (Unable to process logon request) error message when connecting to Tivoli Enterprise Portal, perform the following steps:

1. Edit the **kfwenv** file for Tivoli Enterprise Portal Server.
2. Add USE\_EGG1\_FLAG=1 statement as last statement.
3. Save the file.
4. Recycle Tivoli Enterprise Portal Server.

The KFWITM215E is caused by an encryption mismatch. By default, the z/OS Tivoli Enterprise Monitoring Server disables encryption and Tivoli Enterprise Portal Server enables it.

## Install prerequisites

- Install IBM Tivoli Monitoring Services z/OS 6.3 components.  
FMIDs: HKDS620, HKLV620, and HKCI320.
- Set up the Configuration Tool or PARMGEN environment

### *Install prerequisites*

IBM Tivoli Monitoring Services z/OS has five Function Management IDs (FMIDs). You do not need all five; you only need the three shown on this slide:

- HKDS610: Tivoli Enterprise Monitoring Server on z/OS
- HKLV610: IBM Tivoli Monitoring Services on z/OS engine
- HKCI310: ICAT

## Define SA z/OS policy

- Customize and import \*ITM add-on: Automation Monitoring Agent (TEMA)

| AOFGIMP3                   |                  | Entries of selected Add-on Policies |               |   |  | Row 1 to 11 of 11 |
|----------------------------|------------------|-------------------------------------|---------------|---|--|-------------------|
| Option ==>                 |                  |                                     |               |   |  | SCROLL==> CSR     |
| 1 Modify data              | 2 Check data     | 3 Run import                        | 4 View report |   |  |                   |
| Action                     | Entry Name       | Type                                | C D           | Short Description                       |  |                   |
|                            | ING TEMA         | APG                                 |               | Y Automation Monitoring Agent (TEMA)    |  |                   |
|                            | ING TEMA X       | APG                                 |               | Y Automation TEMA and PPI               |  |                   |
|                            | OM TOM X         | APG                                 |               | Y OMEGAMON - TOM                        |  |                   |
|                            | C ITM            | APL                                 | *             | Y ITM Application Class                 |  |                   |
|                            | IBMAH            | APL                                 |               | Y Automation Monitoring Agent (TEMA)    |  |                   |
|                            | IBMCN            | APL                                 |               | Y OMEGAMON Subsystem Interface          |  |                   |
|                            | IBMDS            | APL                                 |               | Y ITM Monitoring Server (TEMS)          |  |                   |
|                            | IBMTOM           | APL                                 |               | Y OMEGAMON Enhanced 3270 User Interface |  |                   |
|                            | INGAHUPI         | APL                                 |               | Y TEMA communication task               |  |                   |
|                            | ING TEMA AUTOOPS | AOP                                 |               | Y Automation Monitoring Agent Auto Ops  |  |                   |
|                            | SESS AUTOOPS     | AOP                                 |               | Y OMEGAMON Session Operators            |  |                   |
| ***** Bottom of data ***** |                  |                                     |               |   |  |                   |

- Review APL policy for CANSAH, CANSSTS, and INGAHPPI
- Rebuild and refresh automation configuration file (ACF)

On this slide, you see the importing of the **SA ITM Agent** policies from the \*ITM add-on. After the import is complete, you modify the policy for your environment. For example, you might have to modify items such as job name or Inform List.

# Configure SA z/OS monitoring agent

- Define PPI buffer queues: INGAHINI
  - INGAHRCV: PPI receiver for requests, including heartbeat, from SA z/OS monitoring agent
    - Must match KAH\_PPI\_RECEIVER
  - KAHNVLIS: PPI receiver for events from SA z/OS monitoring agent
    - Must match KAH\_PPI\_LISTENER
- Other PPI buffer queues are created dynamically
  - KAH\$HTBT: Queue for heartbeat responses
  - KAHxx: Queue for request responses
    - 12 possible queues: KAH00 through KAH11
  - KAH\$PLEX: Queue for automation environment data
  - Data buffer size of 512
- If you use default values, customizing INGAHINI is optional

## Configure SA z/OS monitoring agent

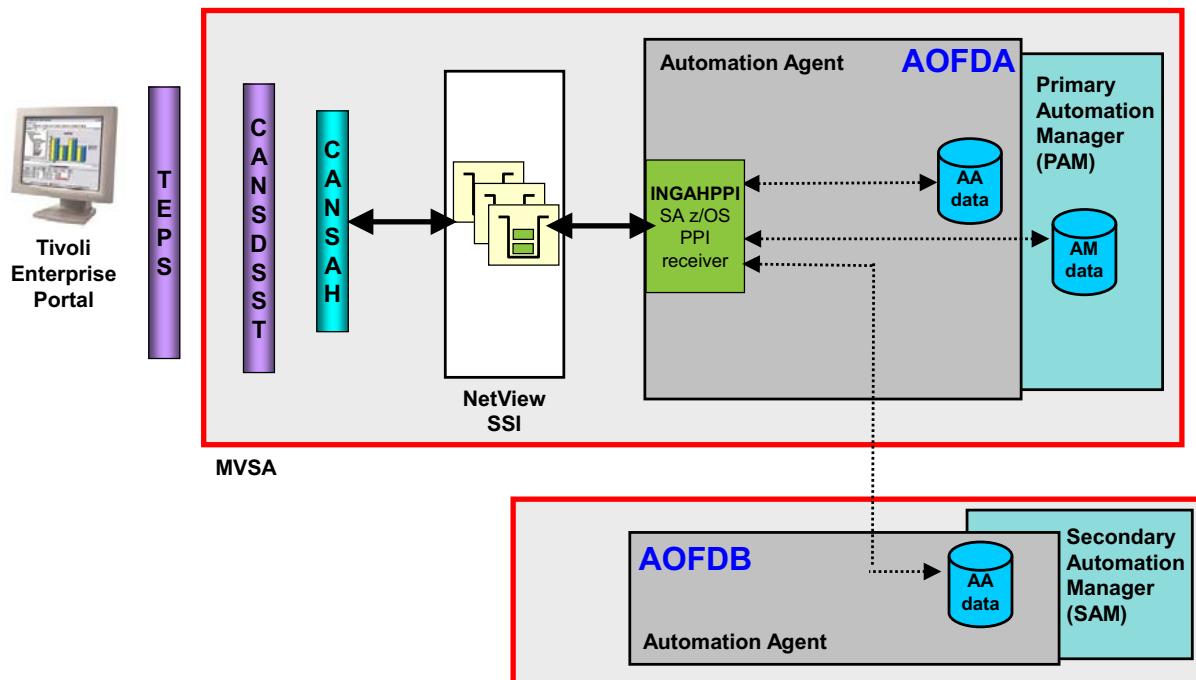
Locate INGAHINI in one of your DSIPARM data sets. INGAHINI defines the necessary buffers within the PPI to support communication between the SA z/OS monitoring agent address space (CANSAH by default) and the SA z/OS PPI receiver (INGAHPPI):

- KAH00: Automation statistics workspace requests
- KAH01: Monitor resources workspace requests
- KAH02: OMEGAMON sessions workspace requests
- KAH03: Automation environment workspace requests
- KAH04: Automation manager detail information workspace requests
- KAH05: Resource agent information workspace requests
- KAH06: Resource manager information workspace requests
- KAH07: Resource list workspace requests
- KAH08: Resource requests workspace requests
- KAH09: Resource votes workspace requests

- KAH10: Automation agent detail information workspace requests
- KAH11: User-defined status items workspace requests
- INGAHRCV: PPI receiver for requests, including heartbeat, from SA z/OS monitoring agent
- KAHNVLIS: PPI receiver for events from SA z/OS monitoring agent
- KAH\$HTBT: Queue for heartbeat responses
- KAH\$PLEX: Queue for automation environment data

If you experience problems with communication between the Tivoli Enterprise Portal Server and the SA z/OS monitoring agent, you can display the PPI buffer queues. For example, issue the DISPPI ALL command and search for the KAH and ING buffer queue name prefixes.

# SA z/OS monitoring agent with z/OS Tivoli Enterprise Monitoring Server



**CANSAH:** Default monitoring agent address space. **MVSB**

**CANSDSST:** Default Tivoli Enterprise Monitoring Server address space.

© Copyright IBM Corporation 2019

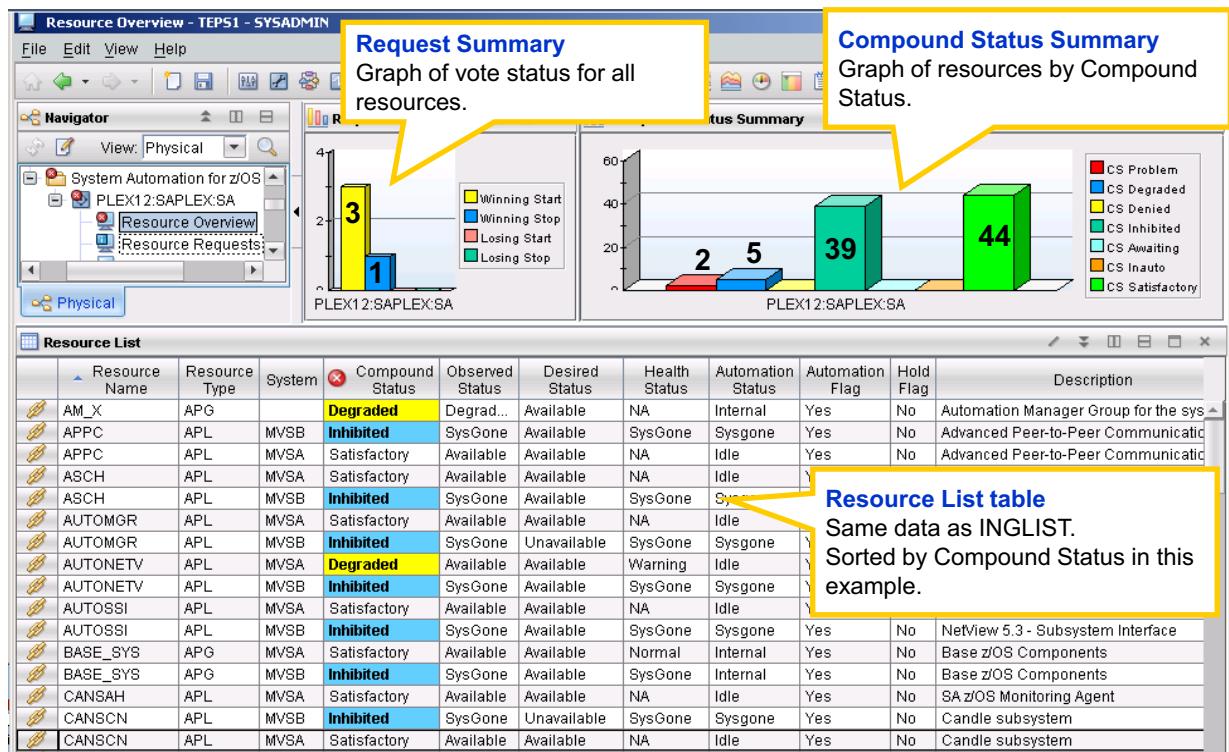
10-61

## SA z/OS monitoring agent with z/OS Tivoli Enterprise Monitoring Server

This slide provides an overview of the components that are required for the SA z/OS monitoring agent:

- SA z/OS automation agents, in this case, domains AOFDA and AOFDB
- SA z/OS automation manager; both primary and secondary are shown
- SA z/OS PPI receiver (INGAHPPi)
- Buffer queues in the NetView PPI
- SA z/OS monitoring agent, CANSAH
- Tivoli Enterprise Monitoring Server, in this case, a z/OS Tivoli Enterprise Monitoring Server, CANSDSST
- Tivoli Enterprise Portal Server running on a distributed workstation
- Tivoli Enterprise Portal

# Resource Overview workspace



© Copyright IBM Corporation 2019

10-62

## Resource Overview workspace

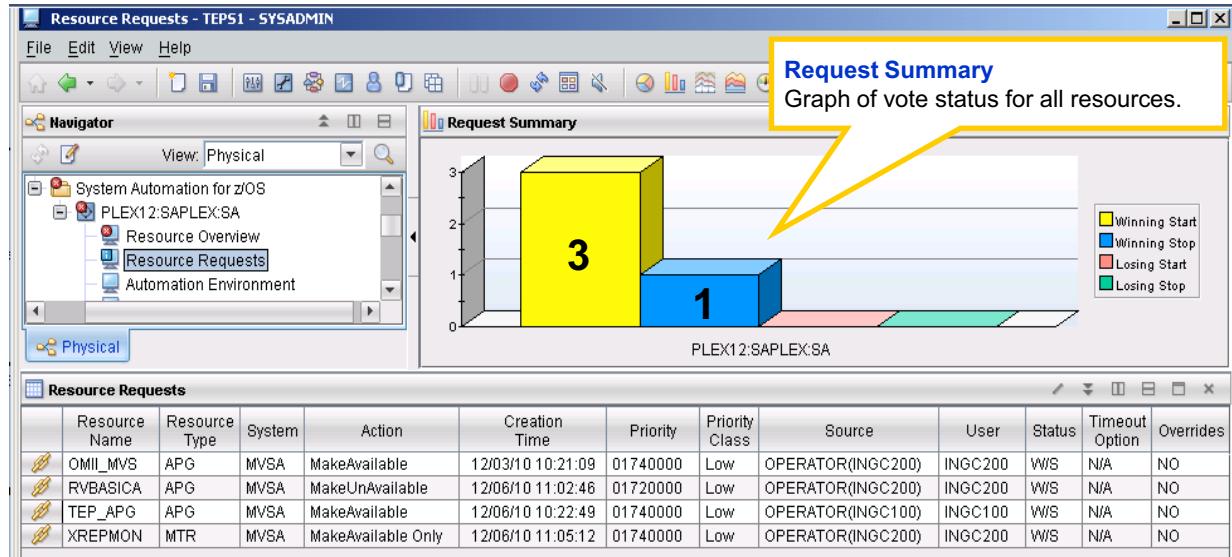
On the Resource Overview workspace, you can see the data that is equal to the INGLIST command. The Request Summary bar chart includes requests by types start, stop, winning, and losing. Links to resource requests are also on the chart.

The Compound Status Summary bar chart includes the distribution of resources based on compound status. The Resource List table has the following details about resources:

- All states (compound, observed, desired, automation, and health)
- Schedules (Service Periods)
- Triggers

Links are provided to start the resource details for a selected resource and Resource Requests workspace.

## Resource Requests workspace



- Start request against OMII\_MVS APG.
- Stop request against RVBASICA APG.
- Start request against TEP\_APG APG.
- Start request against XREPMON MTR.

© Copyright IBM Corporation 2019

10-63

### Resource Requests workspace

The Resource Requests workspace includes data that is equal to the INGVOTE command. The Request Summary bar chart has requests by type: start, stop, winning, and losing. The Resource Requests table includes all current requests:

- Resource name
- Resource type
- System
- Request type (for example, MakeUnavailable)
- Origin of request
- Numeric priority value
- Status (for example, winning/satisfied)

Links are available for locating resource details for a selected resource workspace.

## Other Tivoli Enterprise Portal customization

- Define Take Action commands, for example:
  - INGREQ START | STOP
  - INGREQ CANCEL to cancel a request (vote)
  - Event-based automation commands (INGSIT)
- Define event-based automation with Situation Editor

### Other Tivoli Enterprise Portal customization

You can define commands for use by operators and event-based automation, called *Take Action* commands. Forwarding of Take Action commands to NetView is enabled in the Installation and Configuration Tool (ICAT). Enabling Take Action commands defines NetView to the z/OS Tivoli Enterprise Monitoring Server and SA z/OS monitoring agent:

- Configure hub Tivoli Enterprise Monitoring Server: Specify Configuration Values panel (KDS61PP1).  
Tivoli Enterprise Monitoring Server PPI sender defaults to Tivoli Enterprise Monitoring Server job name (CANSDDSST).
- Configure SA z/OS monitoring agent: Specify Advanced Agent Configuration panel (KAG61P5).



**Note:** You must add NetView CNMLINK to RKANMODL concatenation in CANSDDSST and CANSAH.

Command authorization is based on NetView operator ID, such as SYSADMIN. You can define command authorization (command name, parameters, and parameter values) with these items:

- Command authorization table (CAT): CNMSCAT2
- Security access facility: CNMSAF2

## Event-based automation: Setup tasks

- Tivoli Enterprise Portal situation editor
  - Define situation, for example, MyAuxShortage\_Warn: Formula, action, expert advice
  - Action is INGSIT command
- SA z/OS policy
  - Define monitor resource (MTR): AUXSHORT
    - MONITOR INFO policy item:
      - Monitored Object name = ITM.MYAUXSHORTAGE\_WARN
    - MESSAGES/USER DATA policy item
      - Define Automation Table override for ING150I to issue command
      - Optionally, define CODEs for ING150I to set Health Status
    - HEALTHSTATE policy item: Define command to be issued when Health Status changes
  - Optionally, define SOAP SERVER in NTW policy

### Event-based automation: Setup tasks

This slide provides an overview of the tasks that are required to implement event-based automation. Use the Tivoli Enterprise Portal situation editor to define the situation; for example, MyAuxShortage\_Warn. With the formula, you can test one or more attributes or a combination of situations. The action is the INGSIT command for event-based automation. INGSIT generates an ING150I message that contains the situation name prepended with **ITM**. For example, ITM.MYAUXSHORTAGE\_WARN.

SA z/OS searches MTR policies for any messages that contain a *monitored object name* of MYAUXSHORTAGE\_WARN. The monitored object name provides a link between the monitored resource and the situation name. This field must be uppercase and must contain the situation name prepended with **ITM**, to match the text of the ING150I message generated for a situation event. The MESSAGES/USER DATA policy item defines the Health Status for the monitor resource plus the actions to take when an ING150I message is issued. Optionally, you can define the SOAP SERVER policy item if you use INGOMX to issue requests, such as retrieving attributes and values or acknowledging an event.

# MONITOR INFO policy

```
AOFGMTR          Monitor Resource Information
Command ==> __

Entry Type : Monitor Resource      PolicyDB Name   : NEW_PDB
Entry Name  : AUXSHORT            Enterprise Name : SAPLEX

Monitored Object. . . . . ITM.AUXSHORT_WARN
Monitored Jobname . . . . .
Activate Command. . . . .

Deactivate Command . . . .

Monitor Command. . . . .

Monitoring Interval . . . . . (hh:mm)
Captured Messages Limit . . . . . (0 to 999)
Desired Available . . . . . (ALWAYS ONDEMAND ASIS)
Monitor for IPL complete. . . . . (YES NO)
```

**Monitored Object** and  
**Monitored Jobname** fields bind  
monitor resources to real  
objects and optionally their jobs

- Event-based monitoring: Situation event generates an ING150I message that contains the monitored object name and, optionally, the job name
- SA z/OS uses the monitored object name and job name to locate the Monitor Resource to set the Health Status and issue commands

## MONITOR INFO policy

The Monitored Object name must begin with **ITM.**, followed by the situation name. Otherwise, SA z/OS cannot correlate the situation event with the ING150I message that is generated by the situation.

# SOAP server requests

- INGOMX command can send XML requests to SOAP server
  - Requests can be stored in default PIPE safe or member of a data set
  - SOAP server does not have to be defined to SA z/OS
- New SOAP SERVER policy item of NTW policy defines server
  - IP address or host name
  - IP port
  - USS path
- Supported SOAP requests
  - CT\_ACKNOWLEDGE
  - CT\_ACTIVATE
  - CT\_ALERT
  - CT\_DEACTIVATE
  - CT\_EXECUTE
  - CT\_GET
  - CT\_REDIRECT
  - CT\_RESET
  - CT\_RESURFACE
  - CT\_WTO
- Requests and parameters are documented in the *IBM Tivoli Monitoring 6.3 Administrator Guide*

You might have to use NETVASIC for mixed case text

## SOAP server requests

Requests can be sent to a SOAP server by using the **INGOMX** command. Although it is not required, you can define the SOAP server to SA z/OS with the SOAP SERVER policy item of the NTW policy. For example, in your event-based automation you can acknowledge the situation event by sending a CT\_ACKNOWLEDGE request to the Tivoli Enterprise Monitoring Server.



**Note:** SOAP server requests are case-sensitive.

## SOAP SERVER policy item

| AOFGSOAA                |                | SOAP-Server Attributes                              |
|-------------------------|----------------|---|
| Command                 | ==>            |   |
| Entry Type              | : Network      | PolicyDB Name : ZPDT_MASTER                         |
| Entry Name              | : TM1_SOAP_SRV | Enterprise Name : ADCD                              |
| Server Name             | TEM SZOS       |   |
| Host name . . . . .     | tm1-win-host   |   |
| IP stack . . . . .      |                | Symbolic host name or IP-address of the SOAP-Server |
| Port number . . . . .   | 1920           | Port number of the SOAP-Server (1-65535)            |
| Protocol . . . . .      | HTTP           | Protocol used for SOAP requests (HTTP or HTTPS)     |
| User ID . . . . .       |                | User ID to log on to SOAP-Server                    |
| Password . . . . .      |                | Password of the logon user or SAFPW                 |
| Absolute path . . . . . | ///cms/soap    | Absolute path of the SOAP-Server on that host       |

### Network policy (NTW), SOAP SERVER policy item:

- Define default parameters for SOAP server
- INGOMX SOAPREQ SERVER=TEM SZOS retrieves host name, port number, and absolute path from policy

On this slide, you see an example of the SOAP SERVER policy item. It is defined as part of the NTW (network) policy. In this case, the z/OS Tivoli Enterprise Monitoring Server is defined as a SOAP server, named TEM SZOS. When an INGOMX SOAPREQ command is issued for the TEM SZOS server, these fields are retrieved from the policy and added to the INGOMX command.

## SOAP server request example

Use INGOMX to acknowledge the MyAuxShortage\_Warn situation with text of *automation in progress*

```
INGOMX SOAPREQ SERVER=TEMSZOS TYPE=ITM  
DATA=USER.SOAPREQ(ACK)
```

Where USER.SOAPREQ(ACK) contains these statements

```
<CT_Acknowledge>  
  <target>TEMSzOS:CMS</target>  
  <name>MyAuxShortage_Warn</name>  
  <source>PLEX1:SAPLEX:SA</source>  
  <data>automation in progress.</data>  
</CT_Acknowledge>
```

Use NetView PIPE if you need to process response to INGOMX

© Copyright IBM Corporation 2019

10-69

### SOAP server request example

On this slide, you see an example of sending a CT\_ACKNOWLEDGE request to a z/OS Tivoli Enterprise Monitoring Server to identify that event-based automation is in progress. This example uses a data set member, USER.SOAPREQ(ACK), as the source of the XML statements. The XML parameters for the CT\_ACKNOWLEDGE request are documented in the *IBM Tivoli Monitoring 6.2 Administrator Guide*.



**Note:** Consider using NetView pipelines if you want to process the response. You can also turn on tracing for the INGOMX command to see more details related to the command and response (AOCTRACE INGROMX0,ON). Remember to turn off tracing.

# Student exercises

## Unit 10

### 1. OMEGAMON integration and exception-based automation

- Import the OMEGAMON policy
- Define OMEGAMON sessions and operators
- Define a monitor resource for OMEGAMON exceptions
- Verify your XREP monitor resource definitions



© Copyright IBM Corporation 2019

10-70

### Student exercises

Perform the exercises for this unit.

## Summary

---

Now that you have completed this unit, you should be able to perform the following tasks:

- Customize the PDB for OMEGAMON Applications
  - Import the \*ITM add-on policy
  - Define and manage the sessions between OMEGAMON and SA z/OS
- Issue OMEGAMON commands from NetView
- Describe exception-based monitoring
- Define OMEGAMON Monitor resources
- Describe event-based monitoring
- Describe how to set up and use the Tivoli Enterprise Portal workspaces

### *Summary*

Now that you have completed this unit, you should be able to perform the following tasks:

- Customize the PDB for OMEGAMON Applications
- Import the \*ITM add-on policy
- Define and manage the sessions between OMEGAMON and SA z/OS
- Issue OMEGAMON commands from NetView
- Describe exception-based monitoring
- Define OMEGAMON Monitor resources
- Describe event-based monitoring
- Describe how to set up and use the Tivoli Enterprise Portal workspaces



# 11 Migration and coexistence

---



## Unit 11 Migration and coexistence



© Copyright IBM Corporation 2019  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

---

**What this unit is about**

This unit describes the migration from and coexistence with previous releases of SA z/OS.

**How you check your progress**

You can check your progress in the review questions.

**References**

*SC34-2716 System Automation for z/OS Version 4*

*Release 1 Installation and Planning*

<http://www.ibm.com/servers/eserver/zseries/zos/bkserv>

## Objectives

---

After completing this unit, you should be able to describe migration and coexistence issues with SA z/OS 4.1

# Lesson 1. Migration

## Lesson 1: Migration

- These slides cover the most common migration concerns when you migrate from SA z/OS 3.4 or SA z/OS 3.5 to SA z/OS 4.1
- For more details, including information related to all migration concerns, read Appendix B of the *IBM Tivoli System Automation for z/OS 4.1 Planning and Installation* guide
  - Migration Notes and Advice when Migrating to SA z/OS 4.1
  - Migration Notes and Advice when Migrating from SA z/OS 3.4
  - Coexistence of SA z/OS 4.1 with Previous Releases



### What this lesson is about

This lesson review migration from a previous release of System Automation.

### Objectives

After completing this lesson, you should be able to describe the migration issues associated with migrating from a previous release of System Automation.

### References

*SC34-2716 System Automation for z/OS Version 4 Release 1 Installation and Planning*

## Apply maintenance and convert PDBs

Before migrating to SA z/OS 4.1, perform the following steps:

- Install continuous enhancements (post-GA service-level):
  - OA55386: Service Management Unite V1.1.4, End-to-end automation...
  - OA52610: Planned suspend capability, SMU enhancements...
  - OA52638, OA53587: IBM z14 Toleration and Exploitation Support
- Create a backup copy of your SA z/OS 3.4 and 3.5 PDBs
- Install compatibility APAR OA51668 on SA z/OS 3.4 and 3.5 to use a SA z/OS 4.1 configuration on a system that runs SA z/OS 3.4 or SA z/OS 3.5 in a mixed environment
- Make a copy of your V3.x policy database and edit it with the SA z/OS 4.1 Customization Dialog to convert the database to a V4.1 policy database

### Apply maintenance and convert PDBs

1. Install continuous enhancements (post-GA service-level):
  - OA55386: Service Management Unite V1.1.4, End-to-end automation...
  - OA52610: Planned suspend capability, SMU enhancements...
  - OA52638, OA53587: IBM z14 Toleration and Exploitation Support
  - OA54030 – INGRDS Command Enhancements
  - OA53366 – Small enhancement
2. Install the compatibility APAR OA51668 (SA z/OS 3.4 and SA z/OS 3.5) before migrating to SA z/OS 4.1. Open the customization dialog before converting to a SA z/OS 4.1 policy database in step 2. This APAR also enables you to use a SA z/OS 4.1-built configuration file on a system running SA z/OS 3.4 or SA z/OS 3.5 in a mixed environment.
3. Make a copy of your V3.n policy database and edit it with the SA z/OS 4.1 customization dialog. This converts it to a V4.1 policy database. For more information, see "Conversion Function" in IBM System Automation for z/OS Defining Automation Policy.r

## Review release-related changes

- Review support for suspending and resuming resources
  - Changes in delivered Policy Entry +SA\_PREDEFINED\_MSGS
  - NMC component removal
  - Introduction of automated functions INITOPR1 and INITOPR2
  - With NetView 6.1: drop your modified DSIRXPRM module
  - Start NetView task DSIRQJOB (part of the infrastructure for the NetView SUBMIT and ALLOCATE command)
  - Make sure that you use the unchanged INGMSGSA AT
- Coming from V3.4
- REXX error messages are now run through the AT
  - File Update changes for specifications for Message Automation
  - 'subtype' changed to 'subcategory' in INGLIST, INGFILT, INGIMS commands
  - SDF: convert the AAO variable AOF\_AAO\_SDFROOT\_LISTn into AOF\_AAO\_SDFROOT.n.
  - Output format of Policy report changed

© Copyright IBM Corporation 2019

11-5

### Review release-related changes

Review release-related changes and decide which new features and functions might affect how you want to automate your enterprise:

- The ProcOps Automation Table ISQMSG01 has been retired. Its content is merged into INGMSGSA
- The NetView task DSIRQJOB must be started also. Consult INGSTGEN member and NetView's Getting started book GI11-9443. SA z/OS provides a sample JCL in SINGSAMP. NetView provides a sample JCL in CNMSAMP.

### Suspending and resuming resources

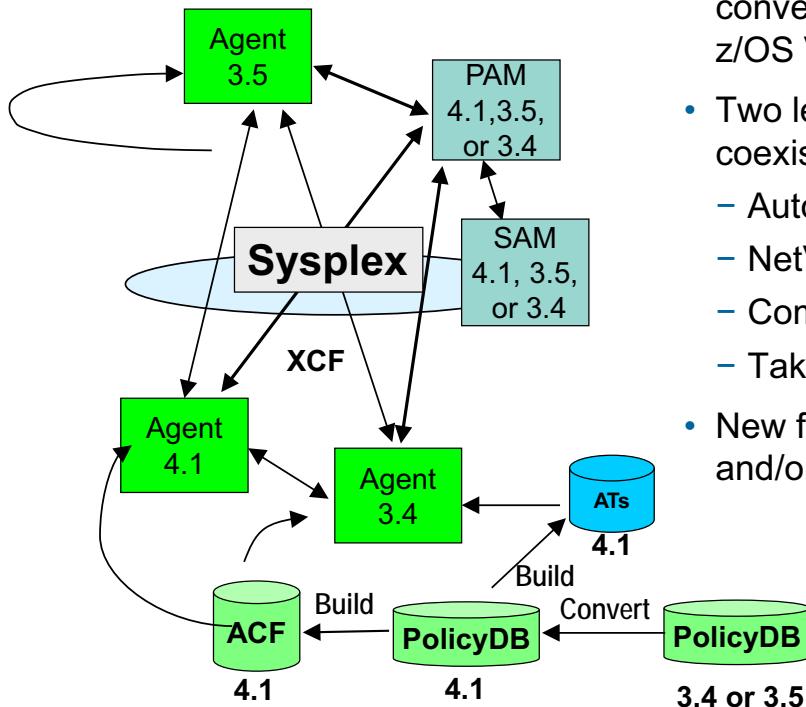
The support for suspending and resuming resources affects many commands and displays. Consider implementing planned suspend capability.

## Coming from V3.4

- REXX error messages are now run through the AT
- File Update changes for specifications for Message Automation
- 'subtype' changed to 'subcategory' in INGLIST, INGFILT, INGIMS commands
- For the definition of the systems participating in SDF: convert the AAO variable AOF\_AAO\_SDFROOT\_LISTn into AOF\_AAO\_SDFROOT.n. in the NetView style sheet
- Output format of Policy report changed

# Lesson 2. Coexistence

## Lesson 2: Coexistence



- Older policy is automatically converted when opened with SA z/OS V4.1 Customization Dialog
- Two levels down supported to coexist in runtime:
  - Automation configuration file
  - NetView automation table
  - Communication via XCF
  - Takeover File
- New functions require new AM and/or agent

© Copyright IBM Corporation 2019

11-6

### What this lesson is about

This lesson reviews coexistence considerations for multiple versions of System Automation in the same environment.

### Objectives

After completing this lesson, you should be able to describe coexistence considerations for multiple versions of System Automation in the same environment.

### References

SC34-2716 System Automation for z/OS Version 4  
Release 1 Installation and Planning

You might not want to convert all your systems from previous releases to SA z/OS 4.1 at the same time because you might be running different releases simultaneously. SA z/OS 4.1 systems can coexist with SA z/OS 3.4 and SA z/OS 3.5 systems in the same sysplex. This slide illustrates how a

sysplex with three automated systems, and a separate primary automation manager and its secondary automation manager can coexist.

Any policy database that is created by an earlier version of the customization dialog; that is, earlier than SA z/OS 4.1, is automatically converted into the SA z/OS 4.1 format. This conversion happens when the policy database is opened the first time by using the SA z/OS 4.1 customization dialog. The automation configuration files that are built by the SA z/OS 4.1 customization dialog can be used by any automation agent that runs either SA z/OS 4.1, SA z/OS 3.4, or SA z/OS 3.5.

The NetView automation table (AT) that is created by the SA z/OS 4.1 customization dialog can be used by automation agents that run either SA z/OS 3.4 or SA z/OS 3.5, but the automation table INGMSGSA is required for compatibility with SA z/OS 3.4. A new NetView automation table (AT), INGMSGSA, is delivered by SA z/OS. It contains about 50 message traps, basically for NetView and SA z/OS SysOps base messages.

In a sysplex, automation agents that run SA z/OS 4.1, SA z/OS 3.4, or SA z/OS 3.5 can communicate with an SA z/OS 4.1 automation manager in the same XCF group. The communication uses XCF, and the automation agents communicate with each other by using XCF.

## Restrictions concerning suspend and resume functionality (1 of 2)

If you try to manage a suspended resource running on a SA 4.1 system with a down-level system, you will observe the following results:

- Resources are not shown as SUSPENDED
- Suspend votes and requests are shown but are marked with a red colored text \*\*Unsupported SUSPEND request\*\*
- Suspend flag is shown but can not be changed

### *Restrictions concerning suspend and resume functionality (1 of 2)*

The suspend functionality is available only with SA z/OS 4.1 and higher versions. If you try to manage a suspended resource running on a SA 4.1 system with a down-level system, you will observe the following results:

- INGLIST and DISPMTR do not show the resource as SUSPENDED.
- INGINFO and INGVOTE will show the suspend votes and requests, but they are marked with a red colored text \*\*Unsupported SUSPEND request\*\*.
- DISPFLGS with TARGET to the SA 4.1 system will show the correct data but will not allow to change the suspend flag.
- INGAUTO will report AOF144I message due to the value of S in the automation flag.
- DISPSTAT shows correct data.
- INGAUTO with TARGET to the SA 4.1 behaves the same way as on the SA 4.1 system and does not allow to change any automation flag.

## Restrictions concerning suspend and resume functionality (2 of 2)

If a resource running on a down-level system has been suspended by a SA 4.1 system, either by accident or because the resource is in the dependency tree of a suspended resource running on a SA 4.1 system, you will observe the following results:

- INGLIST shows nothing on back-level system. On SA 4.1 system, INGLIST shows the resource as SUSPENDED, but the automation status remains ORDERED
- The automation flag of the resource running on the back-level system will not be set
- The following error messages appear: INGY1015I, ING008I, INGY1000I
- Resource is suspended in the automation manager, but not in the down-level automation agent: automation status remains ORDERED.  
This has to be fixed manually via INGSET or INGLIST-Update

### *Restrictions concerning suspend and resume functionality (2 of 2)*

If a resource running on a down-level system has been suspended by a SA 4.1 system, either by accident or because the resource is in the dependency tree of a suspended resource running on a SA 4.1 system, you will observe the following results:

- INGLIST shows nothing on back-level system. On SA 4.1 system, INGLIST shows the resource as SUSPENDED, but the automation status remains ORDERED.
- The automation flag of the resource running on the back-level system will not be set.
- The following error messages appear in NetView log:

```
INGY1015I Conversion error detected. See INGDUMP data set for details
ING008I INGPYAMC SERVICE FAILED, RC=8 - REASON=UNKNOWN -> 8 MOD=INGRXCAM -
GET_ORDERS
INGY1000I INGPYDEC FAILED IN MODULE INGPYBLD, RC=x'0008', REASON=x'0003'
000000002000000804040404040404040404
```

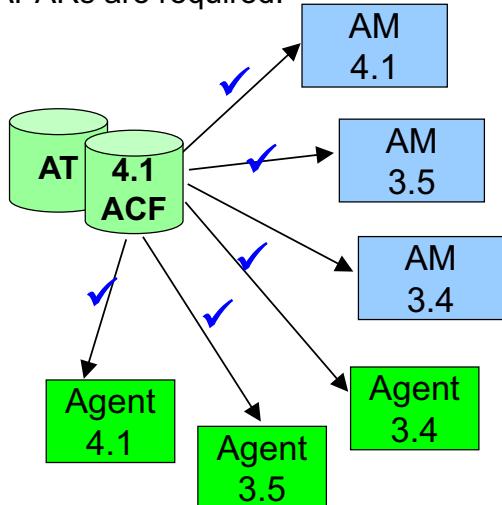
- Resource is suspended in the automation manager, but not in the down-level automation agent. Therefore, resuming the suspended resource is accepted and honored by the automation manager, nevertheless the automation status remains ORDERED.

This has to be fixed manually via INGSET or INGLIST-Update.

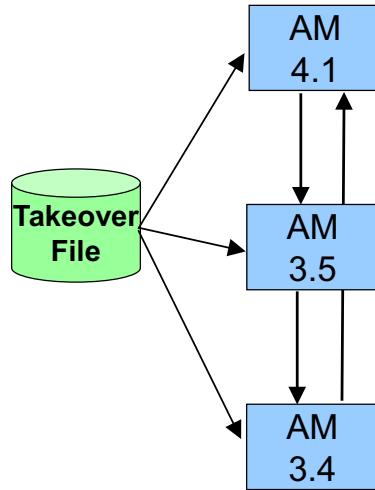
## ACF and takeover file coexistence

ACF built by SA z/OS 4.1 customization dialog can be used by SA z/OS 3.5 and 3.4 agents, however 3.4 is out of support

APARs are required.



Format of Takeover File compatible with all Automation Manager versions



© Copyright IBM Corporation 2019

11-9

ACF and takeover file coexistence

### ACF coexistence

ACFs that you build by using the SA z/OS 4.1 customization dialog is *downwardly compatible* to SA z/OS V3.4 and V3.5 automation managers and automation agents.

### Takeover File coexistence

The Takeover File that is processed by an SA z/OS V4.1 automation manager can be read by a V3.4 or V3.5 automation manager when a PAM switch occurs. The SA z/OS V4.1 automation manager can read a Takeover File that is written by a SA z/OS V3.4 or V3.5 automation manager.

Remember that the Takeover File contains the configuration data, resource states, all requests, and the logic deck associated configuration data. The logic deck that is provided by SA z/OS 4.1 is accepted by an SA z/OS V3.4 or V3.5 automation manager

# Review questions

1. What System Automation for z/OS APAR should be applied to SA for z/OS versions 3.5 and 3.4 when migrating to System Automation for z/OS version 4.1?
2. What previous versions of System Automation for z/OS can coexist with version 4.1?
3. How are PDBs defined using previous versions of SA for z/OS converted to SA for z/OS version 4.1 PDBs?

## Review answers

1. What System Automation for z/OS APAR should be applied to SA for z/OS versions 3.5 and 3.4 when migrating to System Automation for z/OS version 4.1?

*APAR OA51668*

2. What previous versions of System Automation for z/OS can coexist with version 4.1?

*SA for z/OS versions 3.5 and 3.4*

3. How are PDBs defined using previous versions of SA for z/OS converted to SA for z/OS version 4.1 PDBs?

*They are automatically converted when opened in the SA for z/OS version 4.1 ISPF customization dialog.*

## Summary

---

Now that you have completed this unit, you should be able to describe migration and coexistence issues with SA z/OS 4.1





SM937 1.0



[ibm.com/training](http://ibm.com/training)

Authorized  
**IBM | Training**