Tivoli. software

IBM

IBM Tivoli System Automation for z/OS Enterprise Automation

Enterprise class automation on IBM System z

Migration from other automation platforms

Support for end-to-end automation

Budi Darmawan
Adam Palmese
Andrew McIntyre
Francisco Duarte
Mark Talstein

Redbooks

ibm.com/redbooks

International Technical Support Organization

IBM Tivoli System Automation for z/OS Enterprise Automation

January 2007

SG24-7308-00

Note: Before using this information and the product it supports, read the information in "Notices" on page xi.

First Edition (January 2007)

This edition applies to Version 3, Release 1 of IBM Tivoli Systems Automation for z/OS (product number 5698-A14).

Contents

Figures

Tables

Notices

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®
CICS®
CICSPlex®
Database 2™
DB2 Universal Database™
DB2®
ESCON®
FICON®
Geographically Dispersed Parallel Sysplex™
GDPS®
IBM®
ibm.com®
IMS™
MVS™
NetView®
OMEGAMON II®
OMEGAMON®
OS/2®
OS/390®
Parallel Sysplex®
Passport Advantage®
Redbooks™
Redbooks (logo) ™
RACF®
RMF™
S/360™
System x™
System z™
Tivoli Enterprise™
Tivoli Enterprise Console®
Tivoli®
VTAM®
WebSphere®
z/OS®
z9™

The following terms are trademarks of other companies:

SAP, and SAP logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

Java, JRE, PDB, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

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

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

Other company, product, or service names may be trademarks or service marks of others.

Preface

This IBM® Redbook provides an overview of IBM Tivoli® System Automation for z/OS® concepts and new features. We also discuss some considerations on migrating non-IBM products to IBM Tivoli System Automation for z/OS.

The discussion is primarily aimed at technical professionals that are looking to understand the IBM Tivoli System Automation for z/OS new features and find migration options.

This IBM Redbook explains some important features of IBM Tivoli System Automation for z/OS, including using the Processor Operations feature, integration with IBM Tivoli OMEGAMON® Classic, and integration with End-to-end automation feature of the IBM System Automation for Multiplatform.

We will also discuss the available services from the Software Migration Project Office that provide migration services for mainframe based products, including IBM Tivoli System Automation for z/OS, in this new implementation of IBM Tivoli System Automation for z/OS. We also provide some overview of the migration procedures from non-IBM products.

The team that wrote this redbook

This IBM Redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Austin Center.

Budi Darmawan is a Consulting IT Specialist at the International Technical Support Organization, Austin Center. He writes extensively and teaches IBM classes worldwide on all areas of system management. Before joining the ITSO seven years ago, Budi worked as lead implementer and solution architect for system management solution in Integrated Technology Services of IBM Indonesia. His current interests include end-to-end system management, business availability management, and application management.

Adam Palmese is an IT Specialist in the Software Migration Project Office (SMPO) of the System z™ Software Technical Sales group. He is based out of Raleigh, North Carolina. Adam has worked for IBM for 10 years in various roles in large MVS™ installations. He has extensive experience in Parallel Sysplex® environments supporting SAP®. Adam holds a degree from Pace University and is currently a member of the US Air Force Reserve.

Andrew McIntyre is a Consulting IT Specialist in the Software Migration Project Office (SMPO) of the System z Software Technical Sales group. He is based in Atlanta, Georgia. He has eight years of experience in working with NetView® and SA in his eight year career at IBM. He has over 30 years of experience in mainframe software and wrote his first S/360™ assembler program in 1972.

Francisco Duarte is a IT Specialist Automation Systems in Brazil. He has over 20 years of experience in System and Network Management in large MVS installations and over eight years of experience in NetView and System Automation implementation and operations. During the last three years, he has been developing, designing, and supporting the System Management area and preparing some clients to implement GDPS® and assisting them in system automation implementation. He holds a degree in Mathematics from Universidade Federal de Pernambuco.

Mark Talstein from IBM United States.

Thanks to the following people for their contributions to this project:

Bob Haimowitz and Wade Wallace
International Technical Support Organization

Joachim Schmalzried, Wiltrud Rebmann
IBM Germany

Butch Rambish
IBM Software Group, Tivoli System

Sue Hamner
IBM Software Migration Project Office

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and client satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

`ibm.com/redbooks/residencies.html`

Comments welcome

Your comments are important to us!

We want our Redbooks™ to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at:

  `ibm.com/redbooks`

- Send your comments in an e-mail to:

  `redbooks@us.ibm.com`

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

1

Overview of IBM Tivoli System Automation for z/OS V3.1

In this chapter, we describe the new features of IBM Tivoli System Automation for z/OS V3.1. The discussion contains the following:

1.1 IBM Tivoli System Automation for z/OS overview

IBM Tivoli System Automation for z/OS is an IBM Tivoli NetView for z/OS base software product that provides a single point of control for a various range of systems management functionality. IBM Tivoli System Automation for z/OS plays a key role in supplying high end-to-end automation solutions.

IBM delivers integrated cross-platform management functions. IBM Tivoli System Automation for z/OS functions include monitoring, control, and automation of a large range of system elements spanning both the hardware and software resources of your enterprise.

IBM Tivoli System Automation for z/OS is a system management program with a single point of control. You see a single system image for a full range of essential systems management functions. IBM Tivoli System Automation for z/OS monitors, controls, and automates the following:

- Monitors the following resources to respond before they affect users:
  - Hardware components
  - Software products and applications
  - Automated processes
  - Messages and alerts
- Controls and takes action on resources based on conditions:
  - Start and stop your entire enterprise system and initiate hardware and software startup and shutdown sequences
  - Manage both remote and local operations and support any IBM System z and 390-CMOS processors within a Parallel Sysplex
  - Manage several operating systems: z/OS, OS/390®, MVS, VM, and VSE
  - Control a coupling facility as a target system with coupling links in a Parallel Sysplex environment
  - React to errors and unscheduled events
- Automates many repetitive and complex tasks:
  - Start/shutdown software resources
  - Control channels and channel paths
  - Control availability of I/O devices
  - Control switching of I/O device ports
  - Detect and respond to system messages
  - Perform initial program load (IPL)

– Perform system power-on reset (POR)

– Build an automation policy for your enterprise

– Extend the built-in automation routines by writing your own automation policies

IBM Tivoli System Automation for z/OS is an IBM Tivoli NetView for z/OS-based application. Its functions include monitoring, controlling, and automating a large range of system elements, spanning both hardware and software resources of an enterprise (see Figure 1-1).

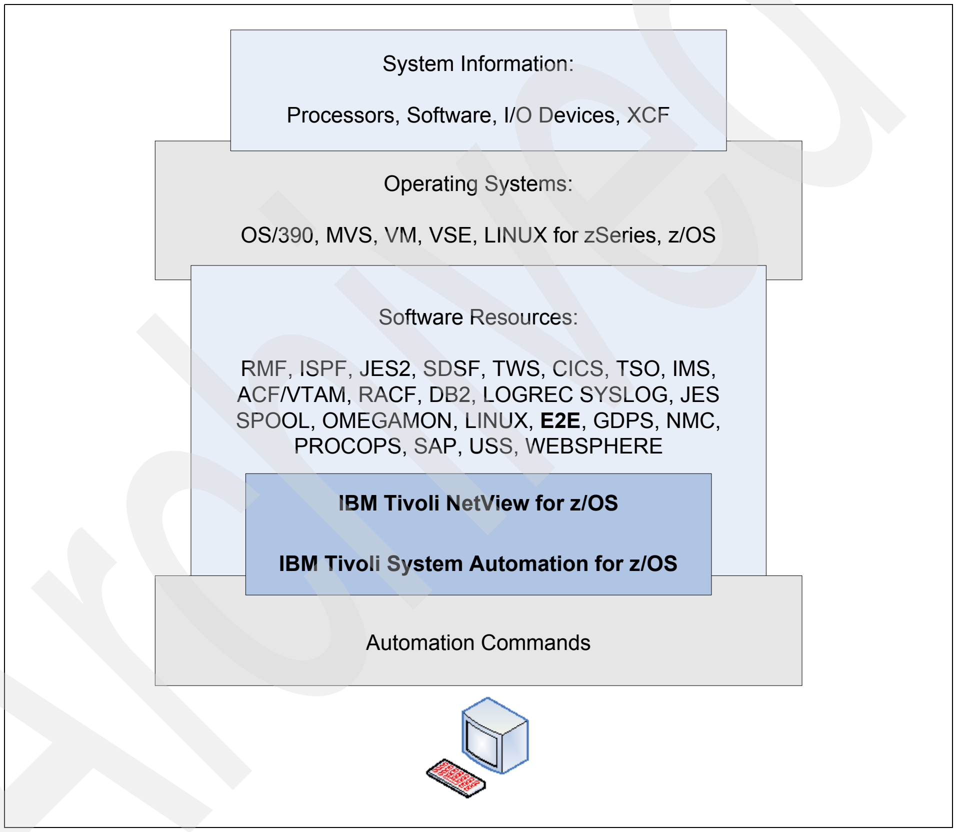

System Information:
Processors, Software, I/O Devices, XCF
Operating Systems:
OS/390, MVS, VM, VSE, LINUX for zSeries, z/OS
Software Resources:
RMF, ISPF, JES2, SDSF, TWS, CICS, TSO, IMS, ACF/VTAM, RACF, DB2, LOGREC SYSLOG, JES SPOOL, OMEGAMON, LINUX, E2E, GDPS, NMC, PROCOPS, SAP, USS, WEBSPHERE
IBM Tivoli NetView for z/OS
IBM Tivoli System Automation for z/OS
Automation Commands

Figure 1-1 Monitor, control, and automation functions

IBM Tivoli System Automation for z/OS performs the following major operations:

- System Operations
- Processor Operations
- I/O Operations

System Operations

System Operations (SysOps) monitors and controls system operations applications and subsystems, such as IBM Tivoli NetView for z/OS, SDSF, JES, RMF™, TSO, RODM, ACF/VTAM®, DB2®, CICS®, IMS™, OMEGAMON, and Tivoli Workload Scheduler. With system operations, you can automate Parallel Sysplex applications. IBM Tivoli System Automation for z/OS can automate applications distributed over a sysplex by virtually removing system boundaries for automation through its automation manager/automation agent design. IBM Tivoli System Automation for z/OS reduces the complexity of managing a Parallel Sysplex through its goal driven automation and its concepts, such as grouping, and powerful dependency support, which enables you to model your configuration. Single systems are also fully supported; the automation scope is then just one system. IBM Tivoli System Automation for z/OS uses Enterprise monitoring to update the NetView Management Console (NMC) resource status information, which is stored in the Resource Object Data Manager (RODM). IBM Tivoli System Automation for z/OS provides controlled startup, controlled shutdown, and automated recovery of various software resources on your systems. System Operations (SysOps) provides the ability to monitor and control all subsystems in a sysplex from any system in the sysplex. In other words, if your Focal Point (FP) is down for whatever reason, you can view the status from any other system within a SysPlex.

Processor Operations

Processor Operations monitor and control processor hardware operations. It provides a connection from a focal point processor to a target processor. With IBM Tivoli NetView for z/OS on the focal point system, processor operations automate operator and system consoles for monitoring and recovering target processors.

The Processor Operations facility (ProcOps), formerly known as Target System Control Facility (TSCF), provides connections from a Focal Point processor to target processors. In addition, ProcOps allow you to power multiple target processors on, off, and reset systems, perform IPLs, set time of day clocks, respond to messages, monitor status, and detect and recover wait states.

I/O Operations

I/O operations provide a single point of control for managing connectivity in your active I/O configurations. It takes an active role in detecting unusual I/O conditions and lets you view and change paths between a processor and an input/output device, which can involve using dynamic switching, such as the enterprise systems connection (ESCON®) or Fibre Channel connection (FICON®) switch. I/O operations changes paths by letting you control channels, ports, switches, control units, and input/output devices. You can do this through an operator console or API.

1.2 System Automation for z/OS V3.1 new functions

IBM Tivoli System Automation for z/OS V3.1 can help increase z/OS application availability, and automate I/O, processor, and system operations. IBM Tivoli System Automation for z/OS V3.1 is easier to use than ever, providing:

- Integration with IBM Tivoli OMEGAMON
- Enhanced Geographically Dispersed Parallel Sysplex™ (GDPS) integration
- Integration with IBM Tivoli System Automation for Multiplatforms
- Reduced automation implementation time and cost through self-configuration advances customization dialog enhancements and sample policies
- Enhancements to some System Automation for z/OS commands
- Improved CICS and IMS product automation
- NetView Management Console (NMC) enhancements

1.2.1 Integration with IBM Tivoli OMEGAMON

IBM Tivoli OMEGAMON interface for System Automation for z/OS lets you gather performance data on the system into the automation system. Data can be collected from the following performance monitoring products:

- OMEGAMON for MVS
- OMEGAMON for CICS
- OMEGAMON for IMS
- OMEGAMON II® for DB2

The data is collected from the exception analysis feature. Exception analysis is an OMEGAMON feature that analyzes system metrics and compares them against predefined thresholds in the system. When a threshold is exceeded, an exception event is shown. This event is commonly called an exception.

The exception analysis is an OMEGAMON 3270 interface feature. It displays the exception on the OMEGAMON application. System Automation for z/OS can establish direct VTAM communications with the four OMEGAMON subsystems. The interface is performed through the LU 2 interface to the OMEGAMON classic interface. These VTAM links enable System Automation for z/OS to directly receive OMEGAMON exceptions and to react to these exceptions accordingly by means of user-written responses. These responses can be running programs or issuing commands, including issuing commands back to the host OMEGAMON.

You can use policy definitions to instruct System Automation for z/OS to query a particular OMEGAMON at regular intervals for an exception and then take an action when that exception occurs. For example, you could trap an OMEGAMON for CICS PAGE exception, which means the page-in rate is too high. When this exception is received, a user written reaction routine can respond to reduce the system load.

The automation administrator can use the INGOMX interface to send OMEGAMON commands and receive their responses in System Automation for z/OS. For example, you might have trapped an exception that indicates that a job is in a wait condition. You can then use OMEGAMON commands to provide more informtion about the exception and the job, or take actions such as cancelling the job.

The concept of the monitoring resource that was introduced in System Automation for z/OS V2.3 to provide health information about the resource has been extended to provide exception data from the corresponding OMEGAMON monitor. This is achieved by a new layer, referred to as the OMEGAMON API. The OMEGAMON API serves the following purposes:

- It establishes sessions with the various OMEGAMONs using logon data that is specified in the PolicyDB.
- It monitors the sessions and re-establishes a session if necessary.
- It acts as the bridge between the monitoring resource and the OMEGAMON that the resource monitor is linked to. The communication is semidirectional.
- It is always the monitoring resource that asks the appropriate OMEGAMON to do something and then waits for the response from the command.

The OMEGAMON exception processing is shown in Figure 1-2 on page 7. More discussion on setting up this interface is provided in Chapter 3, “Working with IBM Tivoli OMEGAMON classic interface” on page 33.

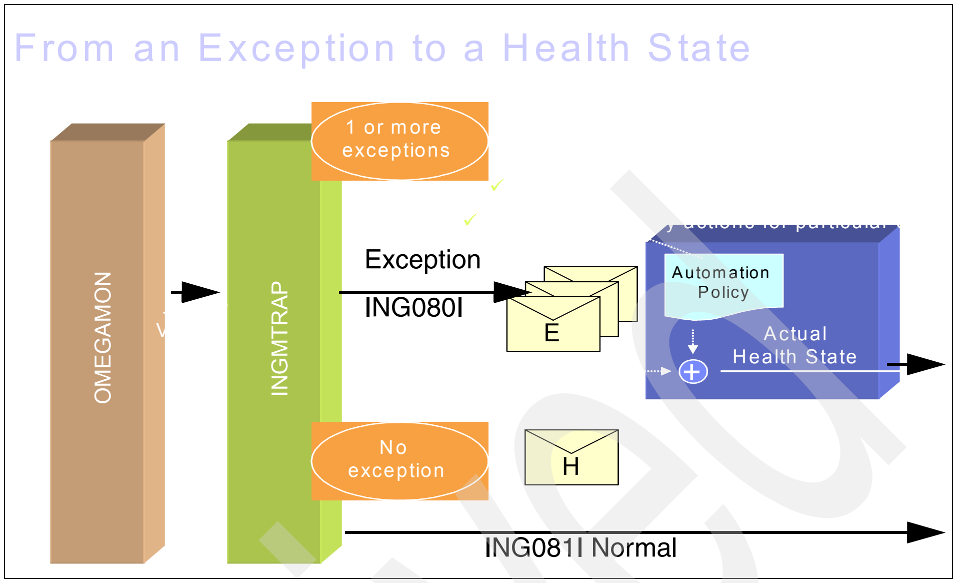


Figure 1-2 OMEGAMON exception processing

1.2.2 Enhanced GDPS integration

When working with GDPS, a predefined GDPS environment and definitions will significantly reduce setup time for System Automation for z/OS and NetView for z/OS. This is achieved by:

- A sample NetView for System Automation for z/OS startup procedure that contains all System Automation for z/OS and GDPS specific elements.
- Inclusion of the GDPS-specific messages in the System Automation for z/OS provided message table INGMSG01, thus eliminating the need for you to customize the message table for use by GDPS.
- A sample policy that includes all definitions for a complete GDPS environment.
- Sample z/OS PARMLIB members necessary for GDPS.
- Automatic disabling of System Automation for z/OS provided automation functions that are inconsistent with GDPS when GDPS is installed. This eliminates the need for you to manually turn off the appropriate function in System Automation for z/OS, such as:
  - Automation when the system leaves the sysplex (IXC102A message)
  - CDS recovery

1.2.3 Integration with Tivoli System Automation for Multiplatforms

With Tivoli System Automation for Multiplatform, z/OS based applications automation can now be integrated with end-to-end automation of heterogeneous on demand applications. This allows you to:

- Ease operations through a Web-based single point of control across z/OS, Linux®, and AIX®
- Increase application availability by resolving cross-platform dependencies
- Self-configuration advances with plug and play automation modules
- The plug and play automation modules include base and add-on policies. A collection of twelve add-on policies, based on best practices, can help reduce time and effort to create a new or update an existing policy. New plug and play automation modules help you to:
  - Increase WebSphere® Application Server for z/OS V5.1 availability and ease operations
  - Implement a Geographically Dispersed Parallel Sysplex

Automation is performed on two levels. The first level automation domains have local automation technology of their own, such as System Automation for z/OS and System Automation for Multiplatform. These first-level automation domains are connected to the end-to-end automation manager through an automation adapter.

The end-to-end automation manager is provided by IBM Tivoli System Automation for Multiplatforms. It provides the following:

- Continuous availability for heterogeneous distributed IT business applications
- Reduces the total cost of ownership

The automation adapter acts as the link between the end-to-end automation manager and its first-level automation domain, such as System Automation for z/OS in a sysplex group. The purpose of the automation adapter is to:

- Monitor resources within its first-level automation domain
- Propagate resource attribute changes to the end-to-end automation manager
- Start and stop resources within the first-level automation domain by request of the end-to-end automation manager
- Provide information about resources that are available within the first-level automation domain in response to queries from operators

1.2.4 Reduced implementation time

Reducing implementation time for System Automation for z/OS is achieved from several feature enhancements, such as:

- Enhancements to the customization dialog:
  - Adding, updating, or deleting large amounts of policy data.
  - As an alternative to entering or modifying policy data with the customization dialog, you can now create sequential data sets with policy data specifications in a newly designed text format. These sequential data sets allow you to see multiple policy items from multiple database entries at a glance, but in contrast to database reports, they can be edited and, as long as you do not violate the syntax designed for them, they can be imported into existing policy databases.
  - Two keywords in the sequential data set, new and update, control whether the import will add entries to the policy database or modify existing entries in it. Data sets for import into policy databases can be created either “from scratch” or by exporting parts of an existing policy database.
  - In addition to deleting single policy objects one by one, you can now delete any number of policy objects in one action. With this new feature, you can specify how often you want to be asked for confirmation:
    - Confirmation for each policy object that is to be deleted.
    - Confirmation only for those policy objects that are still connected to other policy objects (using links, membership, class instantiation, and so on).
    - No confirmation at all.
    - If you confirm the deletion of an object that is still connected to other objects, any connections to it will also be removed.
- Policy database (PDB™) import

  The PDB Import function, provided in System Automation for z/OS V2.3, allows you to import one or multiple entries of a single entry type from another PDB into the current one, although the data imported does not contain any link information.

  This restriction has now been removed. When importing a policy entry, the associated policy objects that are referenced using an entry-type link or class/instance links are also copied. This allows you to import entire “logical” units in a single import operation. The entire application group and everything that “belongs” to it is copied in the target Policy database.

The PDB Import function supports the inclusion of the “linked” objects for the following entry types:

APG	Application group with APLs, SVPs, and TRGs
APL	Application instance with class APL, or class APL with its APL instances, both with linked SVPs, TRGs, CSAs, and ISAs
TRG	Trigger with EVTs

- Streamlining of policy definitions

  The AUTOMATION INFO policy that is currently attached to the APL entry type has been retired. The APPLICATION INFO policy has been updated to contain all the input fields that were previously available on the AUTOMATION INFO policy and a field for ‘startup parameters’.

  The NETVIEW and AUTOMATION SETUP policies that are currently attached to the SYS entry type have been retired. The SYSTEM INFO policy has been updated to contain all the input fields previously available on the NETVIEW and AUTOMATION SETUP policies.

Additional policy enhancements include:

- Concurrent multi-user access to system definitions.

  Concurrent write access by multiple users is now also possible for policy objects of type SYS.

- Report member consolidation.

  Currently there are several members created by the Customization Dialog in the Policy Build Output data set either for special reports, for example, the unlinked report, or as a log for special, long-running tasks, for example, the control file build. The report members are now written to the newly introduced report data set.

- Supporting system symbols and AOCCLONES.
- The OPC CONTROL policy has been enhanced to allow input of symbols in the subsystem ID field.
- Subtype for applications of type STANDARD.

  The support of subtypes, previously available only for nonstandard applications, is now extended to applications of all types, for example, to use it as a filter for the INGLIST command output.

- Sample policies – Best practices.
  - Developing an automation policy requires knowledge of the automated product (for example, WebSphere, SAP) as well as knowledge of the automating product (System Automation for z/OS).

- Best practice policies are delivered with System Automation for z/OS V3.1 where both aspects are combined: Using System Automation for z/OS concepts, best operating practices are stored as integrated sample policies. This enables you to get productive much faster, also in the case of migrating from competitive products (significantly reduced time to value).
- Even if you have developed your own automation, you may use System Automation for z/OS best practice policies as a proof point.
- If you start working with a new product, you may use System Automation for z/OS best practice automation policies as a valuable knowledge base of product topology and dependencies.
- System Automation for z/OS best practice policies provide z/OS base automation as well as middleware or application add-on automation that can be tailored to your needs.
- Best practice policies can be added on top of your existing automation, thus allowing for a stepwise enhancement.

1.2.5 Enhancements to System Automation for z/OS commands

These commands has been modified and enhanced:

- INGMOVE

  A new command called INGMOVE has been introduced that makes moving sysplex application groups a lot easier. Rather than forcing the operator to manipulate the preference value of each member in the sysplex application group, the operator simply specifies where the group should be moved to. In a sysplex application group of type move, only one member is active at a time. By specifying the new location of the move group, the active member is terminated and the member associated with the new location is activated.
- INGSESS

  A new command called INGSESS has been introduced that displays the OMEGAMON session definitions, the session status, and statistical information about the session. With this command, the operator is able to see the session details, both policy definitions and run time details, as well as to start or stop an OMEGAMON session.
- INGSTR
  - A new command called INGSTR has been introduced that relocates structures to their desired location. It fills the leak left when a coupling facility has been drained for maintenance purposes and enabled again later or a new CFRM policy has been activated with different preference lists.

  - The INGSTR command is modeled on the INGCF STRUCTURE command. The main difference to the existing INGCF command is that the new INGSTR command also shows the current location (or locations) and the preferred location (or locations) of each structure. In addition, the indicator '*' prefixes the preferred location (or locations) if the structure does not allow XCF to perform the reallocation.
- INGGROUP
  - The INGGROUP command has been enhanced to invoke user exit AOFEXC13 to allow the installation to check whether the issuer of the command is authorized to perform the action. This is similar to the other command exits that where introduced in former releases of System Automation for z/OS.
  - A new option for the ACTION parameter has been introduced that shows the policy settings, such as the group type list of systems excluded or to be avoided for the specified resource groups. With the ACTION=POLICY parameter you can see the current attributes that are assigned to the resource group.
- INGMON

  The INGMON command is the main interface for telling System Automation for z/OS the health status for a particular monitor resource. Its primary use is from within the NetView automation table when trapping a message that can be directly mapped into a health status.

  The command has been enhanced so that it can be coded into the NetView automation table to issue commands in response to an OMEGAMON exception.
- INGMTRAP

  The INGMTRAP command facilitates the use of INGOMX for monitoring that is related to Monitor Resources. It invokes the INGOMX command to trap a list of exceptions that are monitored by the given OMEGAMON monitor.
- INGREQ

  The INGREQ command has been enhanced to allow the operator to cancel a previously made request. The operator no longer needs to use the INGVOTE or INGSET command in order to remove a request. This is more logical because the INGREQ command is also used to inject a start or stop request for the resource. The source of the request to be cancelled is automatically determined in the same way as it is done when firing off a start or stop request.
- INGLIST

  A new parameter called MEMBERS has been added to the INGLIST command. It causes the members of the specified application group to be displayed. Previously, this was only possible when invoking the INGLIST command in full screen mode and specifying action code G in front of the application group.

- INGINFO
  - The INGINFO command has been enhanced to show the values assigned to user variables.
  - User variables can be associated with a resource by means of the INGVARS command. This eliminates the need to use the INGVARS command in order to see the user variables associated with the resource.
- INGTHRES

  The maximum number of errors before the threshold is reached is now 50. The limit was 10 in previous releases.
- INGVTAM

  Automation of the IMS Applid has been integrated into the base INGVTAM service. There is no need to register the IMS subsystem using the INGVTAM command because this is automatically done if a major node or nodes is or are specified in the IMS Control Policy.
- DISPINFO
  - The DISPINFO command has been enhanced to display more information from CICS, IMS and VTAM:
    - For CICS, the CICSPlex® information, UOW checking, last start type, and last abend code are displayed.
    - For IMS, the IMSID, CQS, FDR, AVM, DC status, and last abend code are displayed.
    - For VTAM, the current and desired status of VTAM Applids for subsystems that are registered to INGVTAM are displayed.
- DISPSYS

  The DISPSYS command has been enhanced to show the messages captured for MVSESA.
- ACF
  - The ACF command has been enhanced to allow the operator to reload the Automation Tables (ATs) that are specified in the Policy DB, thereby disabling the sections in the automation tables that do not apply due to the active configuration.

  - The ACF CHECK option has been introduced to allow the operator to check the Automation Agent Configuration data for consistency. The following checks are made:
    - Matching Configuration token
    - Automation Tables specified in the Policy DB for errors
    - Other configuration validation (correct format of ACF fragments)
- AOCTRACE

  The AOCTRACE command has been enhanced to gather message attributes for a particular message without disturbing automation. This is done by activating message tracing. Once message tracing is active for a particular message and the message is passed to NetView (AT processing), all attributes that are associated with the message are shown. Examples are job name, job number, and MCS flags. This makes debugging a lot easier if the coded message trap does not work.
- ASF and ASFUSER
  - The restriction of the ASF and ASFUSER commands not being PIPEable has been removed. The ASFUSER command supports up to 40 user fields.
  - The structure of the Automation Status File (ASF) has been changed so that it now supports up to 50 error timestamps.
- ISQIPSWT
  - A new command called ISQIPSWT has been introduced to allow you to change the IP adapter address for the connection to the Support Element (SE), thus using the alternate adapter's IP address. Without this support, you must terminate ProcOps, which will affect all ProcOps connections, update the single changed IP address in the ProcOps policy, perform a ProcOps build, and finally issue a ProcOps cold start.

Note: With APAR OA09982, the interesting ProcOps ISQVARS variables SEADDR and HMCADDR have been modified as read/write. This allows the adapter IP address to be changed manually:

- Firstly, close all active ProcOps connections for all target systems for the affected target hardware using the ISQXCLS command.
- Update ISQVARS variable SEADDR with the new IP address.
- Restart all previously closed connections using the ISQXIII command.

  - The loss of communication because of an SE adapter failure cannot be broadcast to ProcOps because there is not a second connection active that can be used for the purpose of internal integrated connection recovery. Therefore, the new command is offered as an optional recovery aid. The command is also applicable in situations when a Support Element's IP address has to be changed because of a service action rather than an adapter failure.

- OPCAQRY

  The OPCAQRY command has been rewritten so that it provides the same look and feel as the other System Automation for z/OS commands. Furthermore, the command provides linemode output and is PIPEable.

1.2.6 Cleanup of CICS and IMS message exit

The message policy definition within the System Automation for z/OS Customization Dialog has been enhanced to allow the automation administrator to define the messages to be exposed to automation from within CICS or IMS.

The administrator has one place to customize the behavior of the CICS or IMS message exit function. When the CICS or IMS subsystem starts, the message policy from the configuration is loaded into the CICS or IMS subsystem respectively and the message exit is enabled. In previous releases, the administrator had to assemble the message exit definitions into a load module and make the load module accessible to the CICS / IMS subsystem at run time.

A refresh of the configuration policy will force a refresh of the message exit definition in the appropriate CICS or IMS subsystem. Each CICS or IMS subsystem can have its own message exit definitions without affecting any other CICS or IMS subsystem.

The current implementation does not allow the administrator to reload Message Exit policies without a restart of CICS or IMS. The new solution allows you to reload policies with the INGAMS refresh function. Furthermore, only CICS or IMS subsystems that have had policy changes will be reloaded.

1.2.7 NMC enhancements

There are several enhancement for NetView Management Console in the area of profile distribution and performance enhancement.

A profile residing on the NMC server is automatically distributed to all clients when starting the NMC client. This is done by NetView using the same process that delivers fixes from the NMC server to the NMC client.

The System Automation for z/OS code has been modified to search both TDS/client/settings and TDS/client/lib directories for the profile. It searches the TDS/client/settings first. The advantage is that any update to the System Automation for z/OS profile within ingnmcpr.txt only needs to be applied to the NMC Server.

The RODM data feed that is triggered when a configuration refresh INGAMS REFRESH command is processed so that only the configuration changes are forwarded to the NMC focal point causing updates in RODM. Previously, the entire configuration – all resources, their relationships and status – was sent to the NMC focal point. Often this is unnecessary because the change in the configuration may not be relevant to RODM.

Sending just the relevant configuration data to the focal point and feeding RODM with the configuration changes results in better performance due to less system utilization. This approach avoids mass updates in RODM that can potentially disrupt NMC client users.

1.3 IBM Tivoli NetView for z/OS V5.2

System Automation for z/OS is based on IBM Tivoli NetView for z/OS. The current version of IBM Tivoli NetView for z/OS is Version 5.2.

Tivoli NetView for z/OS provides the key capabilities and advanced functions to help you maintain the highest degree of availability of your IBM System z networks. It offers an extensive set of tools for managing and maintaining complex, multivendor, multiplatform networks and systems from a single point of control. NetView for z/OS brings advanced automation, support for both TCP/IP and SNA networks, and a set of UIs to meet every user's need. It also provides management functions that work in cooperation with other products.

The features and capabilities of NetView for z/OS V5.2 are:

- Networking and Automation: IBM Tivoli NetView for z/OS V5.2 provides strengthened TCP/IP management capabilities with support for IPv6, automated response to intrusions, more in-depth IP connection management, and support for network address translation, to name just a few. It also provides an automated message attribute handling that extends NetView for z/OS message automation to facilitate and simplify operator interactions
- Enhanced integration with Tivoli Enterprise™ Portal, correlated with performance data from OMEGAMON for Mainframe Networks. The NetView Web application is significantly expanded. In addition to broad ease-of-use enhancements, major new functions include management of TCP/IP availability and performance data from a single console through extensive

integration and interoperability with OMEGAMON for Mainframe Networks, allowing clients to easily manage their IP environment and the ability to easily create base trouble tickets including with IBM Tivoli Information Management for z/OS and others

- Client time to value and ease of use
  - Extensive enhancements in the NetView for z/OS Style Sheet capabilities help clients more easily manage the complete NetView platform. Included are enhancements that help to quickly identify and manage parameter values, tower settings, and other configuration information that are in effect, and to ease the migration to the latest release.
  - Migration of command definitions, including client-defined commands, is greatly simplified.

2

System Automation for z/OS V3R1 configuration

This chapter discusses the basic configuration of IBM Tivoli System Automation for z/OS in our environment. The discussion is divided into the following sections:

- 2.1, “Overview of configuration tasks” on page 20
- 2.2, “Preparation of System Automation for z/OS” on page 20
- 2.3, “Creating System Automation for z/OS dialog screens” on page 21
- 2.4, “Preparing the z/OS system environment and library” on page 22
- 2.5, “VTAM major node definition” on page 28
- 2.6, “Customize common automation DSIPARM library” on page 30
- 2.7, “Defining policy database” on page 31
- 2.8, “Automate starting the tasks” on page 31

2.1 Overview of configuration tasks

In this chapter, we lay out the important steps of the configuration that we perform on System Automation for z/OS in our environment. As System Automation for z/OS is an application based on IBM Tivoli NetView, we assume that NetView has already been installed and customized. The discussion in this chapter only relates to what must be done to customize System Automation for z/OS.

For the purposes of this IBM Redbook, we assumed that most installations would be running two NetViews, one being network automation NetView and System Automation for z/OS NetView. Because of this, we define the System Automation for z/OS as a secondary NetView.

We also use system cloning techniques. System cloning enables the use of variables or symbolics in procedures, data sets, VTAM major node definitions, and so forth. This allows for a single procedure defined in a common library to be executed on multiple systems.

Note: If you are running two NetView programs on the same system, refer to *Tivoli NetView for z/OS Installation: Configuring Additional Components*, SC31-8874.

We follow the configuration step describes in *IBM Tivoli System Automation for z/OS V3.1 Planning and Installation*, SC33-8261.

2.2 Preparation of System Automation for z/OS

System Automation for z/OS V3R1 is supplied in a Custom-Built Product Delivery Offering (CBPDO) with product number 5751-CS3. It is a separate package from the base z/OS operating system. System Automation for z/OS installation is performed using System Modification Program/Extended (SMP/E).

The typical System Automation for z/OS datasets high-level qualifier is ING. System Automation for z/OS provides a lot of sample definitions for your reference, including system parameters, startup procedures, and clists in the sample datasets.

Note: In the ITSO environment, the creation process of the common automation libraries involves working through the System Automation for z/OS install procedures and populating the common libraries with exploitation of system symbolics to avoid the need for local customization as far as possible in the user libraries ING.USER.* . With this approach, it was found that very little customization on the local domain specific: ING.USER.&domain and ING.USER.VSAM.&domain library level was required to establish the system automation environment.

System Automation for z/OS preparation starts with allocating the necessary datasets. The allocation is performed using these customization jobs from the System Automation for z/OS library, ING.SINGSAMP. These jobs are:

INGALLC0	Allocates system-unique data sets for System Operation for NetView
INGALLC1	Allocates system-unique data set for I/O Operation
INGALLC2	Allocates system-unique data set for Automation Agent
INGALLC3	Allocates system-unique data set for Automation Managers
INGALLC4	Allocates sysplex-wide data set for Automation Agent

2.3 Creating System Automation for z/OS dialog screens

The System Automation for z/OS policy database contains all automation policies and interfaces with the System Automation for z/OS engine. The policy database is managed by a set of ISPF-based dialog. These dialogs must be implemented before you define the System Automation for z/OS policy.

The System Automation for z/OS dialog datasets are defined using the sample job INGEDLGA in SINGSAMP. This job allocates data sets required for the I/O operations and the customization dialog. These data sets are normally allocated only on the focal point system, where you use the customization dialog.

You must include the data sets for z/OS system, processor, and I/O operations as follows:

- Systems operations:

AOFTABL	ISPF customization table for customization dialog
SOCNTL	System operations control file

- Processor operations:

AOFTABL	ISPF customization table for customization dialog
POCNTL	Processor operations control file
POLOG	Processor operations control file log

- I/O operations:

IHVCONF	I/O operations configuration file

For further information about how to Install and customize the ISPF Dialog Panels, refer to the *IBM Tivoli System Automation for z/OS V3.1 Planning and Installation*, SC33-8261.

2.4 Preparing the z/OS system environment and library

Now the z/OS system environment and the procedure library need to be modified for System Automation for z/OS. The modifications are:

2.4.1 Modify the system parameter library

The modification for system parameter, typically in SYS1.PARMLIB, is as follows:

- Modify PROGxx to include the following datasets as APF authorized libraries:
  - ING.SINGMOD1
  - ING.SINGMOD2
  - ING.SINGMOD3
- Modify PROGxx to include the following datasets as LNKLST libraries:
  - ING.SINGMOD1
  - ING.SINGMOD2
- Modify LPALSTxx to include the following library:
  - ING.SINGMOD3
- Modify/Check IEFSSNxx to include the additional subsystem names. The following subsystem strings were used for the automation environment within the ITSO project:

```
AOFA
```

Note: If the IPL of the z/OS system is not possible for a longer time frame, all the above modified parameters can be made effective dynamically by using the following z/OS console commands:

```
SETPROG APF,ADD,DSNAME=ING.SINGMOD1
SETPROG APF,ADD,DSNAME=ING.SINGMOD2
SETPROG APF,ADD,DSNAME=ING.SINGMOD3
SETPROG LNK,DEFINE,NAME=LINKITSO,COPYFROM=CURRENT
SETPROG LNK,ADD,DSNAME=ING.SINGMOD1,NAME=LINKITSO
SETPROG LNK,ADD,DSNAME=ING.SINGMOD2,NAME=LINKITSO
SETPROG LNK,ACTIVATE,NAME=LINKITSO
SETPROG LPA,ADD,DSNAME=ING.SINGMOD3,MODNAME=(*),MASK(*)
SETSSI ADD,SUBNAME=AOFA
```

2.4.2 Modify the system procedure library

The system library, typically called SYS1.PROCLIB, contains the definitions of the started tasks. System Automation for z/OS uses the following started tasks that must be defined in the procedure library:

- Automation Manager
- Automation subsystem Interface
- Automation subsystem Agent

The subsystem interface and subsystem agent are NetView address spaces that you may already have. You must change these members to add additional required datasets for System Automation for z/OS. If you have an existing member, we recommend that you back up the startup procedure members that you are going to change.

We copied the following members from the SINGSAMP data set to members of our PROCLIB and followed the customization instructions that are contained within these members:

- INGPIXCU: Access to XCF Utilities
- INGPHOM: Access to HOM Interface
- INGPIPLC: Access to Spare Couple Data Sets, User-Defined Couple Data Sets and Spare Local Page Data Sets
- HSAPIPLC: Access to IPL Information

Additional details can be found in the detail in *IBM Tivoli System Automation for z/OS V3.1 Planning and Installation*, SC33-8261.

Automation manager procedure

We copied a sample startup procedure called INGEAMSA from the data set SINGSAMP into our PROCLIB. All required data sets were allocated there.

Example 2-1 shows the Automation Manager procedure of our case study scenario.

Example 2-1 Our Automation Manager procedure (INGEAMSA)

```
//INGEAMSA PROC TYPE=HOT,          Start type: HOT | WARM | COLD
//              D=0,               Default start delay is none
//              M=00,              Default parmlib suffix is 00
//              P=NO,              Default is NO prompting
//              HLQINST='ING',     SMP/E installed target lib
//              HLQ='ING.USER'     HLQ of shared and unique AM
//*
//AMSA     EXEC PGM=HSAPINIT,REGION=200M,TIME=1440,               X
//             PARM='MEMBER=&M,START=&TYPE,DELAY=&D,PROMPT=&P'
//*
//*-----------------------------------------------------------------
//* HSAMODLE - Non-APF Authorized library concatenation TASKLIB
//*-----------------------------------------------------------------
//HSAMODLE DD  DSN=&HLQINST..SINGMOD1,DISP=SHR       SA z/OS
//*
//*------------------------------------------------------------------
//* HSAOVR  -  Required schedule override file (shared)
//*------------------------------------------------------------------
//HSAOVR   DD  DSN=&HLQ..VSAM.HSAAMOVR,DISP=SHR
//*
//*------------------------------------------------------------------
//* HSACFGIN - Required file that saves AM warm start info (shared)
//*             Note: This is NOT the Automation Control File
//*------------------------------------------------------------------
//HSACFGIN DD  DSN=&HLQ..SHSACFGO,DISP=SHR
//*
//*------------------------------------------------------------------
//* HSAPLIB  - Parameter library containing HSAPRMxx member
//*------------------------------------------------------------------
//HSAPLIB DD  DSN=&HLQ..PARMLIB,DISP=SHR
//*
//*------------------------------------------------------------------
//* SYSOUT DATASET used by LE
//*------------------------------------------------------------------
//SYSOUT   DD  DUMMY
//SYSPRINT DD  DUMMY
```

```
//*
//*-------------------------------------------------------------------
//* CEEDUMP  - is used by the Language Environment Dump Services.
//*            CEEDUMP must be a sequential data set.
//*            See below for the recommended size of the data set.
//*-------------------------------------------------------------------
//CEEDUMP  DD  DUMMY,SPACE=(TRK,(30,100)),
//             DCB=(RECFM=FB,LRECL=133,BLKSIZE=13330)
//*
//*-------------------------------------------------------------------
//* TRACE DATASETS. Uncomment in case of needed trace information
//*-------------------------------------------------------------------
//*TRACET0  DD  DSN=&HLQ..&SLQ..TRACET0,DISP=SHR
//*TRACET1  DD  DSN=&HLQ..&SLQ..TRACET1,DISP=SHR
```

Note: The Automation Manager must be started cold on the first startup. The cold start is performed by default, unless you specify the warm start option. Do not select the warm start option, because you might have policy data from earlier releases in your warm start cache.

Additional details can be found in the detail in I*BM Tivoli* System Automation for z/OS*for z/OS V3.1 Planning and Installation*, SC33-8261.

Automation subsystem interface

NetView provides a sample subsystem interface startup procedure in member CNMSJ010. Copy this member from your CNMSAMP NetView library and adapt it to your needs.

Example 2-2 shows the Automation subsystem Interface procedure of our case study scenario.

Example 2-2 Our System Automation for z/OS *subsystem Interface*

```
//CNMPSSI PROC SQ1='NETVIEW', ** SYSTEM DSN HIGH LEVEL QUALIFIER
//       PROG=CNMINIT,        ** PGM USED TO START NETVIEW SUBSYSTEM
//       REG=1250,            ** REGION SIZE(IN K)
//       MBUF=4000,           ** NUMBER OF MESSAGE BUFFERS TO USE
//       CBUF=200,            ** NUMBER OF COMMAND BUFFERS TO USE
//       DSIG='',             ** Subsystem command designator
//       MSGIFAC='SYSTEM',    ** SSI/EXTENDED CONSOLE OVERRIDE SWITCH
//       PPIOPT='PPI',         ** PPI OPTIONS SWITCH
//       ARM='*NOARM',        ** AUTOMATIC RESTART (ARM) USAGE
//       PFXREG='ONE',        ** Prefix Registration option.
//       P256BUF=300,         ** Number of 256 byte PPI buffers to use
//       P4000BUF=0           ** Number of 4000 byte PPI buffers to use
//SAV3R1  EXEC PGM=&PROG,TIME=1440,REGION=&REG.K,
//         PARM=(&MBUF,&CBUF,'&DSIG','&MSGIFAC','&PPIOPT','&ARM',
//          '&PFXREG',&P256BUF,&P4000BUF),DPRTY=(13,13)
//STEPLIB  DD   DSN=&SQ1..CNMLINK,DISP=SHR
```

Additional details can be found in the detail in *IBM Tivoli* System Automation for z/OS *V3.1 Planning and Installation*, SC33-8261.

Automation subsystem agent

You can use the sample provided in the INGENVSA member of the SINGSAMP data set. Copy it to a member of each system's PROCLIB data set (the one of the focal point system as well as the ones of the target systems).

Example 2-3 shows the Automation subsystem Agent procedure of our case study scenario.

Example 2-3 Our automation subsystem agent

```
//INGENVSA PROC DOMAIN=&SYSNAME.N, ** NETVIEW DOMAIN NAME
//       DOMAIN=&SYSNAME.N, ** PGM FOR AUTOMATION NETVIEW
//       PROG=DSIMNT,       ** PGM FOR AUTOMATION NETVIEW
//*      PROG=BNJLINTX,     ** PGM FOR NETWORK OR COMBINED NETVIEW
//       SQ1='ING',         ** SA Z/OS DSN HIGH LVL QUALIFIER
//       SQ2='NETVIEW',     ** NETVIEW DSN HIGH LVL QUALIFIER
//       Q2='ING.USER',     ** SAUSER DSN HIGH LEVEL QUALIFIER
//       Q1='NETVUSER',     ** NETVIEW USER DSN HIGH LEVEL QUALIFIER
//       SUBSYM='',         ** SYMBOLIC SUBSTITUTION SWITCH
//       NV2I=''
//*
```

```
//NETVIEW  EXEC PGM=&PROG,TIME=1440,
//         REGION=0M,
//         PARM=(24K,200,
//           '&DOMAIN','','','&SUBSYM','&NV2I'),
//         DPRTY=(15,15)
//*
//DSICLD   DD   DSN=&Q2..REXX,DISP=SHR              NETVIEW TARGET
//         DD   DSN=&SQ1..SINGNREX,DISP=SHR         SA TARGET
//         DD   DSN=&SQ1..SINGSAMP,DISP=SHR         SA TARGET
//         DD   DSN=&SQ2..CNMSAMP,DISP=SHR          NETVIEW SAMPLIB
//         DD   DSN=&SQ2..CNMCLST,DISP=SHR          NETVIEW SAMPLIB
//*
//DSIOPEN  DD   DSN=&SQ2..SDSIOPEN,DISP=SHR
//*
//DSIPARM  DD   DSN=&Q2..DSIPARM,DISP=SHR           SA      TARGET
//         DD   DSN=&SQ1..SINGNPRM,DISP=SHR         SA SAMPLE
//         DD   DSN=&SQ2..DSIPARM,DISP=SHR          NETVIEW SAMPLE
//*
//DSILIST  DD   DSN=&Q2..&DOMAIN..DSILIST,DISP=SHR  USER TARGET
//DSIASRC  DD   DSN=&Q2..&DOMAIN..DSIASRC,DISP=SHR  USER TARGET
//DSIARPT  DD   DSN=&Q2..&DOMAIN..DSIARPT,DISP=SHR  USER TARGET
//*
//DSIVTAM  DD   DSN=SYS1.SANDBOX.VTAMLST,DISP=SHR
//*
//DSIPRF   DD   DSN=&SQ1..SINGNPRF,DISP=SHR         SA TARGET
//         DD   DSN=&SQ2..DSIPRF,DISP=SHR           NETVIEW TARGET
//*
//DSIMSG   DD   DSN=&SQ1..SINGNMSG,DISP=SHR         SA US
//         DD   DSN=&SQ2..SDSIMSG1,DISP=SHR         NETVIEW TARGET
//*
//BNJPNL1  DD   DSN=&SQ2..BNJPNL1,DISP=SHR
//*
//BNJPNL2  DD   DSN=&SQ2..BNJPNL2,DISP=SHR
//*
//CNMPNL1  DD   DSN=&SQ1..SINGNPNL,DISP=SHR         SA US
//         DD   DSN=&SQ2..CNMPNL1,DISP=SHR          NETVIEW US
//*
//AOFSTAT  DD   DSN=&Q2..VSAM.&DOMAIN..STATS,DISP=SHR
//*
//HSAIPL   DD   DSN=&Q2..VSAM.&DOMAIN..IPLDATA,DISP=SHR
//*
//DSILOGP  DD   DSN=&Q1..&DOMAIN..DSILOGP,
//         DISP=SHR,AMP='AMORG,BUFNI=20,BUFND=20'
//DSILOGS  DD   DSN=&Q1..&DOMAIN..DSILOGS,
//         DISP=SHR,AMP='AMORG,BUFNI=20,BUFND=20'
```

```
//*
//DSITRCP  DD   DSN=&Q1..&DOMAIN..DSITRCP,DISP=SHR,AMP=AMORG
//DSITRCS  DD   DSN=&Q1..&DOMAIN..DSITRCS,DISP=SHR,AMP=AMORG
//*
//AAUVSPL  DD   DSN=&Q1..&DOMAIN..AAUVSPL,DISP=SHR,AMP=AMORG
//AAUVSSL  DD   DSN=&Q1..&DOMAIN..AAUVSSL,DISP=SHR,AMP=AMORG
//*
//BNJLGPR  DD   DSN=&Q1..&DOMAIN..BNJLGPR,DISP=SHR,AMP=AMORG
//BNJLGSE  DD   DSN=&Q1..&DOMAIN..BNJLGSE,DISP=SHR,AMP=AMORG
//*
//DSISVRT  DD   DSN=&Q1..&DOMAIN..DSISVRT,
//         DISP=SHR,AMP=AMORG
//INGDUMP  DD   DSN=&Q2..VSAM.&DOMAIN..INGDUMP,DISP=SHR
//*
//SYSPRINT DD   DUMMY
//*
```

Additional details can be found in the detail in *IBM Tivoli System Automation for z/OS V3.1 Planning and Installation*, SC33-8261.

2.5 VTAM major node definition

NetView for z/OS is a 3270 based application that is connected through VTAM. As such, it must use VTAM definition for its operation. In Example 2-4 on page 29, the VTAM definition for the PPT needs to change from PPO to SPO (this is the second NetView running on the system). We use system symbols in the major node definition. This allows for a single major node definition to be used for all systems in the sysplex.

For details about VTAM major node names and definitions, refer to Appendix B, “NetView Samples Overview”, in *Tivoli NetView for z/OS V5R2 Installation: Getting Started*, SC31-8872.

Example 2-4   VTAM major node definition member

```
VBUILD TYPE=APPL
*******************************************************************
* NETVIEW MAIN TASK                                               *
*******************************************************************
*
&SYSNAME.A    APPL AUTH=(VPACE,ACQ,PASS),                           X
               MODETAB=AMODETAB,DLOGMOD=DSIL6MOD,                   X
               APPC=YES,PARSESS=YES,                                X
               DMINWNL=4,DMINWNR=4,DSESLIM=8,VPACING=10,            X
               AUTOSES=2
*              STATOPT='NETVIEW'
*******************************************************************
* NETVIEW PRIMARY POI - (PROGRAM OPERATOR INTERFACE)              *
*******************************************************************
&SYSNAME.APPT APPL AUTH=(NVPACE,SPO),EAS=1,                         X
               MODETAB=AMODETAB,DLOGMOD=DSILGMOD
*              STATOPT='NETVIEW PPT'
*******************************************************************
* NETVIEW SUBTASKS                                                *
*******************************************************************
&SYSNAME.A* APPL AUTH=(NVPACE,SPO,ACQ,PASS),EAS=4,                  X
               MODETAB=AMODETAB,DLOGMOD=DSILGMOD
*              STATOPT='NETVIEW OOO'
*******************************************************************
* APPL DEFINITION STATEMENTS FOR TERMINAL ACCESS FACILITY         *
*******************************************************************
SCT&SYSCLONE.OPT APPL MODETAB=AMODETAB,EAS=9,                       X
               DLOGMOD=M3767
SCT&SYSCLONE.O*  APPL MODETAB=AMODETAB,EAS=9,                       X
               DLOGMOD=M3767
```

2.6 Customize common automation DSIPARM library

The modified members in the common automation DSIPARM library are:

- CxxSTGEN: This member overrides certain STYLE definitions from the CxxSTYLE member. You should modify this member instead of the original STYLE parameter so that updates to the STYLE will not override your customization. Our sample CxxSTGEN is shown in Example 2-5.

Example 2-5 Modified CxxSTGEN

```
SECOPTS.OPERSEC = SAFPW
SECOPTS.CMDAUTH = TABLE.CNMSBAK1
function.autotask.idleoff = *NONE*
TOWER =  SA  *AON  *MSM  *GRAPHICS  MVSCMDMGT NPDA NLDM TCPIPCOLLECT
  *AMI *TARA
TOWER.SA = SYSOPS PROCOPS
(MSM)AUTOTASK.?MSMdefault.InitCmd =
COMMON.AOFSENDALERT = NO
COMMON.AOFINITREPLY = 00:00:10
TASK.DSIWTOMT.INIT=N
RMTINIT.IP = Y
*RMTSYN.NET1.AOX14 = tvt1114.tivlab.raleigh.ibm.com
*RMTSYN.NET1.AOX15 = tvt1115.tivlab.raleigh.ibm.com
*%INCLUDE SA&DOMAIN.
```

- INGXINIT: This member is the initialization parameter for System Automation for z/OS. The content for our environment is shown in Example 2-6.

Example 2-6 Content of INGXINIT

```
*GRPID=XY
*MQM=SSSS
*LOGSTREAM=NO
PPI=YES
PPIBQL=5000
*DIAGDUPMSG=50
```

IBM Tivoli System Automation for z/OS V3R1 Planning and Installation, SC33-8261 provides instructions about what you need to change in the above members. For our case study environment, we followed the referenced manual with no problems.

2.7 Defining policy database

Before you can start using automation, you need to define your automation policy using the customization dialog. This involves the following actions:

- If applicable, migrate/merge existing policy information; you can use the sample job INGEBMIG in the SINGSAMP sample library.
- Add further policy definitions.
- Build a new policy database.

Distribute the policy definitions (the policy database) where required. If you start from scratch, use the IBM samples delivered with the product and create your new policy database. In such a case, read the information in the section “Creating a New Policy Database”, in *IBM Tivoli System Automation for z/OS Defining Automation Policy*, SC33-8262, which provides informtion about using the customization dialog for the required definitions.

IBM Tivoli System Automation for z/OS V3.1 provides a sample job named INGEBBLD in the SINGSAMP sample library to perform the build in batch mode. This job is configured according to your installation requirements. In our case study scenario, we used the ISPF Customization dialog.

The BUILD command is available from various screens of the customization dialog. For more informtion about how to perform this step, refer to *IBM Tivoli System Automation for z/OS V3.1 Defining Automation Policy*, SC33-8262.

Attention: Do not change the system operations control files manually. You must use the customization dialog to manipulate the policy objects.

2.8 Automate starting the tasks

IBM Tivoli System Automation for z/OS V3R1 initialization becomes automated and begins with starting System Operations. You accomplish this by making changes to the SYS1.PARMLIB data set member COMMNDxx.

Make sure that the procedure names you choose match those specified in the PROCLIB data set. Compare the contents of the COMMNDxx member with the INGECOM member, which resides in the SINGSAMP sample library.

Example 2-7 shows the COMMNDxx member of our scenario.

Example 2-7 COMMND63 Member

```
COM='SET MPF=00'
COM='MN JOBNAMES,T'
COM='K M,AMRF=Y'
COM='K S,DEL=RD,SEG=20,CON=N,RNUM=20,RTME=001,MFORM=M,L=01'
COM='S AOFASSI,SUB=MSTR'
COM='S AOFAPPL,SUB=MSTR'
COM='S INGEAMSA,SUB=MSTR'
```

Working with IBM Tivoli OMEGAMON classic interface

This chapter discusses the OMEGAMON classic interface of IBM Tivoli System Automation for z/OS in our environment. The discussion is divided into the following sections:

3.1 IBM Tivoli OMEGAMON integration

The OMEGAMON real-time monitor components alert you to application degradation and overall system problems through color-coded status bars or through user-defined status words. This includes a wide range of detailed analysis and exception reporting for every aspect of your enterprise. The valuable information that IBM Tivoli OMEGAMON suite of products convey to operations and support staff quickly becomes relied upon for the day to day running of a data center, making the availability of the Tivoli monitors themselves almost as important as the applications they monitor. The best way to achieve this is to entrust their availability to System Automation for z/OS and its excellent recovery, scheduling, and automation features.

A programmer-less integration with Monitor Resources is achieved by providing an interface tailored specifically for exception monitoring. The automation administrator uses this interface as a monitor command and specifies the exceptions of interest. In addition to the monitor command, the administrator also specifies the health state associated with the exception and commands that System Automation for z/OS should issue to recover from the exception. There are facilities that allow the administrator to escalate commands in case a previously scheduled recovery action was unsuccessful or to insulate from subsequent exceptions for a certain time while a recovery action is in progress.

Integrating OMEGAMON products into System Automation for z/OS gives the benefits of System Automation for z/OS's resource management and scheduling facilities:

- OMEGAMON error conditions and restart/recovery processing are handled by System Automation for z/OS, thus minimizing your application downtime.
- Having your OMEGAMONs integrated into System Automation for z/OS allows you to include the monitors in a fully-automated initial program load (IPL) for maximum efficiency.

A combination of System Automation for z/OS high availability functionality and the ability of IBM Tivoli OMEGAMON to move its Tivoli Enterprise Monitoring Server hub ensures the highest possible levels of application availability. For more discussion on moveable Tivoli Enterprise Monitoring Server hub, see *IBM Tivoli OMEGAMON XE V3.1.0 Deep Dive on z/OS*, SG24-7155.

Exception analysis is an OMEGAMON feature that monitors predefined thresholds in a system. Each time exception analysis is invoked, an exception is displayed on the OMEGAMON console if a threshold is exceeded. Using System Automation for z/OS, you can then act on these exception alerts by running programs or issuing commands back to the OMEGAMON system that has the problem. This feature is called Monitor Resources.

You can set up Monitor Resources to:

- Monitor sets of exceptions that may be of interest
- Set an application's health state based on the existence of such exceptions
- React to and resolve conditions that cause those exceptions

This chapter shows you how to define the sessions and the monitor resource definitions to set the System Automation for z/OS health status.

Various topologies are possible for System Automation for z/OS integration with IBM Tivoli OMEGAMON:

- There can be one or more systems.
- There can be one or more OMEGAMON monitors per system.
- Connectivity is through VTAM and the NetView Terminal Access Facility (TAF).

System Automation for z/OS can act as a focal point either:

- Globally, monitoring data from OMEGAMON monitors running on different systems
- Locally, monitoring data from OMEGAMON monitors running on the local system

The following assumptions are made about the topologies that we implement in this chapter. For background informtion about IBM Tivoli OMEGAMON implementation, refer to *IBM Tivoli OMEGAMON XE V3.1.0 Deep Dive on z/OS*, SG24-7155.

- IBM Tivoli OMEGAMON product is installed on each system.
- IBM Tivoli OMEGAMON classic monitors are installed and configured already to support multiple VTAM-based connections to it. For interoperability with System Automation for z/OS, support for 3270 model 2 (screen size 24x80) is required.
- IBM Tivoli OMEGAMON monitors are set up to interact with an external security product, such as RACF®, to validate access.
- IBM Tivoli OMEGAMON exceptions are reported when the threshold that is defined in OMEGAMON is exceeded. That threshold must be agreed within an installation because it must cater for the least severe condition that there might be an alert for.

Figure 3-1 shows a sample configuration involving a single system environment. In this configuration, each System Automation for z/OS acts as a focal point.

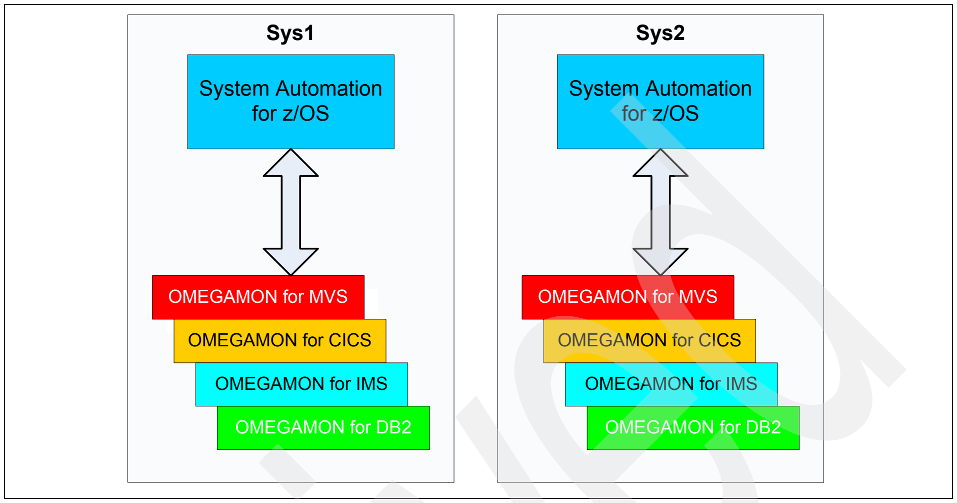


Figure 3-1 Single system environment

Figure 3-2 shows a sample structure for a multi system environment. Here you have a single System Automation for z/OS acting as the focal point for all systems.

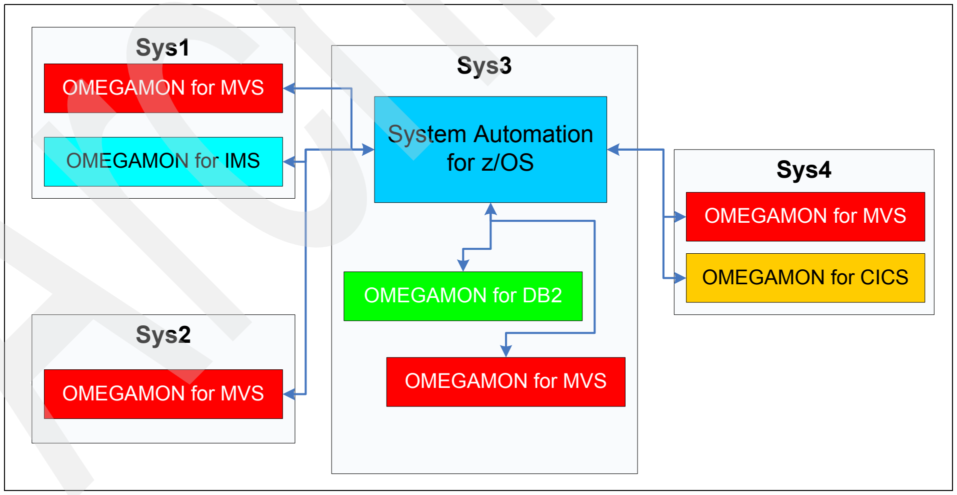


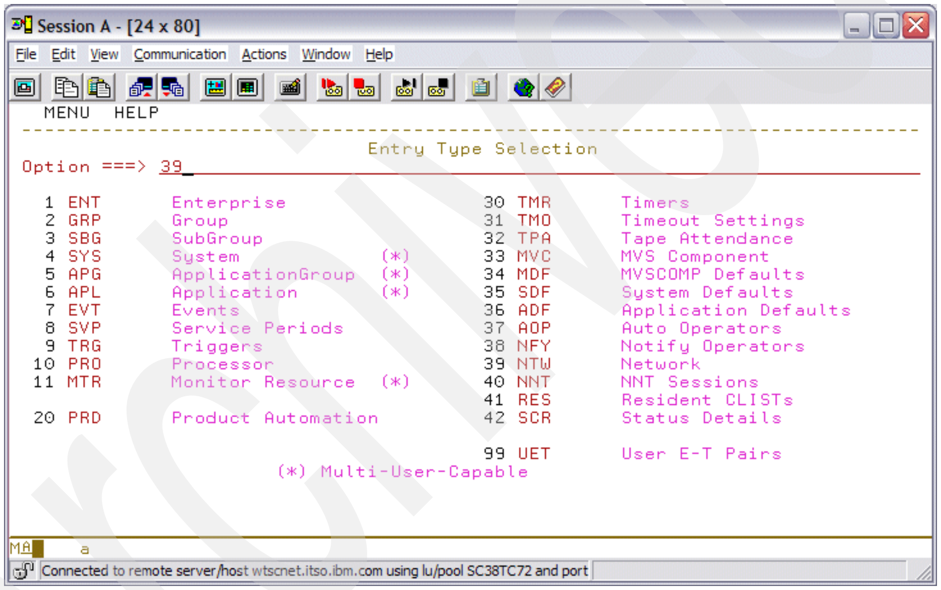
Figure 3-2 Multi systems environment

3.2 Setting up OMEGAMON integration

To set up the integration of System Automation for z/OS to IBM Tivoli OMEGAMON, we must define the OMEGAMON session and activate Terminal Access Facility (TAF).

3.2.1 Define the OMEGAMON network sessions

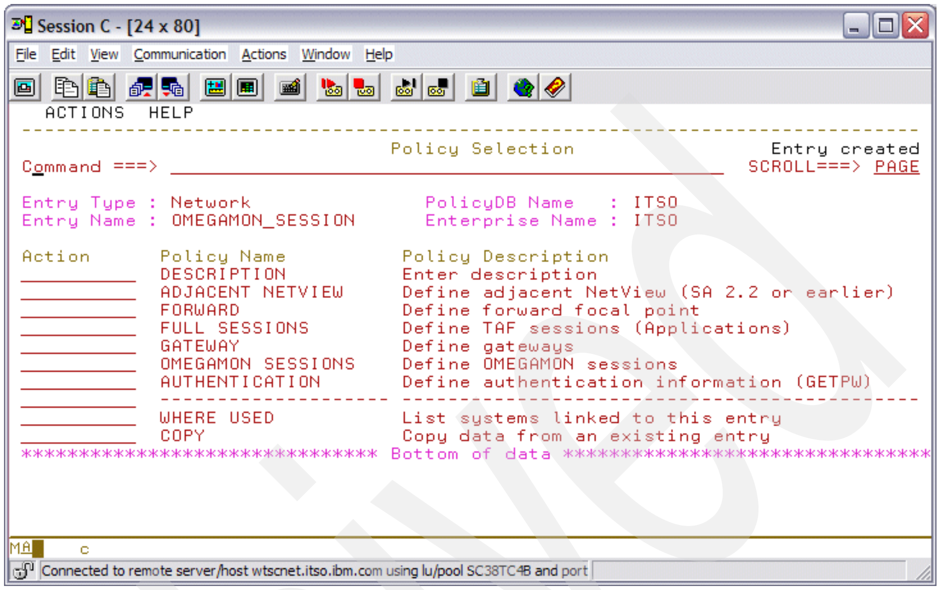
This section describes how to define the Network (NTW) policy definitions for a session between System Automation for z/OS and OMEGAMON classic monitor. From the entry type selection dialog shown in Figure 3-3, select option 39 for the NTW.



```
  MENU   HELP
 ------------------------------------------------------------------------------
                               Entry Type Selection
Option ===> 39_

  1  ENT       Enterprise                        30  TMR       Timers
  2  GRP       Group                             31  TMO       Timeout Settings
  3  SBG       SubGroup                          32  TPA       Tape Attendance
  4  SYS       System              (*)           33  MVC       MVS Component
  5  APG       ApplicationGroup    (*)           34  MDF       MVSCOMP Defaults
  6  APL       Application         (*)           35  SDF       System Defaults
  7  EVT       Events                            36  ADF       Application Defaults
  8  SVP       Service Periods                   37  AOP       Auto Operators
  9  TRG       Triggers                          38  NFY       Notify Operators
 10  PRO       Processor                         39  NTW       Network
 11  MTR       Monitor Resource    (*)           40  NNT       NNT Sessions
                                                 41  RES       Resident CLISTs
 20  PRD       Product Automation                42  SCR       Status Details

                                                 99  UET       User E-T Pairs
                         (*) Multi-User-Capable

MA        a
Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC38TC72 and port
```

Figure 3-3 Entry type selection screen

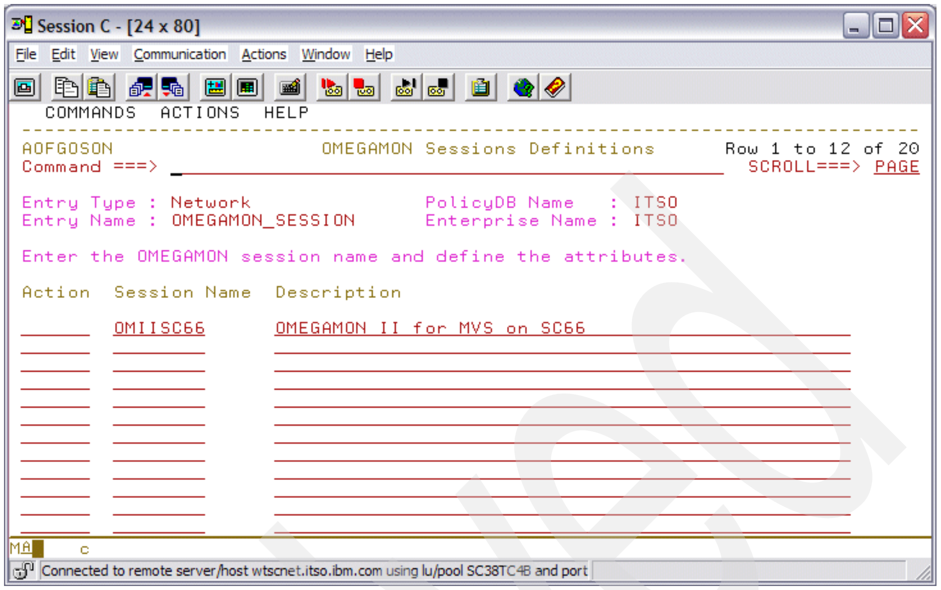
For the network session, we define a new entry called OMEGAMON_SESSION. Figure 3-4 shows the policy list for the OMEGAMON_SESSION.



```
  ACTIONS  HELP
------------------------------------------------------------------------------
                              Policy Selection                  Entry created
Command ===> ______________________________________________   SCROLL===> PAGE

Entry Type : Network               PolicyDB Name    : ITSO
Entry Name : OMEGAMON_SESSION      Enterprise Name : ITSO

Action      Policy Name          Policy Description
__________  DESCRIPTION          Enter description
__________  ADJACENT NETVIEW     Define adjacent NetView (SA 2.2 or earlier)
__________  FORWARD              Define forward focal point
__________  FULL SESSIONS        Define TAF sessions (Applications)
__________  GATEWAY              Define gateways
__________  OMEGAMON SESSIONS    Define OMEGAMON sessions
__________  AUTHENTICATION       Define authentication information (GETPW)
__________  -------------------  ---------------------------------------------
__________  WHERE USED           List systems linked to this entry
__________  COPY                 Copy data from an existing entry
******************************* Bottom of data ********************************

MA     c
Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC38TC4B and port
```

Figure 3-4 NTW policy definition screen

Select OMEGAMON SESSIONS policy. Figure 3-5 on page 39 shows the OMEGAMON sessions definition. This policy is used to define the attributes used to establish and run a communication session with OMEGAMON based on the NetView Terminal Access Facility (TAF).

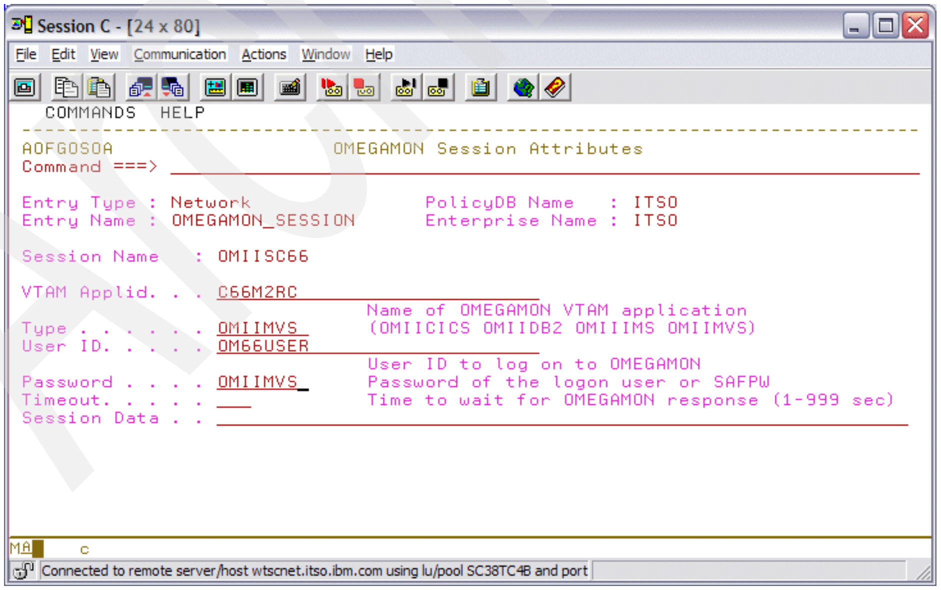

Session C - [24 x 80]
File Edit View Communication Actions Window Help

```
  COMMANDS  ACTIONS  HELP
------------------------------------------------------------------------------
AOFGOSON                  OMEGAMON Sessions Definitions         Row 1 to 12 of 20
Command ===> _                                                   SCROLL===> PAGE

Entry Type : Network                  PolicyDB Name   : ITSO
Entry Name : OMEGAMON_SESSION         Enterprise Name : ITSO

Enter the OMEGAMON session name and define the attributes.

Action   Session Name   Description

______   OMIISC66       OMEGAMON II for MVS on SC66
MA       c
```

Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC38TC4B and port

Figure 3-5 OMEGAMON sessions definitions screen

We define the OMIISC66 session definition that relates to connecting to IBM Tivoli OMEGAMON for z/OS on SC66. Figure 3-6 shows the definition of the OMIISC66 session.



```
  COMMANDS  HELP
------------------------------------------------------------------------------
AOFGOSOA                   OMEGAMON Session Attributes
Command ===>

Entry Type : Network                  PolicyDB Name   : ITSO
Entry Name : OMEGAMON_SESSION         Enterprise Name : ITSO

Session Name    : OMIISC66

VTAM Applid. . . C66M2RC
                           Name of OMEGAMON VTAM application
Type . . . . . . OMIIMVS   (OMIICICS OMIIDB2 OMIIIMS OMIIMVS)
User ID. . . . . OM66USER
                           User ID to log on to OMEGAMON
Password . . . . OMIIMVS_  Password of the logon user or SAFPW
Timeout. . . . . ___       Time to wait for OMEGAMON response (1-999 sec)
Session Data . .

MA       c
```

Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC38TC4B and port

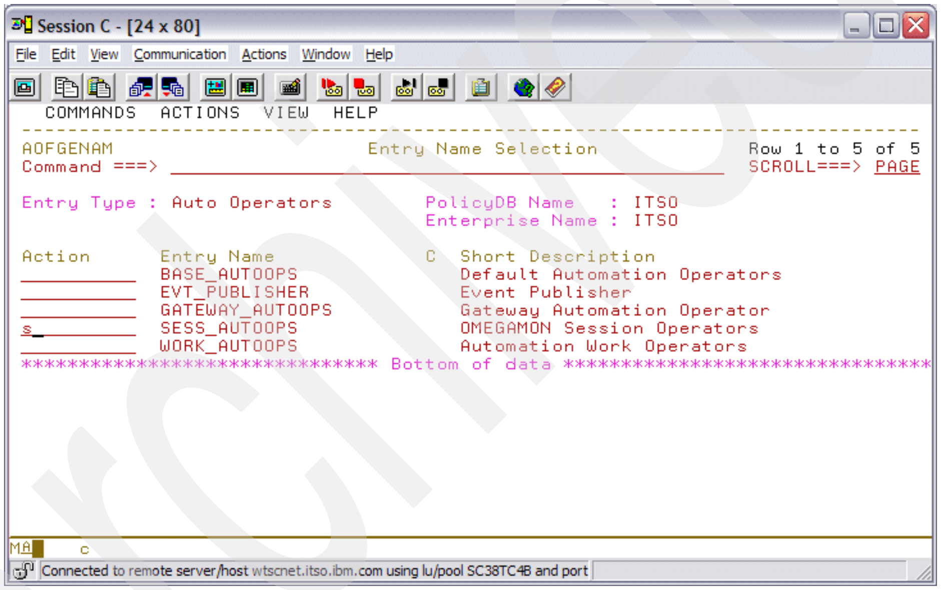
Figure 3-6 OMEGAMON session attributes screen

Our setup uses the VTAM applid of C66M2RC to connect to OMEGAMON for z/OS classic monitor.

Note: The user ID and password usage to access IBM Tivoli OMEGAMON will be discussed in 3.5, “Security consideration” on page 59.

3.2.2 Defining automated proxy operator

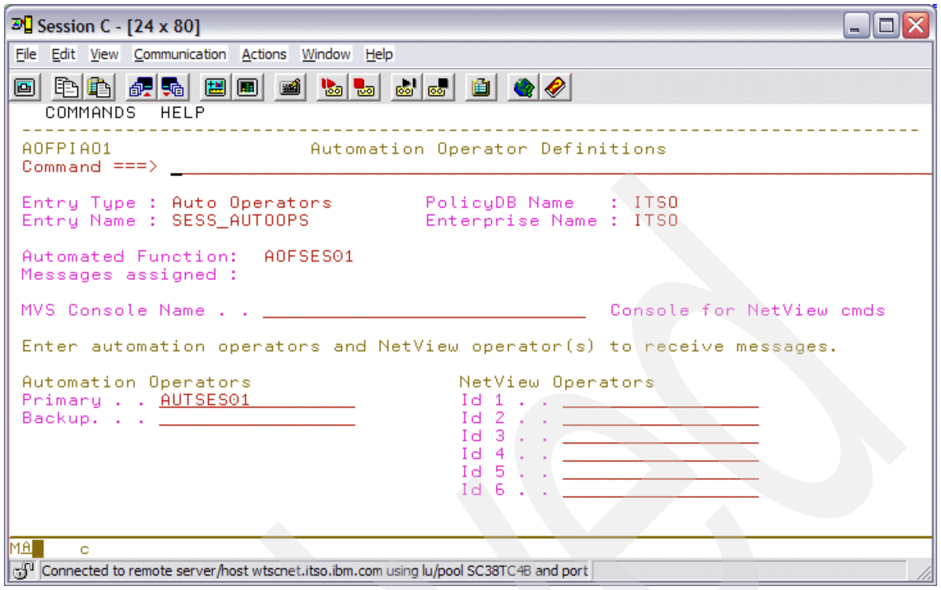
We show the definition of the automated operator for the proxy operator. You define the automated operator using option 37 from the Entry Type selection dialog in Figure 3-3 on page 37. The automated operator policy definition is shown in Figure 3-7.



```
  COMMANDS  ACTIONS  VIEW  HELP
 ------------------------------------------------------------------------------
 AOFGENAM                       Entry Name Selection               Row 1 to 5 of 5
 Command ===> ________________________________________________    SCROLL===> PAGE

 Entry Type : Auto Operators           PolicyDB Name    : ITSO
                                       Enterprise Name : ITSO

 Action       Entry Name               C  Short Description
 __________   BASE_AUTOOPS                Default Automation Operators
 __________   EVT_PUBLISHER               Event Publisher
 __________   GATEWAY_AUTOOPS             Gateway Automation Operator
 s_________   SESS_AUTOOPS                OMEGAMON Session Operators
 __________   WORK_AUTOOPS                Automation Work Operators
 ******************************* Bottom of data ********************************

MA      c
Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC38TC4B and port
```

Figure 3-7 Automated operator policy definition

When you select SESS_AUTOOPS definition, Figure 3-8 on page 41 show the content.



```
   COMMANDS   HELP
 ------------------------------------------------------------------------------
 AOFPIA01                  Automation Operator Definitions
 Command ===> _
 
 Entry Type : Auto Operators          PolicyDB Name    : ITSO
 Entry Name : SESS_AUTOOPS            Enterprise Name : ITSO
 
 Automated Function:   AOFSES01
 Messages assigned :
 
 MVS Console Name . . ____________________________ Console for NetView cmds
 
 Enter automation operators and NetView operator(s) to receive messages.
 
 Automation Operators                    NetView Operators
 Primary . . AUTSES01_________            Id 1 . . _________________
 Backup. . . _________________            Id 2 . . _________________
                                          Id 3 . . _________________
                                          Id 4 . . _________________
                                          Id 5 . . _________________
                                          Id 6 . . _________________
 
MA      c
```

Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC38TC4B and port

Figure 3-8   Automation operator definitions screen

3.2.3 Define the TAF logical units

The connection between System Automation for z/OS and IBM Tivoli OMEGAMON is a Terminal Access Facility (TAF) full screen session. TAF source LUs are defined in VTAMLST. Use model APPL statements similar to Example 3-1.

Example 3-1   TAF SRCLU definition

```
* ---------------------OMEGAMON TAF TERMINAL ------------------*
TFaa#*   APPL MODETAB=AMODETAB,EAS=9,                         X
               DLOGMOD=M2SDLCNQ
*              STATOPT='SCTUSER OMEGAMON'
```

In Example 3-1, the letter `aa` represent the last two characters of your NetView domain ID. This model APPL statements save you time since you do not need to define every SRCLU individually.

If you encounter problems connecting to the OMEGAMON monitor, you can try to issue a NetView BGNSESS command. For connecting to our SC66M2RC OMEGAMON session, issue the BGNSESS FLSCN,APPLID=C66M2RC command.

If the session fails when System Automation for z/OS initializes, but starts when you enter a BGNSESS command, then you have an error in your System Automation for z/OS policy definitions.

3.3 OMEGAMON related commands

In relation to the new integration with IBM Tivoli OMEGAMON, System Automation for z/OS V3.1 provides several new commands:

- The INGSESS command

  This command manages a session with OMEGAMON monitors. It also displays additional session attributes, such as logon data, timeout, and statistics.

- The INGOMX command

  This command issues OMEGAMON major, minor, and immediate commands. It also allows you to retrieve OMEGAMON exceptions.

- The INGMTRAP command

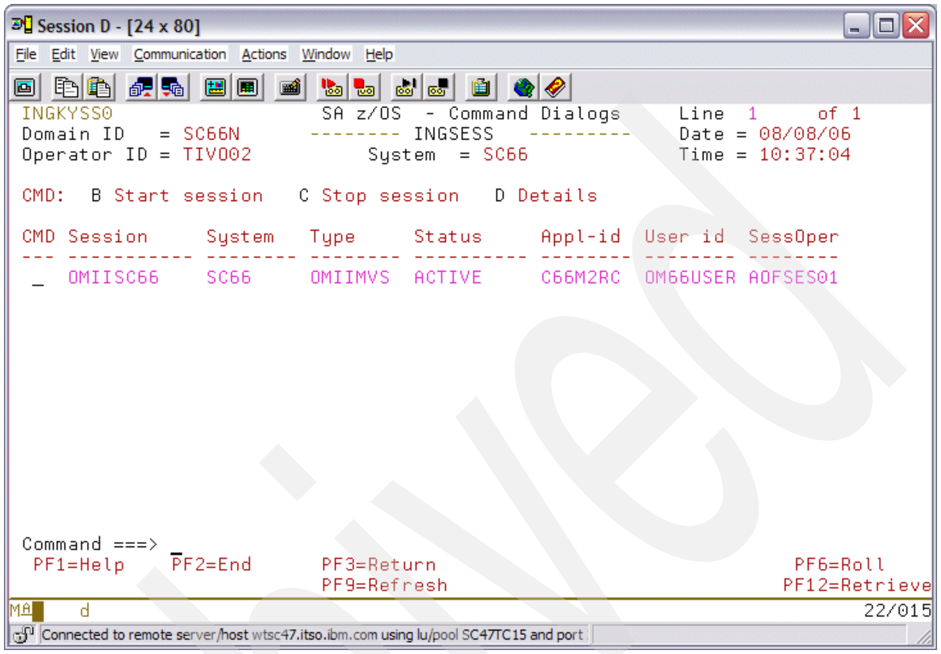
  This command provides a customized interface to process OMEGAMON exceptions.

For a detailed and complete description of System Automation for z/OS V3.1 commands, refer to *IBM Tivoli System Automation for z/OS Operator's Commands*, SC33-8265 and *IBM Tivoli System Automation for z/OS Programmer's Reference*, SC33-8266.

3.3.1 The INGSESS command

The INGSESS command displays OMEGAMON session definitions, the session status, and statistical information about the session.

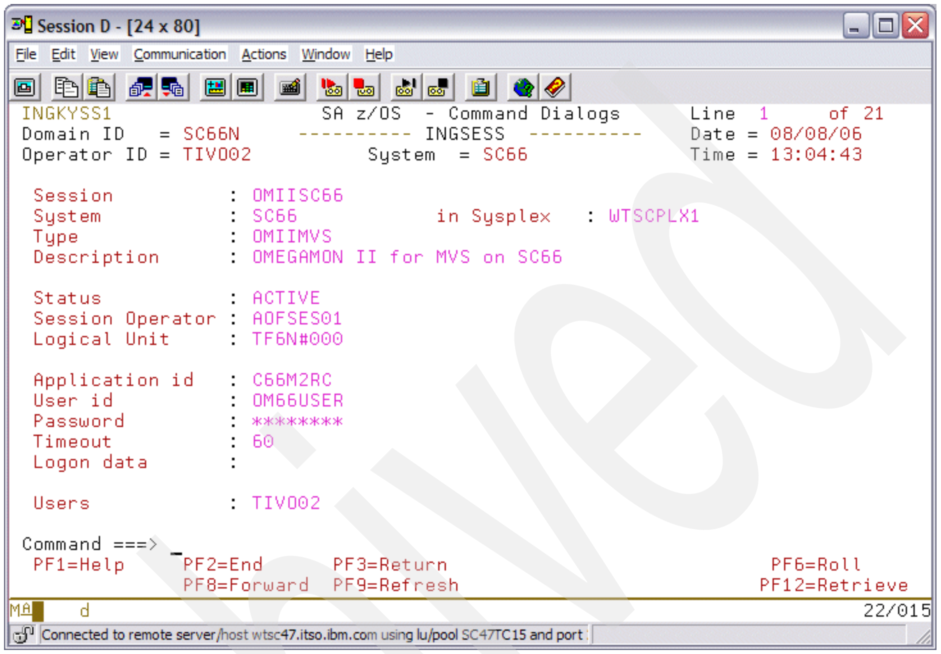
Issuing the command INGSESS REQ=DISPLAY brings up the session list shown in Figure 3-9.



```
INGKYSS0                      SA z/OS  - Command Dialogs      Line  1      of 1
Domain ID   = SC66N         -------- INGSESS  ---------     Date = 08/08/06
Operator ID = TIV002              System   = SC66             Time = 10:37:04

CMD:   B Start session    C Stop session    D Details

CMD  Session      System   Type     Status      Appl-id  User id  SessOper
---  -----------  -------- -------- ----------  -------- -------- --------
 _   OMIISC66     SC66     OMIIMVS  ACTIVE      C66M2RC  OM66USER AOFSES01










Command ===> _
 PF1=Help     PF2=End       PF3=Return                                PF6=Roll
                            PF9=Refresh                              PF12=Retrieve
MA      d                                                                 22/015
Connected to remote server/host wtsc47.itso.ibm.com using lu/pool SC47TC15 and port
```

Figure 3-9 OMEGAMON session list

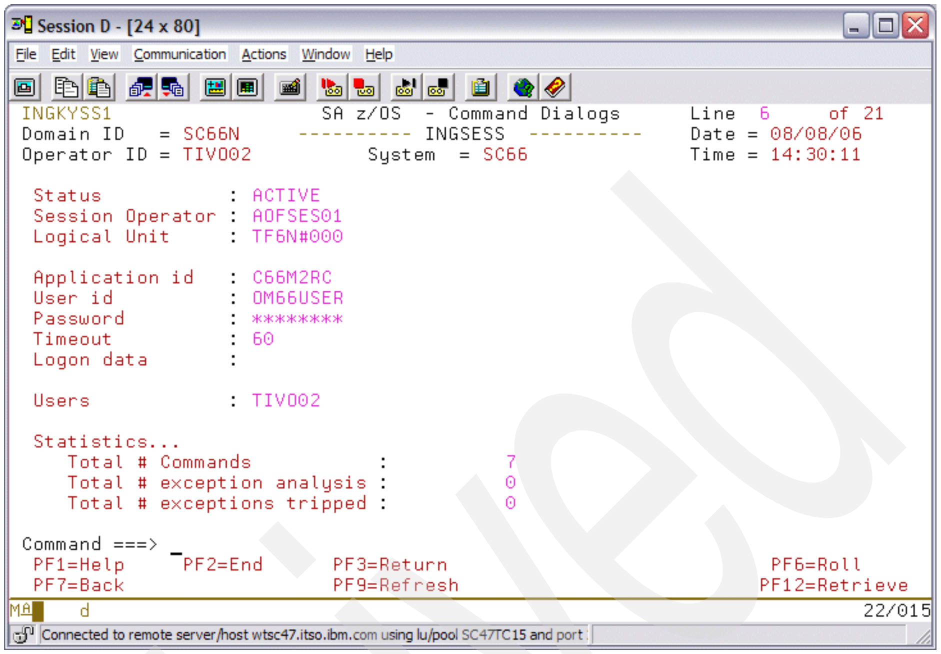
From the session list in Figure 3-9 on page 43, you can display the session detail using the line command D and pressing Enter. Figure 3-10 shows the detailed session information.



```
INGKYSS1                  SA z/OS  - Command Dialogs      Line  1      of  21
Domain ID    = SC66N    ---------- INGSESS ----------     Date = 08/08/06
Operator ID = TIVO02            System  = SC66            Time = 13:04:43

 Session            : OMIISC66
 System             : SC66            in Sysplex    : WTSCPLX1
 Type               : OMIIMVS
 Description        : OMEGAMON II for MVS on SC66

 Status             : ACTIVE
 Session Operator   : AOFSES01
 Logical Unit       : TF6N#000

 Application id     : C66M2RC
 User id            : OM66USER
 Password           : ********
 Timeout            : 60
 Logon data         :

 Users              : TIVO02

Command ===> _
 PF1=Help      PF2=End      PF3=Return                            PF6=Roll
               PF8=Forward  PF9=Refresh                          PF12=Retrieve
MA      d                                                              22/015
Connected to remote server/host wtsc47.itso.ibm.com using lu/pool SC47TC15 and port
```

Figure 3-10   Detailed session information

In Figure 3-10, OMEGAMON security is not enabled; therefore, the User ID is not defined. You can see more session attributes, such as the session statistics, using the PF8 key. Figure 3-11 on page 45 shows the additional information.

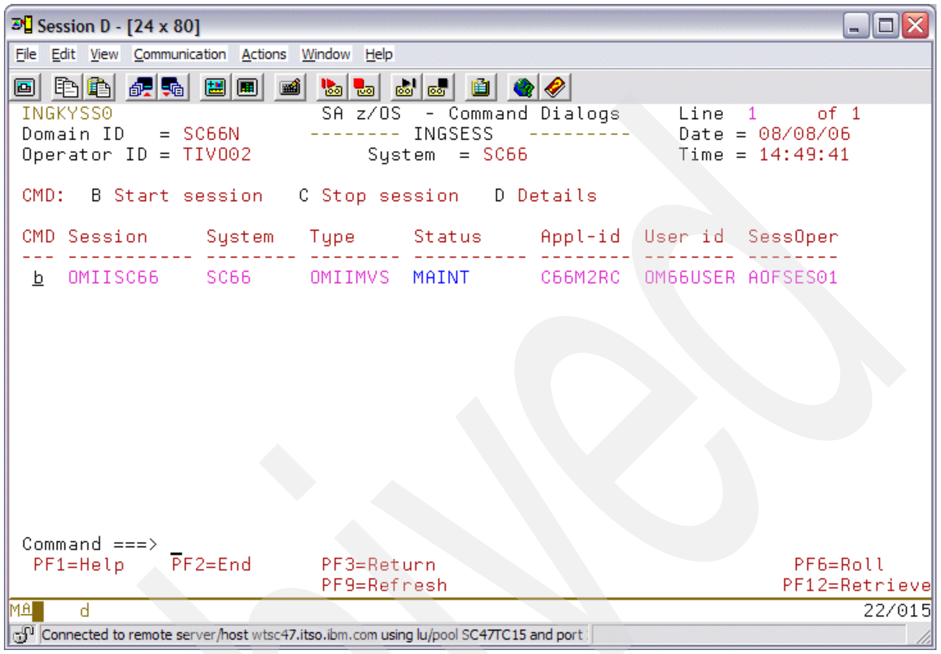


```
INGKYSS1                  SA z/OS  - Command Dialogs       Line  6      of 21
Domain ID    = SC66N     ---------- INGSESS  ----------    Date = 08/08/06
Operator ID = TIV002            System   = SC66             Time = 14:30:11

 Status            : ACTIVE
 Session Operator  : AOFSES01
 Logical Unit      : TF6N#000

 Application id    : C66M2RC
 User id           : OM66USER
 Password          : ********
 Timeout           : 60
 Logon data        :

 Users             : TIV002

 Statistics...
    Total # Commands             :          7
    Total # exception analysis :          0
    Total # exceptions tripped :          0

Command ===> _
 PF1=Help     PF2=End       PF3=Return                       PF6=Roll
 PF7=Back                   PF9=Refresh                     PF12=Retrieve
MA      d                                                               22/015
Connected to remote server/host wtsc47.itso.ibm.com using lu/pool SC47TC15 and port
```

Figure 3-11 Session statistics screen

The total number of commands indicated are the commands that are issued using the INGOMX command interface.

You can stop a session using the C line command. The session status goes into maintenance, as shown in Figure 3-12.



```
INGKYSS0                  SA z/OS  - Command Dialogs      Line  1     of  1
Domain ID   = SC66N      -------- INGSESS  ---------      Date = 08/08/06
Operator ID = TIVO02          System   = SC66             Time = 14:49:41

CMD:   B Start session   C Stop session   D Details

CMD Session      System   Type     Status      Appl-id  User id  SessOper
--- ------------ -------- -------- ----------- -------- -------- --------
 b  OMIISC66     SC66     OMIIMVS  MAINT       C66M2RC  OM66USER AOFSES01








Command ===> _
 PF1=Help     PF2=End       PF3=Return                            PF6=Roll
                            PF9=Refresh                          PF12=Retrieve
MA     d                                                               22/015
Connected to remote server/host wtsc47.itso.ibm.com using lu/pool SC47TC15 and port
```

Figure 3-12 Session stopped

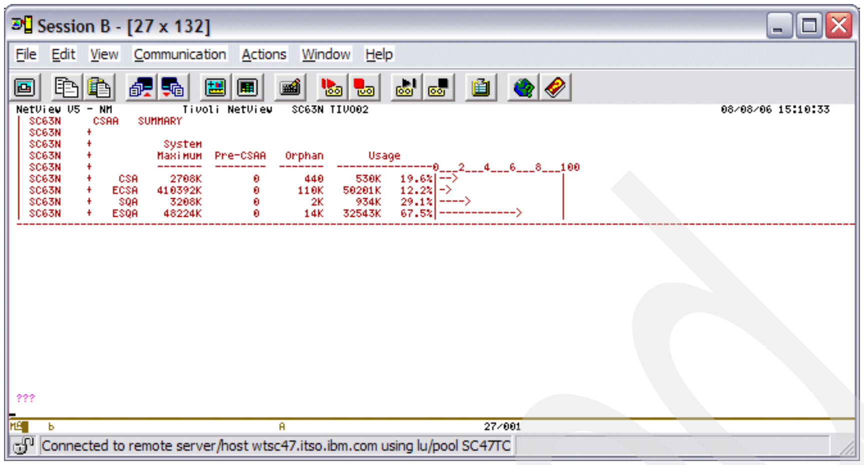
From a stopped session, select line command B to restart the session.

3.3.2 The INGOMX command

The INGOMX command is an interface that provides interaction capabilities with IBM Tivoli OMEGAMON. It allows a program or command list to invoke OMEGAMON exception analysis in order to trap one or more exceptions of interest or to issue one or more OMEGAMON commands. The response generated by OMEGAMON on behalf of a request is written to the console but not exposed to automation.

In this section, we demonstrate issuing the OMEGAMON commands from NetView.

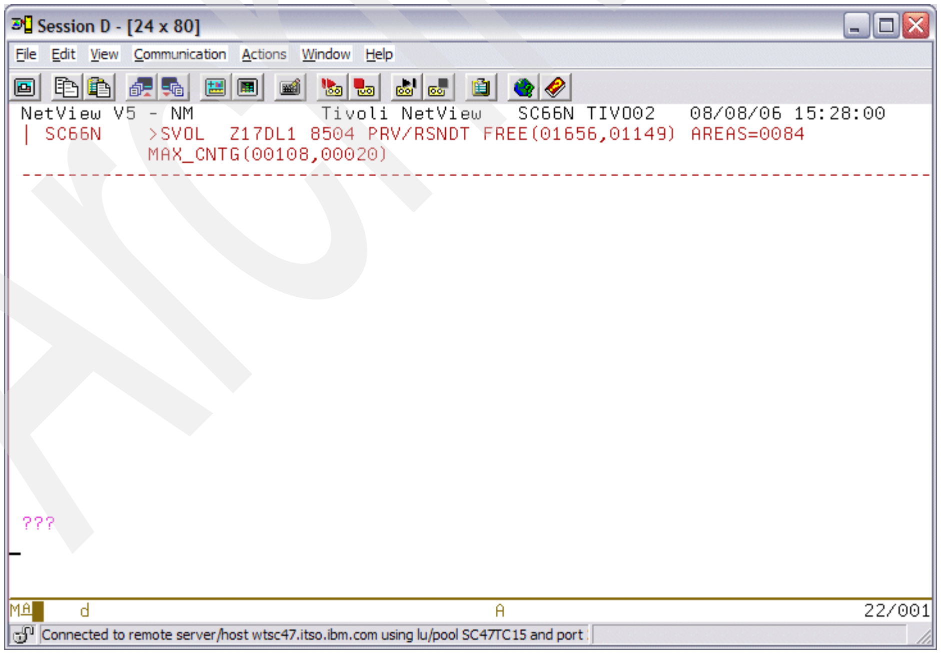
- Requesting OMEGAMON for z/OS common storage usage is achieved using the OMEGAMON command CSAA. The NetView command that we invoke is INGOMX EXECUTE,NAME=OMIISC66,CMD=CSAA. Figure 3-13 on page 47 shows the resulting display.



```
NetView V5 - NM           Tivoli NetView   SC63N TIV002                                                   08/08/06 15:10:33
| SC63N     CSAA   SUMMARY
| SC63N    +
| SC63N    +                System
| SC63N    +               Maximum  Pre-CSAA   Orphan       Usage
| SC63N    +               -------  --------  -------  ---------------0___2___4___6___8___100
| SC63N    +   CSA       2708K         0       440     530K   19.6%|-->                    |
| SC63N    +  ECSA     410392K         0      110K   50201K   12.2%|->                     |
| SC63N    +   SQA       3208K         0        2K     934K   29.1%|---->                  |
| SC63N    +  ESQA      48224K         0       14K   32543K   67.5%|------------->         |
---------------------------------------------------------------------------------------------------------------------------------
???
MA   b                                    A                        27/001
Connected to remote server/host wtsc47.itso.ibm.com using lu/pool SC47TC
```

Figure 3-13 CSAA command response

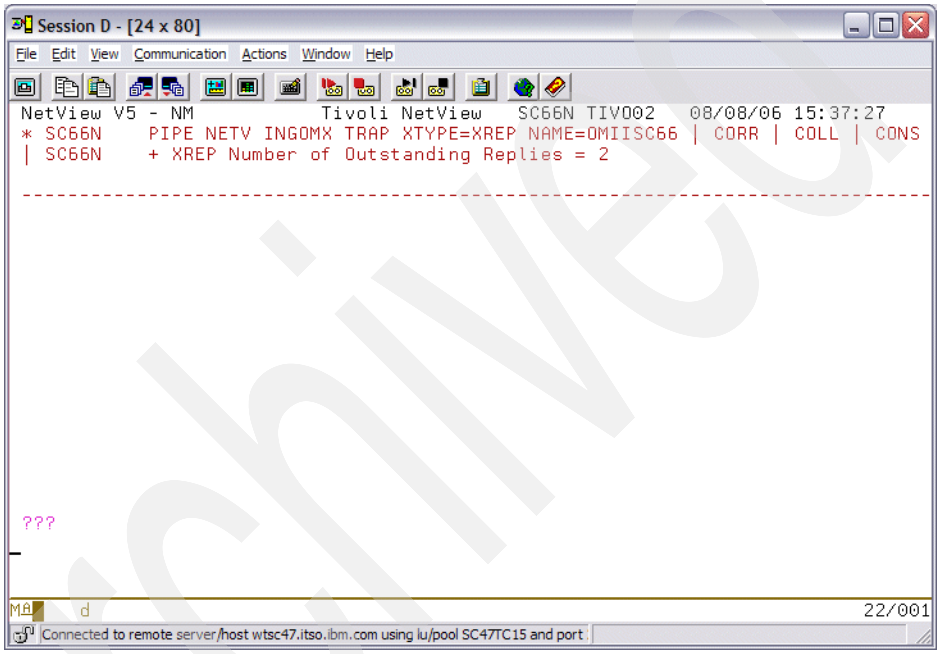
- Some OMEGAMON commands require additional parameters. This is achieved using the PARM keyword of the INGOMX command. An example is the SVOL command that retrieves the DASD volume statistic. The command that we issue is INGOMX EXECUTE,NAME=OMIISC66,CMD= SVOL,PARM=Z17DL1. Figure 3-14 shows the command result.



```
NetView V5 - NM          Tivoli NetView    SC66N TIV002    08/08/06 15:28:00
| SC66N     >SVOL  Z17DL1 8504 PRV/RSNDT FREE(01656,01149) AREAS=0084
           MAX_CNTG(00108,00020)
 ------------------------------------------------------------------------------
???
MA    d                              A                                  22/001
Connected to remote server/host wtsc47.itso.ibm.com using lu/pool SC47TC15 and port
```

Figure 3-14 SVOL command response

- The INGOMX TRAP command allows you to retrieve exceptions, such as the OMEGAMON for z/OS exception for outstanding replies, XREP. The exception retrieval is typically invoked under a NetView PIPE command to be able to analyze and process the returned result from OMEGAMON, as the exception are returned asynchronously, INGOMX does not issue any confirmation messages, and the output is only written to the console when INGOMX returned with code 0. The command that we use is PIPE NETV INGOMX TRAP,XTYPE=XREP,NAME=OMIISC66 | CORR | COLL | CONSOLE. Figure 3-15 shows the command result.



```
NetView V5 - NM             Tivoli NetView    SC66N TIVO02   08/08/06 15:37:27
* SC66N     PIPE NETV INGOMX TRAP XTYPE=XREP NAME=OMIISC66 | CORR | COLL | CONS
| SC66N     + XREP Number of Outstanding Replies = 2

 -------------------------------------------------------------------------------








???
_
MA    d                                                                  22/001
Connected to remote server/host wtsc47.itso.ibm.com using lu/pool SC47TC15 and port
```

Figure 3-15 TRAP command response

3.3.3 Monitor command INGMTRAP

The INGMTRAP command provides a customized interface to INGOMX. It provides filtering capabilities for exceptions of interest as reported by OMEGAMON exception analysis and triggering of automation on behalf of such exceptions. The processing of INGMTRAP is shown in Figure 3-16 on page 49.

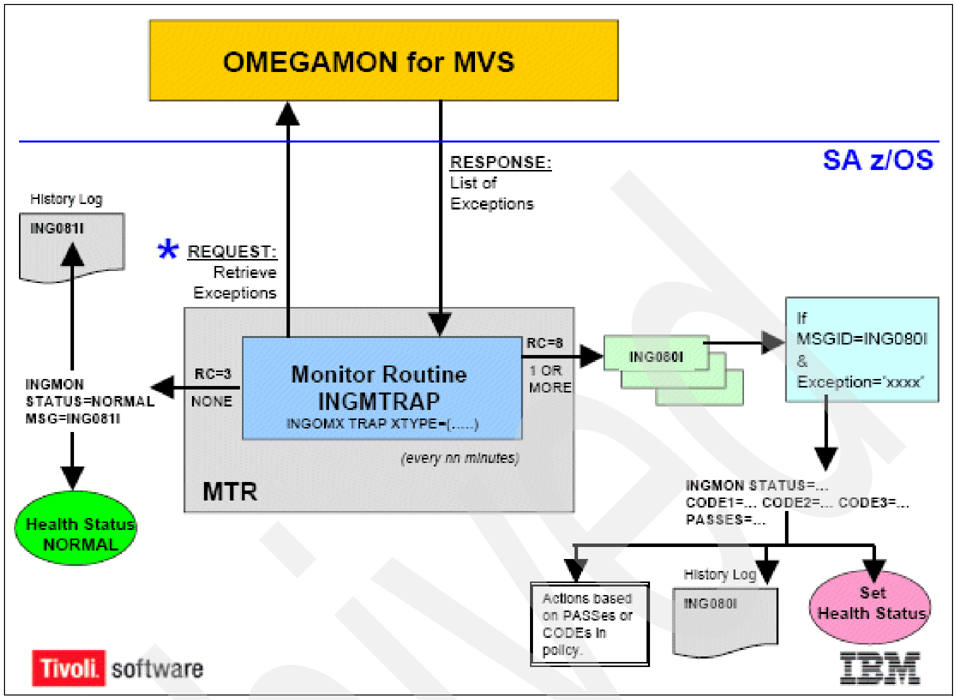

OMEGAMON for MVS
SA z/OS
RESPONSE: List of Exceptions
REQUEST: Retrieve Exceptions
History Log
ING081I
INGMON STATUS=NORMAL MSG=ING081I
Health Status NORMAL
RC=3 NONE
Monitor Routine INGMTRAP
INGOMX TRAP XTYPE=(.....)
(every nn minutes)
MTR
RC=8 1 OR MORE
ING080I
If MSGID=ING080I & Exception='xxxx'
INGMON STATUS=... CODE1=... CODE2=... CODE3=... PASSES=...
Actions based on PASSes or CODEs in policy.
History Log
ING080I
Set Health Status
Tivoli. software
IBM

Figure 3-16 Exceptions and Health Status

INGMTRAP issues the INGOMX TRAP command regularly:

- For each exception that matches the XTYPE filter that is provided by the caller, INGMTRAP issues message ING080I, which is exposed to NetView automation. An example of ING080I is:

```
ING080I SC66XREP/MTR/SC66 OMIISC66 OMIIMVS XREP Number of
Outstanding Replies = 2
```

- If no exception matches the XTYPE filter that is provided by the caller, INGMTRAP creates a ING081I message that is not exposed to NetView but written to the Monitor Resource history log to document that no exception has been found. An example of ING081I is:

```
ING081I SC66XREP/MTR/SC66 OMIISC66 OMIIMVS NO EXCEPTION FOUND
```

INGMTRAP can only be used as a monitor command. This means that it has to be specified directly as a monitor command in the definition of a Monitor Resource, or it has to be called on behalf of such a monitor command.

An example of the monitor resource command for trapping outstanding operator replies (XREP exception) that are reported by IBM Tivoli OMEGAMON for z/OS session OMIISC66 is INGMTRAP NAME=OMIISC66,XTYPE=(XREP).

Be careful when specifying a list of exceptions, each exception may cause an ING080I message to be issued. Because each occurrence of an ING080I message will trigger health state processing of the Monitor Resource, make sure you understand the impact that this may have on the Monitor Resource's final health state.

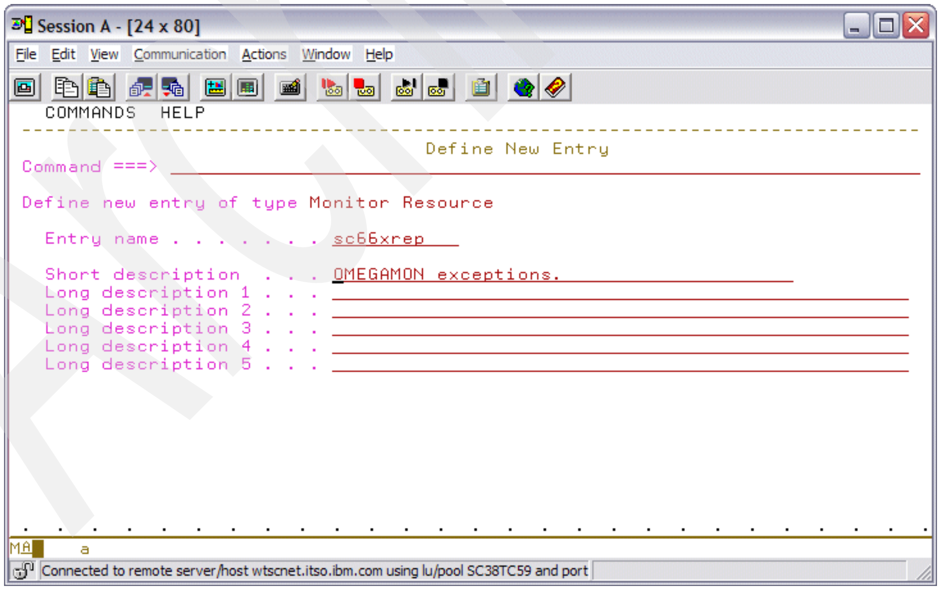
For more details about INGMTRAP, refer to *IBM Tivoli System Automation for z/OS Programmer's Reference*, SC33-8266. For more details about defining Monitor Resources, refer to *IBM Tivoli System Automation for z/OS Defining Automation Policy*, SC33-8262.

3.4 Defining a monitored resource

This section discusses the definition and use of the IBM Tivoli OMEGAMON interface for monitoring the health status of a resource.

3.4.1 Monitored resource policy definitions

Define a monitored resource using option 11 from the entry type selection dialog in Figure 3-3 on page 37. Create a new monitor resource. In Figure 3-17, we define a new resource called SC66XREP.


Session A - [24 x 80]
File Edit View Communication Actions Window Help

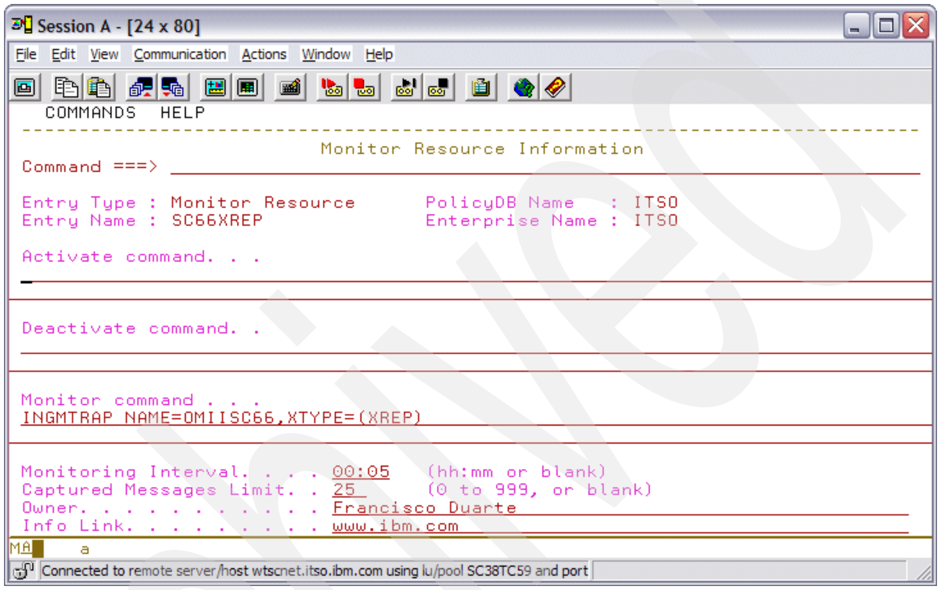
```
  COMMANDS  HELP
------------------------------------------------------------------------------
                                   Define New Entry
Command ===> 

Define new entry of type Monitor Resource

  Entry name . . . . . . . . sc66xrep

  Short description  . . . . OMEGAMON exceptions.
  Long description 1 . . . .
  Long description 2 . . . .
  Long description 3 . . . .
  Long description 4 . . . .
  Long description 5 . . . .
```

MA a
Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC38TC59 and port

Figure 3-17 Define new monitored resource entry

This resource is the resource where we will use INGMTRAP as the monitor routine to identify the XREP exceptions to retrieve. If you code your own monitor routine, you can use INGOMX to retrieve OMEGAMON exceptions.

Press PF3 to exit and save. In the policy selection menu, select MONITOR INFO policy. Figure 3-18 shows the Monitor Resource Information definition. Here we define the monitor command for the resource.



```
  COMMANDS  HELP
------------------------------------------------------------------------------
                          Monitor Resource Information
Command ===> 

Entry Type : Monitor Resource        PolicyDB Name    : ITSO
Entry Name : SC66XREP                Enterprise Name  : ITSO

Activate command. . .


Deactivate command. .


Monitor command . . .
INGMTRAP NAME=OMIISC66,XTYPE=(XREP)

Monitoring Interval. . . . 00:05    (hh:mm or blank)
Captured Messages Limit. . 25       (0 to 999, or blank)
Owner. . . . . . . . . . . Francisco Duarte
Info Link. . . . . . . . . www.ibm.com
MA      a
```

Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC38TC59 and port

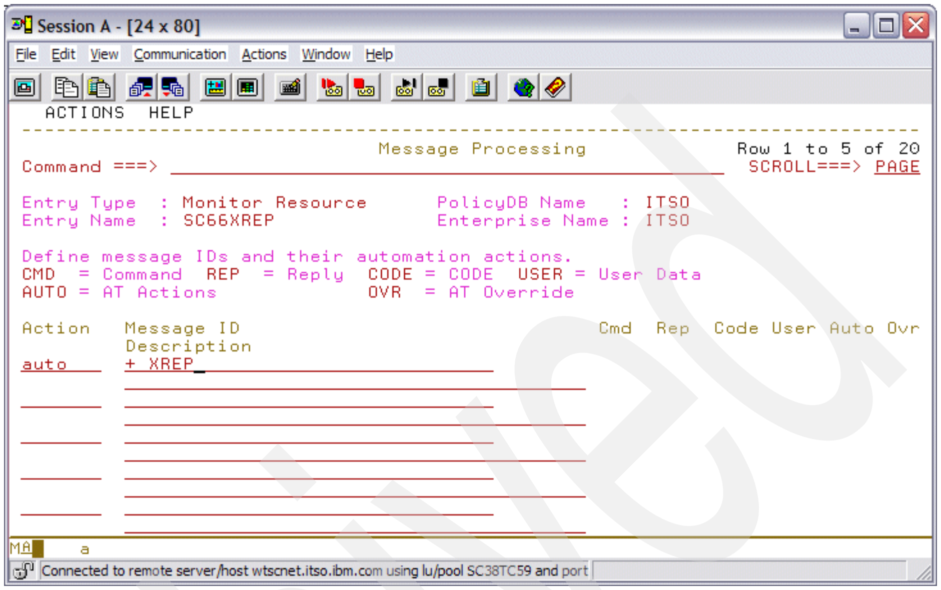
Figure 3-18 Monitor Resource information screen

As shown in Figure 3-18, INGMTRAP is defined to retrieve OMEGAMON XREP exceptions. Press PF3 to exit and save this policy.

In this MTR policy, you can add Messages or User Data policy to support this OMEGAMON exception:

- Map exceptions to Health Status
- Issue commands using PASSes or CODEs
- Issue commands based on health state processing
- Define your own set of automation routines to be called
- Disable exception handling while recovery is being done

Select MESSAGES/USER DATA policy defination. Figure 3-19 shows the message processing screen.



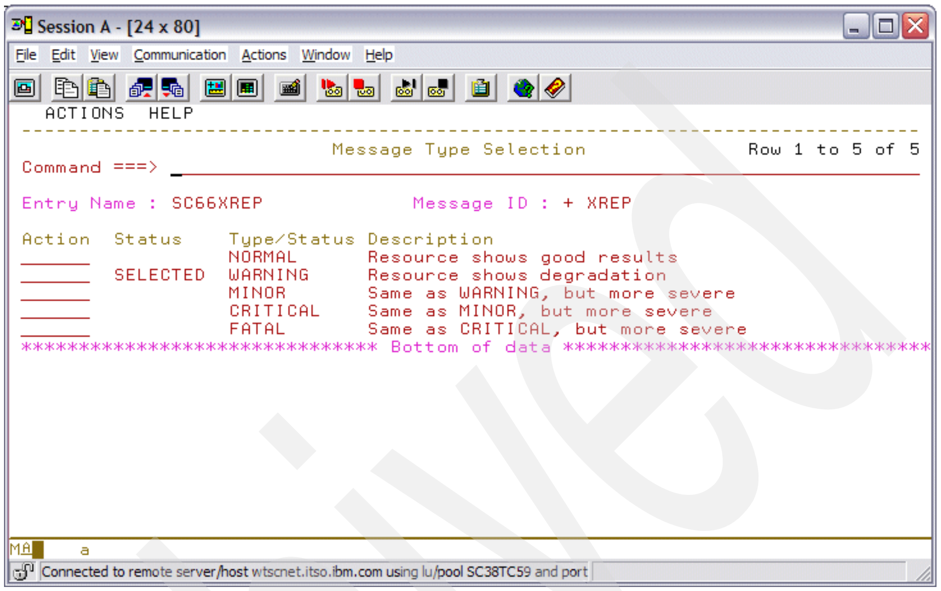
```
  ACTIONS  HELP
 ------------------------------------------------------------------------------
                               Message Processing                Row 1 to 5 of 20
 Command ===> ________________________________________________ SCROLL===> PAGE

 Entry Type  : Monitor Resource       PolicyDB Name   : ITSO
 Entry Name  : SC66XREP               Enterprise Name : ITSO

 Define message IDs and their automation actions.
 CMD  = Command  REP  = Reply  CODE = CODE  USER = User Data
 AUTO = AT Actions             OVR  = AT Override

 Action    Message ID                                Cmd  Rep  Code User Auto Ovr
           Description
 auto      + XREP_
```

```
MA      a
Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC38TC59 and port
```

Figure 3-19 Message processing - defining exception

Use the Message ID field to define a six character message ID. All exceptions must be identified with a + as the first character and must followed by a blank. The + is used by System Automation for z/OS to identify the message as an exception ID instead of an actual message ID. For the XREP exception, we use the message ID of `+ XREP`.

An action of AUTO displays (Figure 3-20). This shows the health status selection dialog. In this example, every time an XREP exception is detected the Health Status is set to Warning.

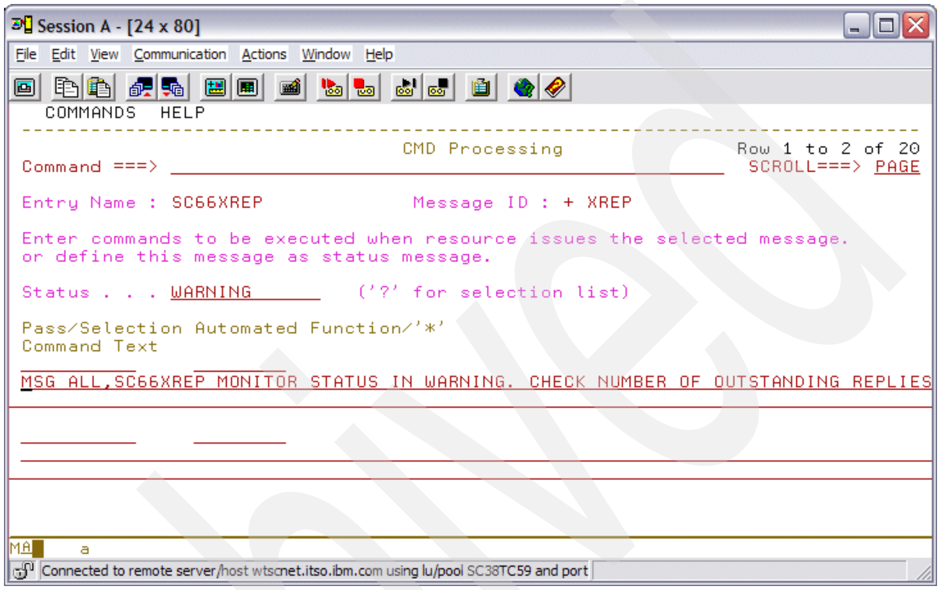


```
  ACTIONS   HELP
 ------------------------------------------------------------------------------
                          Message Type Selection                Row 1 to 5 of 5
Command ===> _

Entry Name : SC66XREP               Message ID : + XREP

Action   Status     Type/Status Description
                    NORMAL      Resource shows good results
______   SELECTED   WARNING     Resource shows degradation
______              MINOR       Same as WARNING, but more severe
______              CRITICAL    Same as MINOR, but more severe
______              FATAL       Same as CRITICAL, but more severe
****************************** Bottom of data *******************************
```

Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC38TC59 and port

Figure 3-20 Message type selection screen

Figure 3-20 on page 53 shows that the selected status is WARNING for the exception +XREP. Press PF3 to exit and save this definition. In the Message Type definition in Figure 3-19 on page 52, use the line command CMD to get to the command definition screen. Figure 3-21 shows this command definition screen.



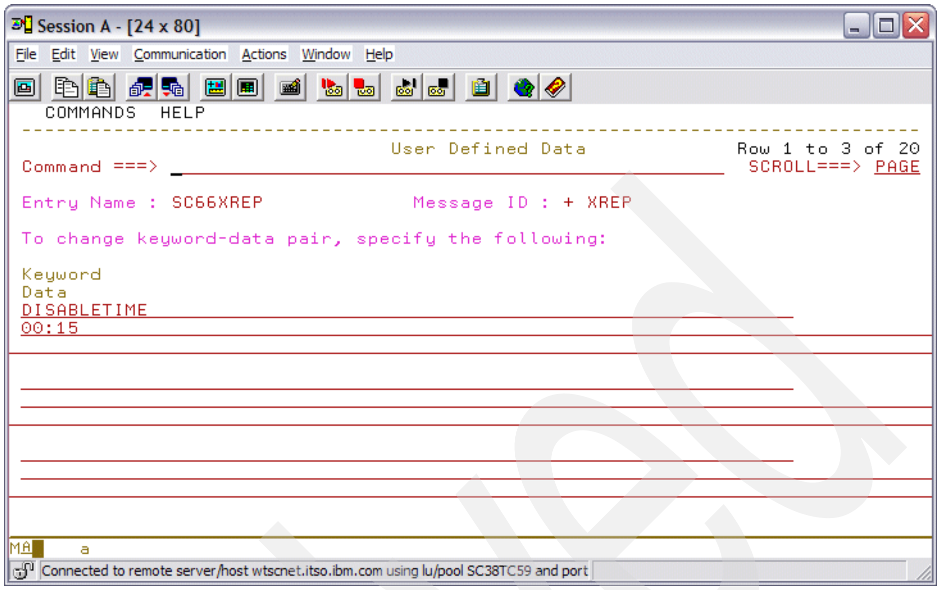
```
  COMMANDS  HELP
------------------------------------------------------------------------------
                                CMD Processing              Row 1 to 2 of 20
Command ===> ________________________________________________ SCROLL===> PAGE

Entry Name : SC66XREP            Message ID : + XREP

Enter commands to be executed when resource issues the selected message.
or define this message as status message.

Status . . . WARNING_____    ('?' for selection list)

Pass/Selection Automated Function/'*'
Command Text

MSG ALL,SC66XREP MONITOR STATUS IN WARNING. CHECK NUMBER OF OUTSTANDING REPLIES
```

Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC38TC59 and port

Figure 3-21 Message processing - CMD screen

In Figure 3-21, the XREP exception from OMEGAMON will trigger a MSG ALL command and set the health status to WARNING. Press PF3 to exit and save this definition back to the message type selection dialog.

Using the USER line command, define an additional user processing keyword. Figure 3-22 on page 55 shows the user processing definition.



```
   COMMANDS   HELP
 ------------------------------------------------------------------------------
                                 User Defined Data                Row 1 to 3 of 20
 Command ===> _                                                   SCROLL===> PAGE

 Entry Name : SC66XREP                 Message ID : + XREP

 To change keyword-data pair, specify the following:

 Keyword
 Data
 DISABLETIME
 00:15
```

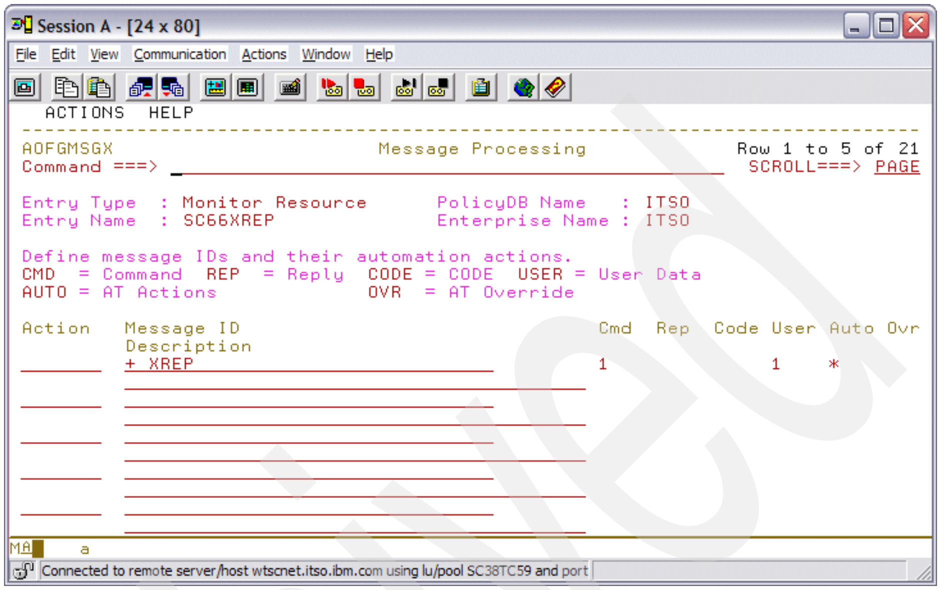
Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC38TC59 and port

Figure 3-22   *Message processing - user defined data screen*

The keyword `DISABLETIME` is used to define a period of time to suspend automation from running further actions.

Warning: If you define DISABLETIME for an OMEGAMON exception that generates more than one ING080I message, the first ING080I message will be processed and automation for the subsequent messages will be suspended due to the DISABLETIME definition.

Press PF3 to exit and save until you return to the AOFGMSGX screen shown in Figure 3-23.



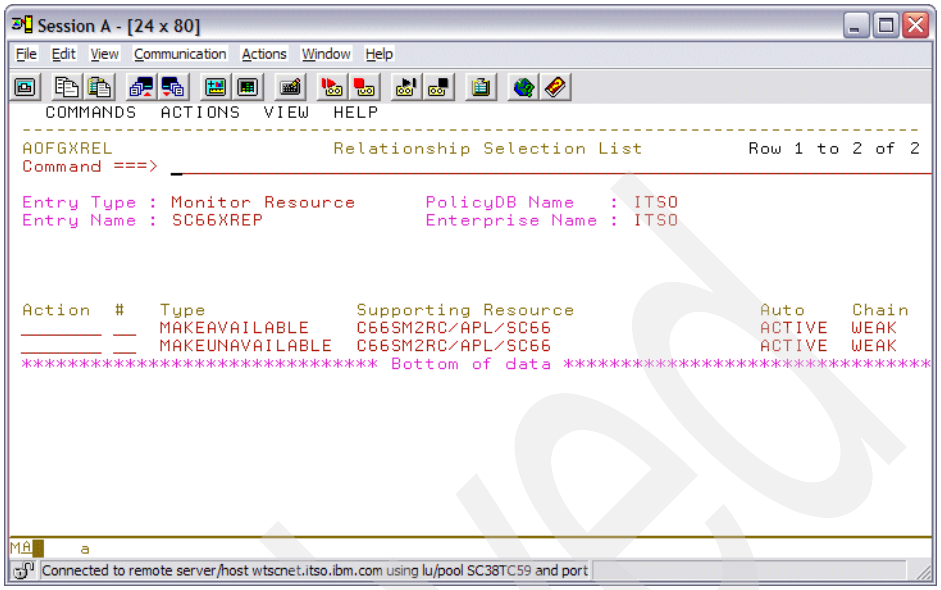
```
  ACTIONS  HELP
------------------------------------------------------------------------------
AOFGMSGX                      Message Processing              Row 1 to 5 of 21
Command ===> _______________________________________________ SCROLL===> PAGE

Entry Type  : Monitor Resource      PolicyDB Name   : ITSO
Entry Name  : SC66XREP              Enterprise Name : ITSO

Define message IDs and their automation actions.
CMD  = Command   REP  = Reply   CODE = CODE  USER = User Data
AUTO = AT Actions               OVR  = AT Override

Action    Message ID                                  Cmd  Rep  Code User Auto Ovr
          Description
_______   + XREP                                       1              1    *
```

MA a
Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC38TC59 and port

Figure 3-23 Message processing screen

Figure 3-23 indicates that the XREP exception has a command, user defined keyword, and automation defined. You can press PF3 until the policy selection dialog appears. Here, you can add a relationship. A relationship is needed to ensure that the monitor is started and stopped when the OMEGAMON monitor session starts and stops.

Figure 3-24 on page 57 shows the RELATIONSHIP policy where we link SC66XREP to the C66M2RC started task.



```
   COMMANDS   ACTIONS   VIEW   HELP
 ------------------------------------------------------------------------------
 AOFGXREL                  Relationship Selection List            Row 1 to 2 of 2
 Command ===> _

 Entry Type : Monitor Resource        PolicyDB Name    : ITSO
 Entry Name : SC66XREP                Enterprise Name  : ITSO



 Action   #   Type              Supporting Resource              Auto    Chain
 _______ __   MAKEAVAILABLE     C66SM2RC/APL/SC66                ACTIVE  WEAK
 _______ __   MAKEUNAVAILABLE   C66SM2RC/APL/SC66                ACTIVE  WEAK
 ******************************* Bottom of data ********************************

MA     a
```

Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC38TC59 and port

Figure 3-24 Relationship selection list screen

As shown in Figure 3-24, the relationship that we define makes SC66XREP available and unavailable according to the application status of C66M2RC in SC66.

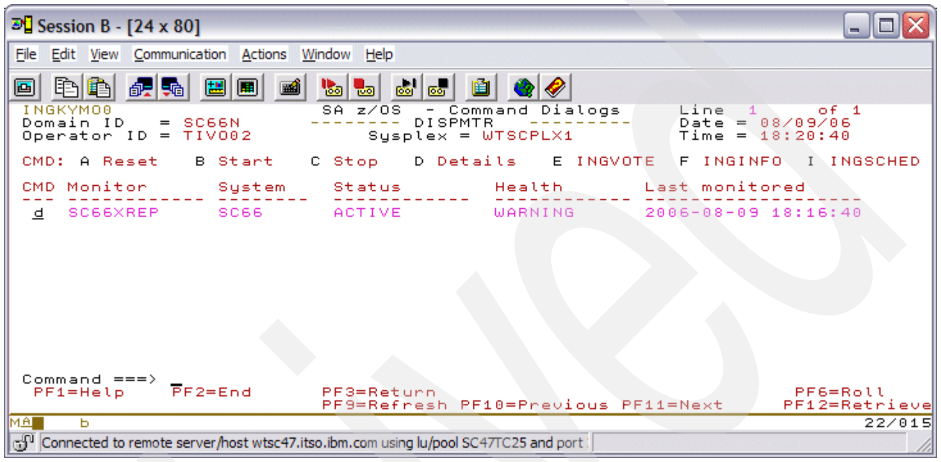
3.4.2 Reset recovery

User data in the MESSAGES/USER DATA policy (see Figure 3-22 on page 55) can be used to disable additional recovery processing while another recovery is already in progress. In combination with the predefined keyword DISABLETIME, the recovery disable time can be specified in the formats hh:mm:ss, mm:ss, :ss, or mm. While recovery is disabled, no commands are processed on behalf of this Monitor Resource for messages and exceptions that are specified in the MESSAGES/USER DATA policy.

Recovery is automatically enabled after the recovery disable time has expired. Recovery can also be enabled prematurely by calling the generic routine INGMON with the option CLEARING=YES, for example, INGMON SC66XREP MSGTYPE=XREP CLEARING=YES.

3.4.3 DISPMTR details example

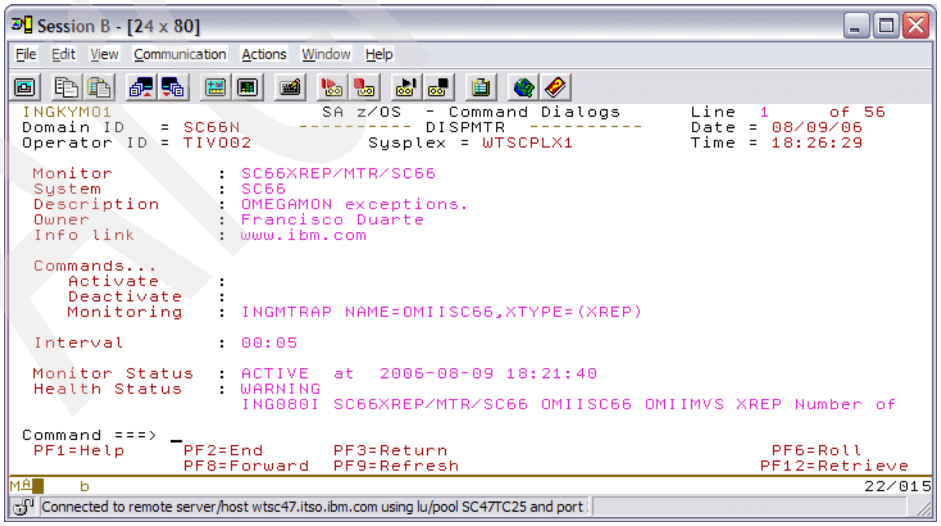
Monitor resources are managed with the DISPMTR command. DISPMTR can be invoked directly from a NetView command line, but will most likely be called by INGLIST because of a bad health state for a resource. Figure 3-25 shows the result of the DISPMTR command. It lists all MTR resources.



```
INGKYM00                  SA z/OS   - Command Dialogs       Line  1     of 1
Domain ID    = SC66N     -------- DISPMTR  ---------       Date = 08/09/06
Operator ID  = TIVO02         Sysplex = WTSCPLX1           Time = 18:20:40

CMD: A Reset    B Start   C Stop   D Details   E INGVOTE  F INGINFO  I INGSCHED

CMD Monitor          System   Status        Health        Last monitored
--- ------------     -------- ------------  ------------  -------------------
 d  SC66XREP         SC66     ACTIVE        WARNING       2006-08-09 18:16:40










Command ===> _
 PF1=Help     PF2=End       PF3=Return                          PF6=Roll
                            PF9=Refresh PF10=Previous PF11=Next PF12=Retrieve
MA      b                                                               22/015
Connected to remote server/host wtsc47.itso.ibm.com using lu/pool SC47TC25 and port
```

Figure 3-25   DISPMTR resource list screen

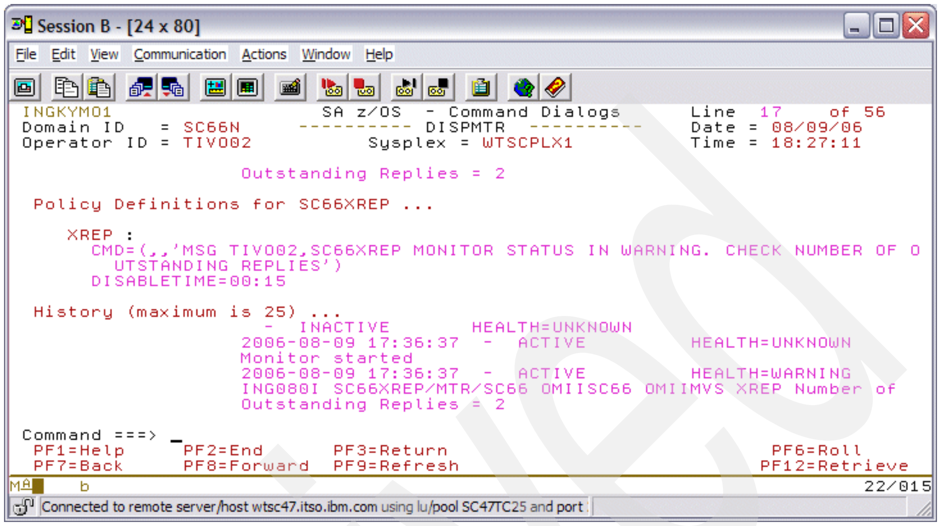
Suppose you want to see the details, including the last twenty messages captured. Use the line command D (Details) and you will get a screen similar to Figure 3-26.



```
INGKYM01                  SA z/OS   - Command Dialogs        Line  1     of 56
Domain ID    = SC66N    ---------- DISPMTR  ----------       Date = 08/09/06
Operator ID  = TIVO02         Sysplex = WTSCPLX1             Time = 18:26:29

 Monitor           : SC66XREP/MTR/SC66
 System            : SC66
 Description       : OMEGAMON exceptions.
 Owner             : Francisco Duarte
 Info link         : www.ibm.com

 Commands...
    Activate       :
    Deactivate     :
    Monitoring     : INGMTRAP NAME=OMIISC66,XTYPE=(XREP)

 Interval          : 00:05

 Monitor Status    : ACTIVE  at  2006-08-09 18:21:40
 Health Status     : WARNING
                     ING080I SC66XREP/MTR/SC66 OMIISC66 OMIMVS XREP Number of

Command ===> _
 PF1=Help     PF2=End      PF3=Return                            PF6=Roll
              PF8=Forward  PF9=Refresh                          PF12=Retrieve
MA      b                                                               22/015
Connected to remote server/host wtsc47.itso.ibm.com using lu/pool SC47TC25 and port
```

Figure 3-26   DISPMTR detail screen

Pressing PF8 displays the messages that were saved for this resource, based on the captured messages limit defined in the MTR policy. See Figure 3-27.



```
INGKYM01                    SA z/OS  - Command Dialogs        Line  17    of 56
Domain ID    = SC66N      ---------- DISPMTR  ----------      Date = 08/09/06
Operator ID  = TIVO02            Sysplex = WTSCPLX1           Time = 18:27:11

                    Outstanding Replies = 2

 Policy Definitions for SC66XREP ...

    XREP :
      CMD=(,,'MSG TIVO02,SC66XREP MONITOR STATUS IN WARNING. CHECK NUMBER OF O
        UTSTANDING REPLIES')
      DISABLETIME=00:15

 History (maximum is 25) ...
                      -  INACTIVE         HEALTH=UNKNOWN
                    2006-08-09 17:36:37  -  ACTIVE          HEALTH=UNKNOWN
                    Monitor started
                    2006-08-09 17:36:37  -  ACTIVE          HEALTH=WARNING
                    ING080I SC66XREP/MTR/SC66 OMIISC66 OMIIMVS XREP Number of
                    Outstanding Replies = 2

Command ===> _
 PF1=Help     PF2=End       PF3=Return                             PF6=Roll
 PF7=Back     PF8=Forward   PF9=Refresh                           PF12=Retrieve
MA     b                                                                22/015
Connected to remote server/host wtsc47.itso.ibm.com using lu/pool SC47TC25 and port
```

Figure 3-27 DISPMTR message history screen

3.5 Security consideration

Regarding the integration between System Automation for z/OS and IBM Tivoli OMEGAMON, security is a major aspect. The advanced ability of IBM Tivoli OMEGAMON to get into the system must be properly protected.

3.5.1 OMEGAMON security

Your installation may require security, such as who can log on to the OMEGAMON and which OMEGAMON commands they can issue. Passwords for the sessions between the System Automation for z/OS automation agent and each OMEGAMON monitor can also be secured. The security can be defined within OMEGAMON or NetView or both.

OMEGAMON supports security with Security Access Facility (SAF) products, such as RACF or an internal security table within OMEGAMON. Items that can be secured are commands, and command parameters based on user ID and password authentication.

For the System Automation for z/OS interface, we recommend defining a user ID to OMEGAMON, which can be defined internally in the OMEGAMON security table or externally to a SAF product. The user ID must be able to access an

INITIALn profile to connect and issue commands; it should be granted the highest level of security defined for the installation. Then use NetView security to control which NetView operators (including automation tasks) have access to the sessions and commands. OMEGAMON security implementation is outside the scope of this IBM Redbook.

3.5.2 NetView security

Using the NetView Command Authorization Table (CAT), you can define security to control access to the INGSESS and INGOMX commands and their parameters. These two commands control who can start or stop a session and which commands they can issue. For example, a subset of operators should be granted access to INGSESS to start or stop the sessions between System Automation for z/OS and IBM Tivoli OMEGAMON. You should also use the CAT table to control access to INGOMX, as some of the OMEGAMON commands can have a widespread effect, such as altering common storage (CSAA) or stopping a thread (KILL).

To properly define command security, you will need to familiarize yourself with the syntax of the INGOMX and INGSESS commands and their parameters. The complete command syntax is not discussed in this IBM Redbook. You can use the NetView HELP command to display the syntax and description of the INGOMX and INGSESS commands.

In general, you will need to define CAT table entries to:

- Restrict access to the INGOMX and INGSESS commands and their parameters.
- Define operators (including automation tasks) to one or more groups.
- Grant group access to the INGOMX and INGSESS commands and their parameters.

Defining INGOMX and INGSESS in CAT

This section deals with restricting access to the commands and their parameters. The two System Automation for z/OS commands you will need to code CAT table statements for are:

- INGOMX: Use the command list name of INGROMX0 in the CAT table.
- INGSESS: Use the command list name of INGRYSS0 in the CAT table.

The following are some sample definitions that you can use

- Restrict INGOMX access to all sessions:

```
PROTECT *.*.INGROMX0.NAME.*
```

- Restrict INGOMX access to a session called OMIIMVSA:

```
PROTECT *.*.INGROMX0.NAME.OMIISC66
```

- Restrict INGOMX access to all OMEGAMON commands:

```
PROTECT *.*.INGROMX0.CMD.*
```

- Restrict INGOMX access to the KILL command:

```
PROTECT *.*.INGROMX0.CMD.KILL
```

- Restrict access to the INGSESS command for all sessions:

```
PROTECT *.*.INGRYSS0.NAME.*
```

- Restrict access to INGSESS for starting and stopping session INGSESS,REQ=START and REQ=STOP:

```
PROTECT *.*.INGRYSS0.REQ.START
PROTECT *.*.INGRYSS0.REQ.STOP
```

Note: Protecting INGSESS with the last 2 bullets above will permit all users to issue INGSESS REQ=DISPLAY to display all sessions and INGSESS REQ=DETAIL to display session details for a specific session if the user has been granted access to the session itself from the NAME.

Define operator groups

Define at least two groups of operators. The first group will be strictly operators and they will not be given access to start or stop sessions, for example. The second group will be administrators who will be given access to the same functions as the operators plus functions, such as the start or stop of sessions. The following statements defines OMOPERS and OMADMIN groups:

```
GROUP OMOPERS OPER1,OPER2,TIV001,TIV002
GROUP OMADMIN TIV001,TIV002,AUTRPC,AUTWRK01,AUTWRK02
```

Note: You should define all System Automation for z/OS work autotasks (AUTWRKxx) and the AUTRPC autotask in the OMADMIN group. Optionally, you may choose to define a third group of operators for the autotasks.

Grant access for groups to INGOMX and INGSESS

Define several PERMIT statements for each group of operators to grant access to INGOMX, INGSESS, and their respective command parameters. The following are some sample statements:

- Define a PERMIT statement to allow operators in the OMOPERS group access to the OMIISC66 session when using the INGOMX command (INGOMX NAME=OMIISC66):

```
PERMIT OMOPERS *.*.INGROMX0.NAME.OMIISC66
```

- Define a PERMIT statement to allow the operators in the OMADMIN group access to the OMEGAMON KILL command (INGOMX EX NAME=OMIISC66 CMD=KILL):

  ```
  PERMIT OMADMIN *.*.INGROMXO.CMD.KILL
  ```

Note: All command definitions that are provided in the PERMIT statement must be defined in the corresponding PROTECT statement.

In general, protect the session with the NAME parameter and the command with the CMD parameter. Then permit the operators or groups with access to a combination of these resources.

For more informtion about System Automation for z/OS security, see *IBM Tivoli System Automation for z/OS: Planning and Installation*, SC33-8261. For more informtion about NetView command security, see *IBM Tivoli NetView for z/OS Security Reference*, SC31-8870.

Authorizing INGOMX TRAP commands

Retrieving exceptions from an OMEGAMON requires that autotasks get access to the INGOMX TRAP command. This command executes OMEGAMON command EXSY.

Note: The EXSY command is used by the IBM Tivoli OMEGAMON for z/OS, DB2, and CICS. IBM Tivoli OMEGAMON for IMS uses a different command, XIMS. You can secure the XIMS command the same as the EXSY command.

As the command EXSY will be issued when an INGOMX TRAP command is issued, you must define a protect statement as follows:

```
PROTECT *.*.INGROMXO.CMD.EXSY
```

Add the corresponding PERMIT statements as necessary to allow the tasks access to the EXSY command:

```
PERMIT OMADMIN *.*.INGROMXO.CMD.EXSY
```

You will need to allow these tasks to use EXSY:

- System Automation for z/OS work autotasks (AUTWRKxx).
- AUTRPC autotask.
- Any NetView operators that need to retrieve exceptions.

Example 3-2 on page 63 shows an unauthorized access using INGOMX TRAP, XTYPE=XREP, NAME=OMIISC66.

Example 3-2 NetView log

```
BNH236E 'TIVO02' IS NOT AUTHORIZED TO USE THE KEYWORD 'CMD' AND VALUE
'EXSY' COMBINATION
BNH237E THE KEYWORD 'CMD' AND VALUE 'EXSY' ARE PROTECTED BY COMMAND
IDENTIFIER '*.*.INGROMXO.CMD.EXSY' IN 'TBLNAME=OMSEC66'
```

Handling special commands

Some OMEGAMON commands contain characters that require special handling in the CAT table definitions. The .RMF command is an example. When you define a PROTECT or PERMIT statement for .RMF, you need to replace the period with an at sign (@) such as:

```
PROTECT *.*.INGROMXO.CMD.@RMF
PERMIT OMADMIN *.*.INGROMXO.CMD.@RMF
```

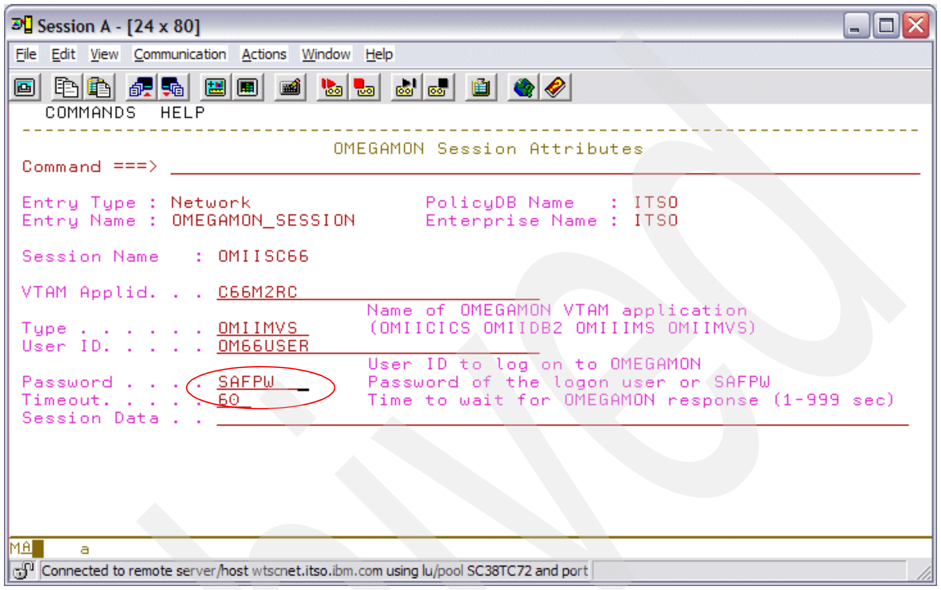
3.5.3 Password security

You can define the session password in two ways:

- As part of the OMEGAMON SESSIONS policy item when you define the session between System Automation for z/OS and an OMEGAMON monitor. Figure 3-6 on page 39 shows this definition.

  Note: Using the OMEGAMON SESSIONS policy item may not be secure enough for your environment since the password can be seen by anyone who has access to the System Automation for z/OS customization dialog.

- Alternatively, you can define the password field in the OMEGAMON SESSIONS policy item as SAFPW. SAFPW forces System Automation for z/OS to use encrypted passwords in the NetView password data set, EZLPSWD. The update definition is shown in Figure 3-28.

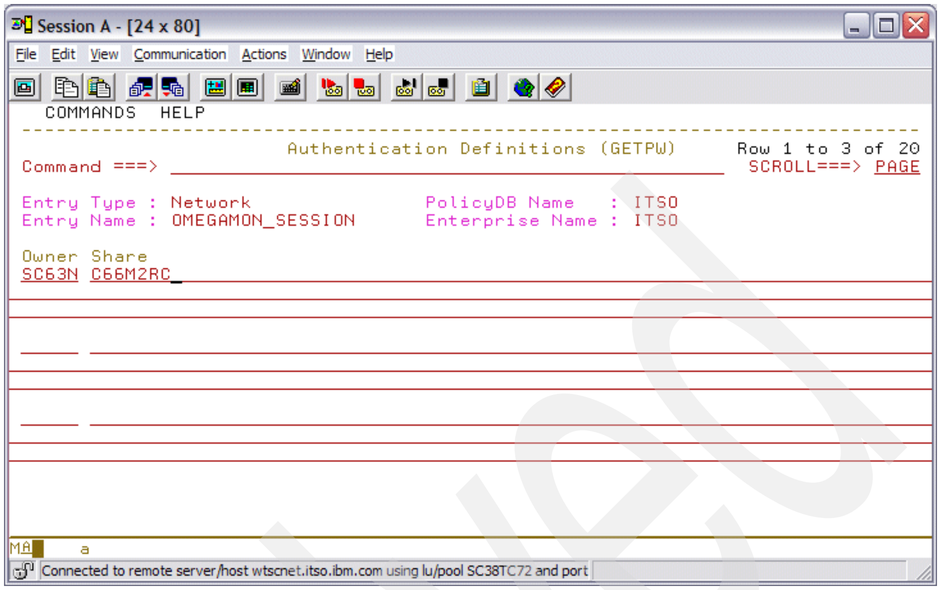


```
  COMMANDS   HELP
------------------------------------------------------------------------------
                         OMEGAMON Session Attributes
Command ===> __________________________________________________________________

Entry Type : Network                    PolicyDB Name    : ITSO
Entry Name : OMEGAMON_SESSION           Enterprise Name : ITSO

Session Name    : OMIISC66

VTAM Applid. . . C66M2RC_________________
                                   Name of OMEGAMON VTAM application
Type . . . . . . OMIIMVS_          (OMIICICS OMIIDB2 OMIIIMS OMIIMVS)
User ID. . . . . OM66USER________
                                   User ID to log on to OMEGAMON
Password . . . . SAFPW___          Password of the logon user or SAFPW
Timeout. . . . . 60_               Time to wait for OMEGAMON response (1-999 sec)
Session Data . . ____________________________________________________________

MA      a
Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC38TC72 and port
```

Figure 3-28 OMEGAMON session attributes screen

In Figure 3-28, the Password field is set to SAFPW, meaning that for this session, OMIISC66, System Automation for z/OS uses the NetView password data set, EZLPSWD, and the GETPW command to retrieve the passwords.

If you specify SAFPW for the session password, you must define the AUTHENTICATION policy item for the Network entry, as shown in Figure 3-29 on page 65.



```
Session A - [24 x 80]
File  Edit  View  Communication  Actions  Window  Help
   COMMANDS   HELP
 ------------------------------------------------------------------------------
                        Authentication Definitions (GETPW)      Row 1 to 3 of 20
 Command ===> ______________________________________________   SCROLL===> PAGE

 Entry Type : Network                   PolicyDB Name     : ITSO
 Entry Name : OMEGAMON_SESSION          Enterprise Name   : ITSO

 Owner  Share
 SC63N  C66M2RC_






MA      a
Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC38TC72 and port
```

Figure 3-29   Authentication definitions (GETPW) screen

This defines an owner of SC66N and a share list of C66M2RC, where:

- Owner: Specifies the name used as domain name parameter for the NetView GETPW command. In this example, the domain parameter will be SC66N.
- Share: A list of all the APPL IDs that use the owner to access the appropriate entry in the NetView password data set. The APPL IDs may delimited by blank or comma.

One advantage of using GETPW and encrypted passwords is that the password values are automatically updated every 30 days by the GETPW command. You must use the GETPW command to define the password values initially. For example, to define the password for OM66USER with a password of NEW2DAY, issue:

```
GETPW OM66USER SC66N,INIT=NEW2DAY,MASK=%A%A%A%N%A%A%A
```

The password mask indicates the password pattern that would be used to change the password.

All users (operators and autotasks) of a session between System Automation for z/OS and IBM Tivoli OMEGAMON must be authorized to use the GETPW command from the CAT table.

For additional installation and customization of the NetView password data set, see *IBM Tivoli NetView Installation: Configuring Additional Components*, SC31-8874.

4

System Automation for z/OS working with End-to-end Automation

This chapter details the setup of the System Automation for z/OS End-to-end Automation feature. It consists of the following sections:

4.1 Distributed software installation

We use an Intel®-based Linux machine running SUSE Linux Enterprise Server (SLES) 9. We are using the installation image from an IBM download site, which can be accessed from the Passport Advantage® Web site at:

http://www.ibm.com/software/passportadvantage

The installation images that we used is under the eAssembly package called IBM Tivoli System Automation for Multiplatforms End-to-End V2.1.1 eAssembly, Multiplatforms, Multilingual (CR2TFML). The files in that assembly are listed in Table 4-1.

Table 4-1 IBM Tivoli System Automation for Multiplatforms End-to-end V2.1.1

Description	Date	File name	Directory
IBM Tivoli System Automation for Multiplatforms V2.1.1 - End-to-End component, Linux for IBM System x™, Multilingual	03-Apr-2006	C892XML.tar	e2e211
IBM Tivoli System Automation for Multiplatforms V2.1.1 Base component, Linux, Multilingual	03-Apr-2006	C899KML.tar	sam211
WebSphere Application Server V6.0 Application Server, IBM HTTP Server, Web server plug ins, Application Clients for Linux (German, International English, Spanish, French, Italian, Japanese, Korean, Brazilian Portuguese, Simplified Chinese, and Traditional Chinese)	25-Nov-2004	C588FML.tar.gz	was6
WebSphere Application Server V6.0 Application Server Toolkit for Linux (German, International English, Spanish, French, Italian, Japanese, Korean, Brazilian Portuguese, Simplified Chinese, and Traditional Chinese)	23-Nov-2004	C5897ML.tar.gz	was6tk
IBM Tivoli System Automation for Multiplatforms V2.1.1 - WebSphere Application Server V6.0 Upgrade, Linux for IBM System x, Multilingual	03-Apr-2006	C8933ML.tar	was6upg
DB2 UDB Enterprise Server Edition V8.2 for Linux (2.6 kernel) on 32-bit Intel and AMD Systems (X86) (English, French, German, Spanish, Italian, Brazilian Portuguese, Russian, Polish, Czech, Japan, Korean, Simplified and Traditional Chinese)	29-Apr-2005	C829GML.tar	db28226

The list in Table 4-1 are files that we use with a Intel-based Linux server. The tar files are expanded using the **`tar -xvf`** command, while tar.gz files are expanded using the **`tar -xzvf`** command. We lists the target directory in the table to show the command that we will use on the installation process later.

The product comes assembled with all the necessary components. If you order the product from IBM, the same installation images come on CD-ROMs.

The installation that we perform follows the System Automation for z/OS manuals. They are available at:

`http://publib.boulder.ibm.com/tividd/td/IBMTivoliSystemAutomationforMultiplatforms2.1.1.1.html`

We use the following documents:

- *IBM Tivoli System Automation for Multiplatforms V2.1.1 Release Notes*, SC33-8214
- *IBM Tivoli System Automation for Multiplatforms V2.1.1 End-to-End Automation Management User's Guide and Reference*, SC33-8211

Some of the installation processes require you to have access to graphical interface tools, such as XServer or VNC. We are using the VNC interface.

An overview of the installation steps follows:

- 4.1.1, "Install DB2 Universal Database V8.2" on page 69
- 4.1.2, "Install DB2 Fix Pack" on page 70
- 4.1.3, "Install WebSphere Application Server" on page 71
- 4.1.4, "Install WebSphere Application Server patches" on page 73
- 4.1.5, "Installing System Automation for Multiplatform End-to-end" on page 74
- 4.2, "Operation for the middleware processes" on page 80

4.1.1 Install DB2 Universal Database V8.2

Here we install DB2 Universal Database™ V8.2.2, also known as DB2 8.1 Fix Pack 9.

1. We use the DB2 installation wizard from the image directory. Run **`db28226/334_ESE_LNX26_32_NLV/db2setup`** to start the DB2 Setup Wizard.
   a. Select **Install Products** on the left side of the display.
   b. Choose **DB2 UDB Enterprise Server Edition** and click **Next**. For each step, you must click the **Next** button.
   c. Accept the License Agreement.
   d. Choose the **Typical** installation method.

e. Make sure the check box is selected to install DB2 UDB Enterprise Server Edition on this computer.

f. Make sure **New user** is selected for the DB2 Administration Server window and type in the password twice. Also remember user ID dasusr1 and group dasadm1 for later use.

g. Make sure **Create a DB2 instance** is selected.

h. Make sure **Single-partition instance** is selected.

i. Make sure **New user** is selected for the DB2 instance owner window and key in a password twice. Also remember user ID db2inst1 and group db2grp1 for later use.

j. Make sure **New user** is selected for the fenced user and key in a password twice. Also remember user **ID** db2fenc1 and group db2fgrp1 for later use.

k. We select **Use a local database** for the tools catalog.

l. Accept the defaults of instance db2inst1 and a New Database of TOOLSDB and a New Schema of SYSTOOLS.

m. You can skip defining contact list and ignore the SMTP warning. Close the warning dialog boxes by pressing **OK**.

n. You now should see the Start copying files window. Look it over and make sure everything is correct and select **Finish** when you are ready to start the copying. You will see the progress bars for a few minutes while everything is installed.

o. You will eventually see the Setup is complete window. Examine it for any errors and correct as required. Click **Finish** when done.

2. Set up the DB2 environment for all user IDs you want to run DB2. You must source the environment for the db2profile. Add the following line to /etc/profile to change all users:

```
source /home/db2inst1/sqllib/db2profile
```

3. You can verify DB2 installation using the **`db2fs`** command. This is recommended after installing DB2; however, this is not required for the operation of the System Automation for z/OS End-to-end operation.

4.1.2 Install DB2 Fix Pack

Install the DB2 Fix Pack 8.2.3, also known as 8.1 Fix Pack 10. This is the required DB2 level that is needed by System Automation for z/OS End-to-end operation. The Fix Pack can be found at:

http://www-306.ibm.com/software/data/db2/udb/support/

The Fix Pack is also available through ftp at:

`ftp://ftp.software.ibm.com/ps/products/db2/fixes2/english-us/db2linux2632/fixpak/FP10_MI00142/FP10_MI00142.tar`

1. We create a directory called db28226_fp10 and place the FP10_MI00142.tar file there and then untar it with the **`tar -xvf`** command. We also download the FixPackReadme.txt readme file to the same directory. The readme has detailed instructions on how to install the Fix Pack, so make sure you read them closely.
   a. We shut down all the running DB2 processes. Run the **`ps -ef | grep db2`** command to check that there are no more running DB2 processes.
   b. We then install the Fix Pack by accessing the db28226_fp10 directory and issuing the **`./installFixPak -y`** command.
   c. Some of the DB2 processes are started back up by the previous step, specifically the db2fmd related processes, as well as db2dasrrm, so we shut them back down.
   d. Update the instance with the **`db2iupdt db2inst1`** command. If you get the message: `DBI1250E Applications are still using instance db2inst1`, you have to do the following steps as root:
      i. Enter **`ln -s /opt/IBM/db2/V8.1/lib/libdb2gcf.so /usr/lib/libdb2gcf.so`**.
      ii. Enter **`db2gcf -i db2inst1 -s -t 10`**, which should return:

      ```
      Instance : db2inst1
      DB2 State : Operable
      ```

      iii. You should now be able to run the **`db2iupdt`** command again and it should return successfully.
   e. We then update the DB2 Administrative server with the **`dasupdt dasusr1`** command, which starts the Administrative server as well.
   f. Since we only had a single database in our system (toolsdb), we run the **`db2updv8 -d toolsdb`** command, which updates the database.
   g. Now we restart everything as the db2inst1 user ID by running **`db2start`**.
2. Verify DB2 with the **`db2fs`** command again.

4.1.3 Install WebSphere Application Server

The WebSphere Application Server installation is performed using the base version 6.0 as follows:

1. Start the installation wizard for WebSphere Application Server. You can use either the **`launchpad.sh`** or the **`was6/WAS/install`** executable.

Note: We had trouble getting **launchpad.sh** to work with Firefox so we just started Firefox and then used **File** → **Open** to load the readme_base_en.html file. It has good information about various setup scenarios. We then opened the top level launchpad_base_en.html file that was in the launchpad directory. This page has all the links needed to complete the install.

We recommend installing WebSphere using a custom installation and do not install the sample applications. The sample applications would add unnecessary processing load to WebSphere. This will create the server1 instance of the application server.

2. From the launchpad Web page, we install IBM HTTP Server. You can also install this using the installer in was6/IBMHTTPServer/install. You can use all the default options. The Plugins wizard starts and we skip the Installation roadmap box. We select the IBM HTTP Server V6 plug-in check box and select the WebSphere Application Server machine (local) check box since both the WebSphere Application Server and HTTP servers are on the same machine in our setup. As we use the default paths, we can leave the default installation directories for both the Plugins and AppServer. We do need to specify the configuration file at /opt/IBMIHS/conf/httpd.conf. The Create default Web server name is set to webserver1. Use the default for most of the values.
3. Verify your setup.
   a. Verify that WebSphere is started by using **`ps -ef | grep server1`**. If it is not started, run **`/opt/IBM/WebSphere/AppServer/bin/startServer.sh server1`**.
   b. Start the IBM HTTP Server by running **`/opt/IBMIHS/bin/apachectl start`**. You can verify that it is running using the **`ps -ef | grep http`** command, it should show some processes.
   c. We then checked to make sure WebSphere is running using the snoop servlet. As our server name is peoria.itsc.austin.ibm.com, we open the URL `http://peoria.itsc.austin.ibm.com:9080/snoop`.
   d. Next, we verify that the HTTP server is running and has a correct plug-in using the URL `http://peoria.itsc.austin.ibm.com/snoop`. This should return the same page.
   e. We also check out the WebSphere Application Server Administrative Console in the URL `http://peoria.itsc.austin.ibm.com:9060/ibm/console/`.

4.1.4 Install WebSphere Application Server patches

IBM Tivoli System Automation for Multiplatform V2.1.1 requires WebSphere Application Server V6.0.2 with interim fix 1 and interim fix 2.

Refresh Pack and interim fixes

1. Install WebSphere Application Server Refresh Pack 2. This brings just the WebSphere Application Server server component to the 6.0.2 level.
   a. We log in as root and stop the HTTP server using the **`/opt/IBMIHS/bin/apachectl stop`** command and stop WebSphere Application Server using the **`/opt/IBM/WebSphere/AppServer/bin/stopServer.sh server1`** command and then verify they are both down with the **`ps -ef`** command.
   b. Change the WebSphere installation directory /opt/IBM/WebSphere/AppServer and expand the refresh pack by running **`tar -xvf /was6upg/WAS6020i386/Upgrade/6.0-WS-WAS-LinuxX32-RP0000002.tar`**, which creates various files in the updateinstaller directory below the AppServer directory.
   c. Activate the WebSphere command environment using the **`source /opt/IBM/WebSphere/AppServer/bin/setupCmdLine.sh`** command.
   d. Run the update installer wizard using the **`/opt/IBM/WebSphere/AppServer/updateinstaller/update`** command.
   e. The wizard will initially refresh the Java™ environment and you are required to restart the wizard to use a temporary Java environment so it can update the WebSphere Java environment.
   f. From the wizard, select **Install maintenance package** check box. It will discover Refresh Pack 2. The installation takes a while, after it completes successfully, click **Finish**.
2. Using the same procedure, we install the interim fixes. The fixes are installed using the same update wizard from /opt/IBM/WebSphere/AppServer/updateinstaller/update. There are two interim fixes that must be installed for IBM Tivoli System Automation for Multiplatforms. To install the interim fix, use the **Install maintenance** check box. The first fix resides in /was6upg/WAS6020i386/Fixes/01_PK07351; follow the wizard to install 01_PK07351.
3. When installation is complete, you can click **Relaunch** to start the wizard again. This time use the fix in /was6upg/WAS6020i386/Fixes/02_PK09347. After the installation is successfully performed, click **Finish**.
4. Restart the application server using the **`/opt/IBM/WebSphere/AppServer/bin/startServer.sh server1`** commands

and then start the HTTP server using the `/opt/IBMIHS/bin/apachectl start` command.

Use the snoop application to check that WebSphere and the HTTP Server are running properly.

4.1.5 Installing System Automation for Multiplatform End-to-end

The System Automation for Multiplatform End-to-end component is installed using the image that we unpack to the /e2e211 directory.

1. If you have not sourced the db2profile, you must source it before running the installation. We recommend putting it in /etc/profile to be executed by everyone that using the system.
2. Run the installation wizard using `/e2e211/EEZ2110E2EI386/i386/setup` command.
   a. Click **Next**, accept the License Agreement, and then leave the **End-to-End System Automation Management component** check box checked, and then accept the default directory for the product.
   b. After it checks for a previous operations console installation, accept the default Tivoli Common directory name, leave the **IBM DB2 UDB on same system (local)** check box checked, and then accept the default Automation Manager database name.
   c. On the next screen, accept the default location of the DB2 product and key in the db2inst1 password from the DB2 install. We accept the default host name and port on the next screen. Note that you can check DB2 port from /etc/services by finding db2c_db2inst1. It then tries to verify connection to the database.
   d. Accept the default WebSphere Application Server directory location and the default WebSphere Application Server profile called default, and the WebSphere Application Server server name of server1. You can stop the WebSphere Application Server server and accept the default for other WebSphere Application Server related screens.
   e. Check for a previous installation of the operations console. Accept the default operations console database name, and leave the default **Enable security with a DB2 database user registry** check box checked. Accept the default directory for the operations console and key in a password twice for the console user ID.
   f. Check the operations console port numbers from /etc/services; we accept all the defaults. Remember the first one, 8421; this is used to access the console (unless you regenerate the plug-in, then you can access the console using port 80). Accept the default Console help server port.

   g. We accept the default to register the ISC as a system service as well as the default Console Service ID. We then check the input values for the operations console.
   h. We do not check the **Enable TEC server connection** check box.
   i. We accept the default Automation domain name.
   j. In the summary information screen, click **Install**.

Note: If you get the installation error:

```
ISC Install RC = 1 For details see:
/opt/IBM/tsamp/eez/install/logs/ISCInstall.log
```

the file does not get created; the log file is in /tmp/ISCRuntimeInstall.log.

3. To verify the installation, we perform the following actions:
   a. Run DB2 Control Center using the **`db2cc`** command. Verify that the database EAUTODB existed and tables are created.
   b. Get to the admin console of WebSphere Application Server using the user ID of iscadmin. The administration console can be accessed from the URL:

      `http://peoria/ibm/console`
   c. You should also be able to log in as iscadmin ro manage System Automation for Multiplatform End-to-end. This is accessed at:

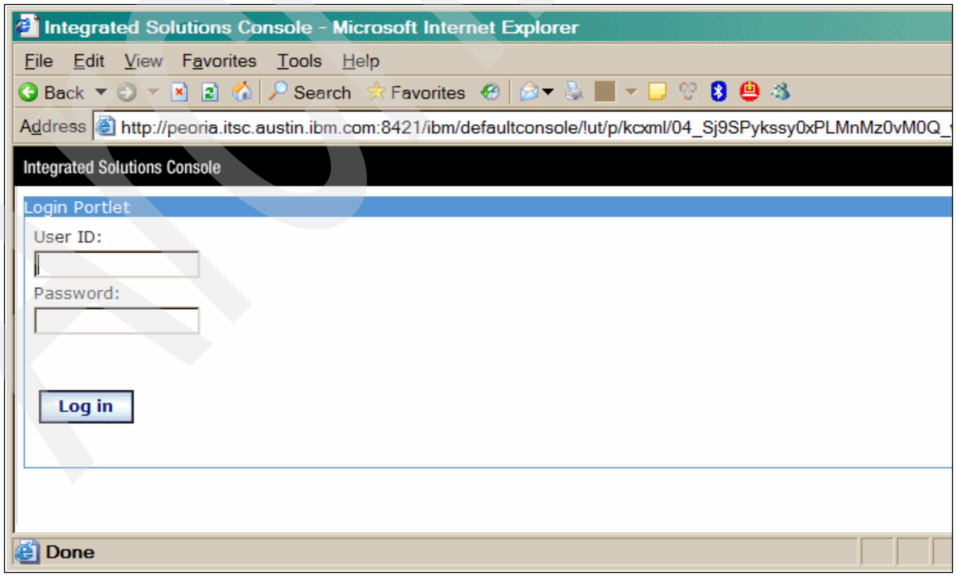
      `http://peoria/`

4. From /opt/IBM/tsamp/eez/bin, issue **./eezdmn.sh -?**, which returned the output in Figure 4-1.

```
IBM Tivoli System Automation end-to-end automation engine
Version: 2.1.1.061202, NO_APAR

Usage:
eezdmn [option]

  -START                   Starts the automation engine
  -SHUTDOWN | -SHUTD       Stops the automation engine
  -MONITOR  | -M           Displays the current state
  -RECONFIG | -R           Re-configures the automation engine
  -CO                      Starts only the EIF2JMS conversion thread
  -XD ("*" | "<RES_NAME>[,<RES_NAME>]") <DUMPFILE>
                           Dumps (all | specific) resources to a file

When no option is specified, START is used
```

Figure 4-1 Running ezdmn

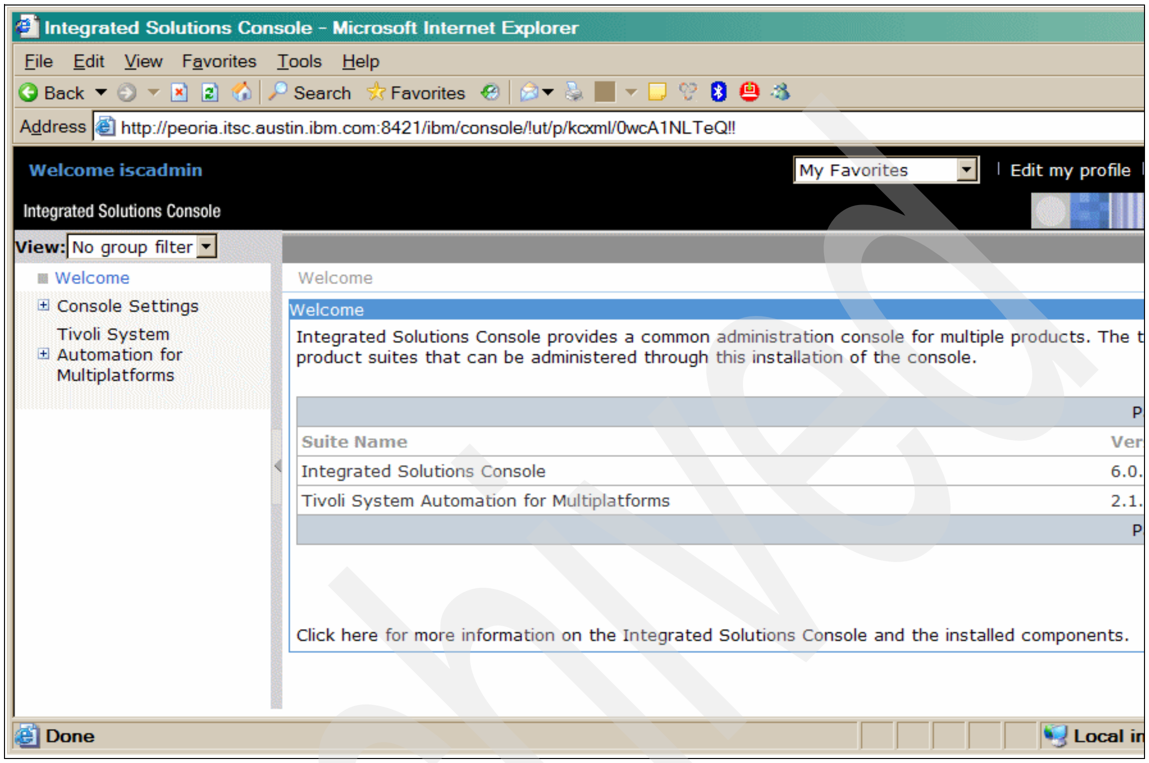
5. Log in to the management server at: `http://peoria.itsc.austin.ibm.com:8421/ibm/console`, where we see the screen shown in Figure 4-2.



http://peoria.itsc.austin.ibm.com:8421/ibm/defaultconsole/!ut/p/kcxml/04_Sj9SPykssy0xPLMnMz0vM0Q_

Figure 4-2 Log in page

d. Log in with the iscadmin user ID and password. You should see the screen in Figure 4-3.


Integrated Solutions Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://peoria.itsc.austin.ibm.com:8421/ibm/console/!ut/p/kcxml/0wcA1NLTeQ!!
Welcome iscadmin
My Favorites
Edit my profile
Integrated Solutions Console
View: No group filter
Welcome
Console Settings
Tivoli System Automation for Multiplatforms
Welcome
Welcome
Done
Local in

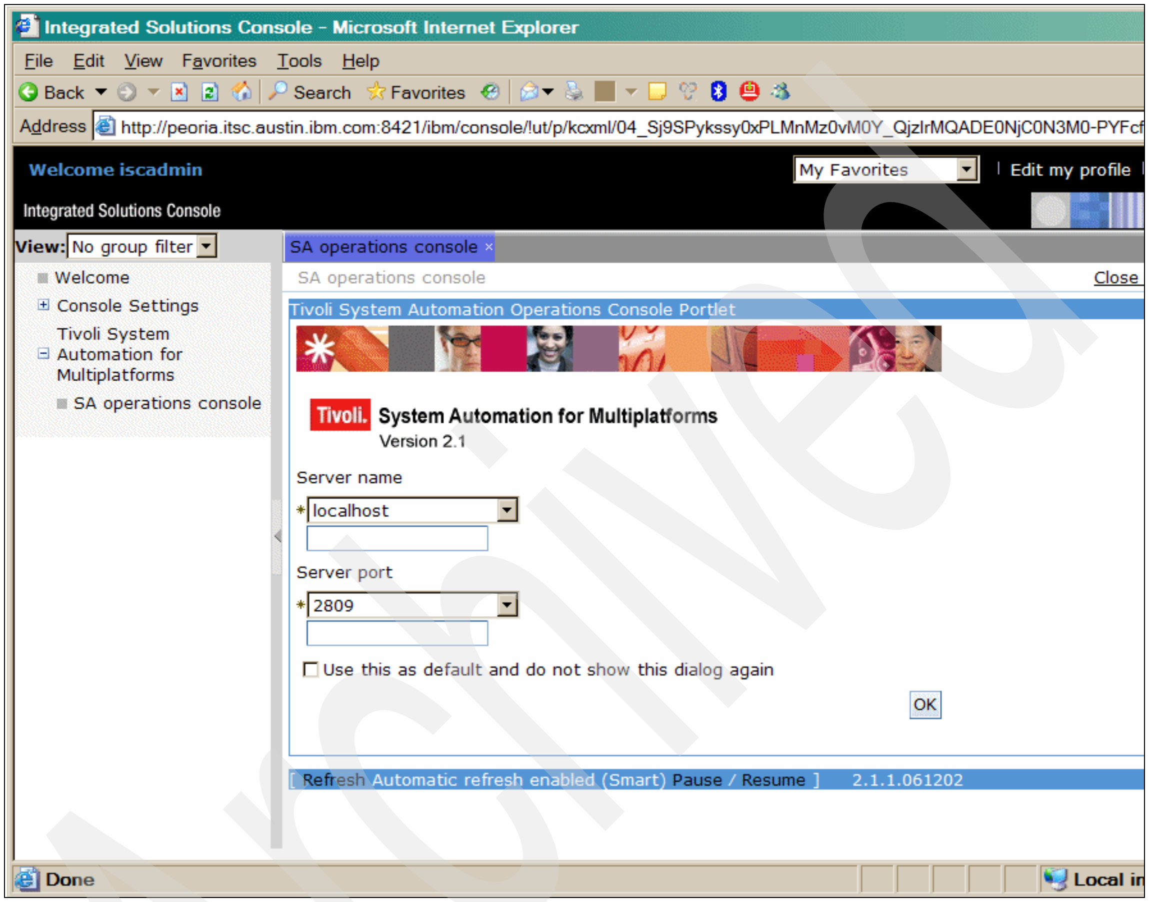
Integrated Solutions Console provides a common administration console for multiple products. The t
product suites that can be administered through this installation of the console.

Suite Name	Ver
Integrated Solutions Console	6.0.
Tivoli System Automation for Multiplatforms	2.1.

Click here for more information on the Integrated Solutions Console and the installed components.

Figure 4-3 Welcome page for System Automation for Multiplatform

e. Click on the + sign beside Tivoli System Automation for Multiplatforms (on the left) item and select the **SA operations console** link. You should see the screen in Figure 4-4.


Integrated Solutions Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://peoria.itsc.austin.ibm.com:8421/ibm/console/!ut/p/kcxml/04_Sj9SPykssy0xPLMnMz0vM0Y_QjzlrMQADE0NjC0N3M0-PYFcf
Welcome iscadmin
My Favorites
Edit my profile
Integrated Solutions Console
View: No group filter
Welcome
Console Settings
Tivoli System Automation for Multiplatforms
SA operations console
SA operations console
Close
Tivoli System Automation Operations Console Portlet
Tivoli. System Automation for Multiplatforms
Version 2.1
OK
[Refresh Automatic refresh enabled (Smart) Pause / Resume] 2.1.1.061202
Done
Local in

Server name: localhost

Server port: 2809

☐ Use this as default and do not show this dialog again

Figure 4-4 Selecting servers

f. We keep the defaults and click **OK**, and then see the window in Figure 4-5 on page 79.

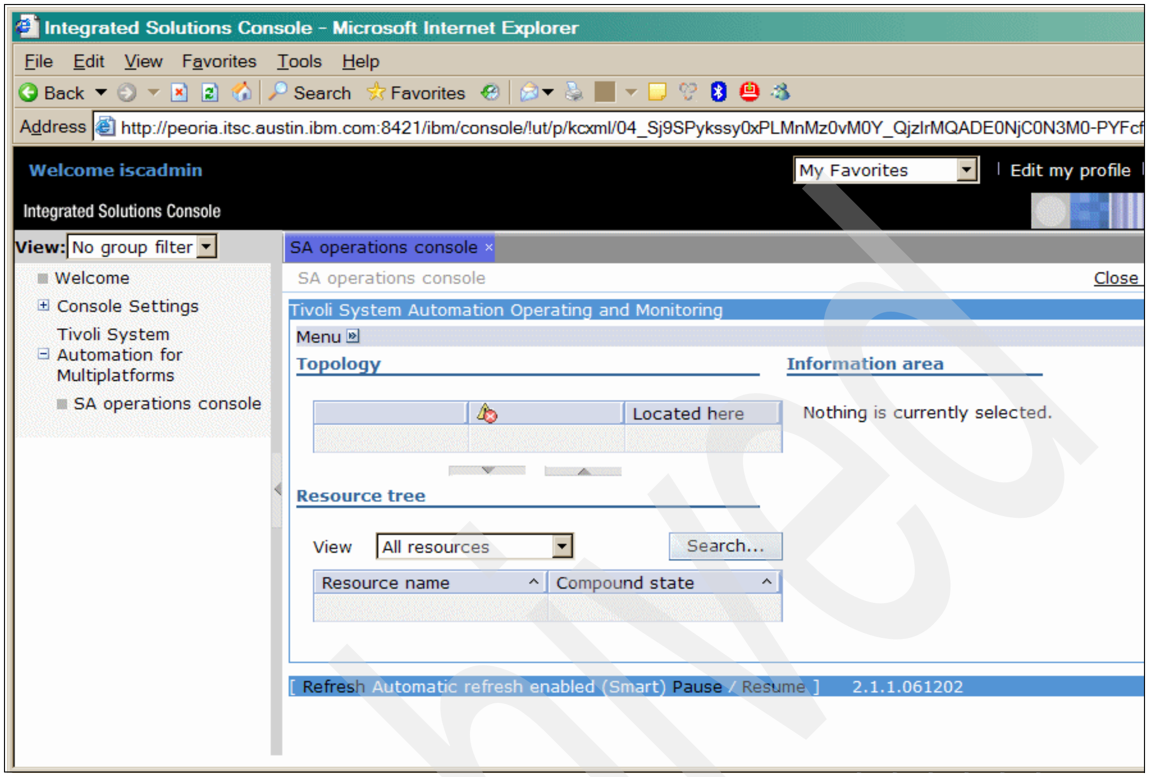

Integrated Solutions Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://peoria.itsc.austin.ibm.com:8421/ibm/console/!ut/p/kcxml/04_Sj9SPykssy0xPLMnMz0vM0Y_QjzlrMQADE0NjC0N3M0-PYFcf
Welcome iscadmin
My Favorites
Edit my profile
Integrated Solutions Console
View: No group filter
Welcome
Console Settings
Tivoli System Automation for Multiplatforms
SA operations console
SA operations console
SA operations console
Close
Tivoli System Automation Operating and Monitoring
Menu
Topology
Information area
Located here
Nothing is currently selected.
Resource tree
View
All resources
Search...
Resource name
Compound state
[Refresh Automatic refresh enabled (Smart) Pause / Resume] 2.1.1.061202

Figure 4-5 Loading topology

g. As this is a new system, there are no policies yet other than the provided sample.xml.

h. We select the **Log out** link on the upper right.

6. Generate the plug in for the Web server so that we can use the default port 80 to access the Integrated Solution Console for System Automation for Multiplatform.

4.2 Operation for the middleware processes

As there are multiple components in the solution, we must understand the operational consideration for starting and stopping each one of them.

4.2.1 DB2 Universal Database

Operating DB2 must be performed by the instance owner db2inst1. You can start the command as db2inst1 using the **`su`** command with the -c option.

- To stop DB2 instance, you can perform the following sequence:
  a. Check whether there is still an application accessing DB2 using the **`db2 list application`** command.
  b. You can wait until all transaction terminates or issue the **`db2 force applications all`** command.
  c. Stop the instance using db2stop; you can also forcing termination using the **`db2stop -force`** command.
- You can also stop the administration server as dasusr1. Stop the DB2 administration using the **`su -dasusr1 -c db2admin stop`** command.
- To start DB2, use the **`db2start`** command as the db2inst1 user.
- To start the DB2 administration instance, use the dasusr1 user and issue the command **`db2admin start`**.

4.2.2 WebSphere Application Server

The WebSphere Application Server hosts the System Automation for Multiplatform component with security turned on. You need a valid user ID and password to stop it. Assuming $WAS_HOME points to /opt/IBM/WebSphere/AppServer and you have the default profile as the default, you can:

- Stop the administration instance server1:
  `$WAS_HOME/bin/stopServer.sh server1 -username iscadmin -password password`
- Stop the ISC_Portal server instance:
  `$WAS_HOME/bin/stopServer.sh ISC_Portal -username iscadmin -password password`
- Start the administration instance server1:
  `$WAS_HOME/bin/startServer.sh server1`
- Start the ISC_Portal instance:
  `$WAS_HOME/bin/startServer.sh ISC_Portal`

4.2.3 IBM HTTP Server

IBM HTTP Server allows mapping of a standard Web address to a WebSphere Application Server request on the back end.

- Stopping IBM HTTP Server: **`/opt/IBMIHS/bin/apachectl stop`**
- Starting IBM HTTP Server: **`/opt/IBMIHS/bin/apachectl start`**

4.2.4 System Automation for Multiplatform

The command to manage System Automation for Multiplatform is **`eezdmn`**. See Figure 4-1 on page 76.

4.3 Post-installation tasks

Follow Chapter 12, “Post-installation tasks”, of *IBM Tivoli System Automation for Multiplatforms V2.1.1 End-to-End Automation Management User's Guide and Reference*, SC33-8211 to perform the post-installation tasks. We only perform the minimum required steps here.

Note: In the manual, there are references to EEZ_INST_ROOT or EEZ_INSTALL_ROOT for the location of all the components of the product. This refers to /etc/opt/IBM/tsamp/eez. This is explained in the Release Notes.

1. We do not set up any additional user IDs at this time, since we plan on using the iscadmin for everything.
2. We start the System Automation for Multiplatform automation engine by logging on as root and running /opt/IBM/tsamp/eez/bin/eezdmn.sh.
3. Log on to the System Automation for Multiplatform console using the iscadmin user:

   ```
   http://peoria.itsc.austin.ibm.com:8421/ibm/console
   ```
4. We activate the sample policy by selecting **General** → **Policy** and clicking the **Activate new policy...** button. Now you see the sample policy. We then click **Activate** to activate it so we could explore the console. After the processing, some resource names show up with their compound states displayed.
5. Now we need to create our own policy. The policy is an XML file; however, we decide to use the basic KDE provided Kate editor. We examine both the template.xml and the sample.xml policies located in $EEZ_INST_ROOT/policyPool. We create our policy called redbook.xml in this same directory.

4.4 End-to-end Automation Adapter configuration

End-to-end automation can be used to automate the operation of resources within heterogeneous environments (called first-level automation domains) that each have a local automation technology of their own. Each first-level automation domain is connected to the end-to-end automation manager by an automation adapter.

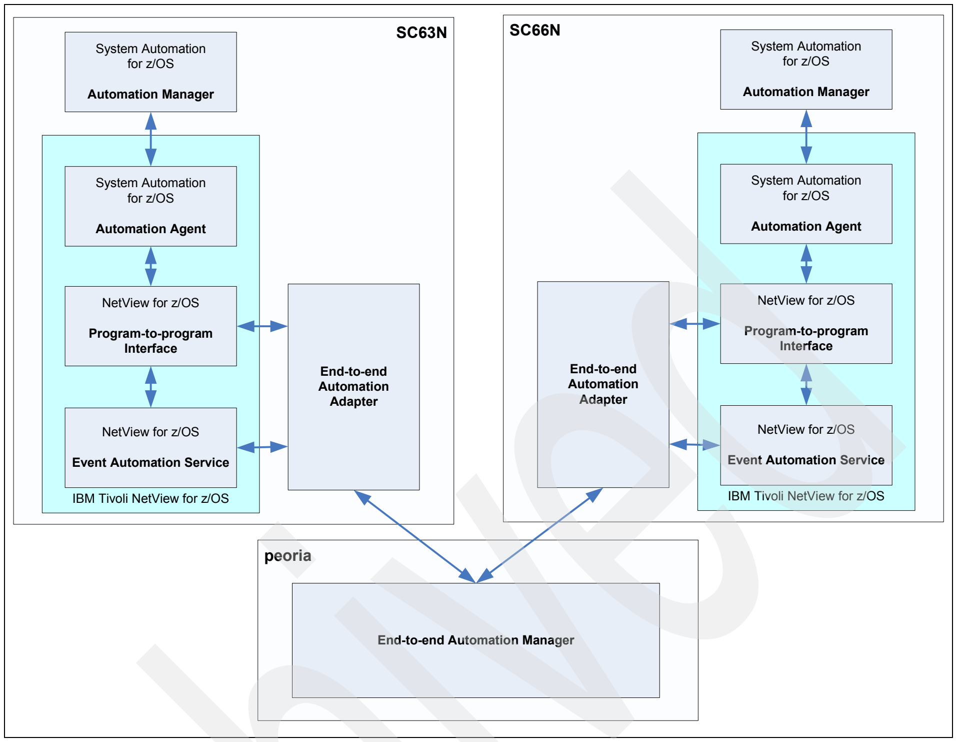
There can be only one End-to-end Automation Adapter per first-level automation domain. In the context of IBM Tivoli System Automation for z/OS V3.1, that means only one End-to-end Automation Adapter per IBM Tivoli System Automation for z/OS V3.1 z/OS sysplex group.

Although there can be more than one IBM Tivoli System Automation for z/OS V3.1 domain in a sysplex, there can be only one End-to-end Automation Adapter per z/OS sysplex group. In addition, the End-to-end Automation Adapter must run on the same system as the focal point IBM Tivoli System Automation for z/OS V3.1 agent.

When the End-to-end Automation Adapter runs on a z/OS system, the System Automation for z/OS V3.1 Agent on that z/OS system is the primary automation agent.

The End-to-end Automation Adapter registers with the primary automation agent at startup time. The End-to-end Automation Adapter also subscribes with the primary automation agent to enable communication channels. The primary automation agent then enables the End-to-end Automation Adapter to communicate with the primary IBM Tivoli System Automation for z/OS V3.1 Automation Manager.

After a system failure, event subscriptions are lost and the End-to-end Automation Adapter has to re-subscribe to the primary automation agent. In our case study scenario environment, we have two single z/OS systems. In this case, each End-to-end Automation Adapter primary automation agent and the primary IBM Tivoli System Automation for z/OS V3.1 Automation Manager are all running on the same z/OS system. See Figure 4-6 on page 83.



SC63N
System Automation for z/OS
Automation Manager
System Automation for z/OS
Automation Agent
NetView for z/OS
Program-to-program Interface
NetView for z/OS
Event Automation Service
IBM Tivoli NetView for z/OS
End-to-end Automation Adapter
SC66N
System Automation for z/OS
Automation Manager
System Automation for z/OS
Automation Agent
NetView for z/OS
Program-to-program Interface
NetView for z/OS
Event Automation Service
IBM Tivoli NetView for z/OS
End-to-end Automation Adapter
peoria
End-to-end Automation Manager

Figure 4-6 End-to-end automation domain

Using *IBM Tivoli System Automation for z/OS V3.1 End-to-end Automation Adapter*, SC33-8271 as a reference, setting up the System Automation for z/OS V3.1 End-to-end Automation Adapter in our environment requires the following configuration steps:

- 4.4.1, “Prerequisites and dependencies” on page 84
- 4.4.2, “Automated operator functions” on page 84
- 4.4.3, “Enabling the Event Automation Service” on page 85
- 4.4.4, “Customize the End-to-end Automation Adapter” on page 89
- 4.4.5, “Perform configuration for security” on page 95
- 4.4.6, “Verify startup of the Automation Adapter” on page 96
- 4.4.7, “Solve timeout problems” on page 97

4.4.1 Prerequisites and dependencies

In order to implement the End-to-end Automation Adapter on System Automation for z/OS V3.1, you must have the following prerequisites and dependencies:

- IBM Tivoli System Automation for z/OS V3.1
- IBM Tivoli NetView V5.1
- z/OS V1.3 or higher
- TCP/IP communication
- Java Runtime Environment (JRE™) 1.4.2
- Java Software Development Kit (SDK) for creating sample keys

Additionally, the following must be prepared and configured:

- Full z/OS UNIX® System Services (USS) with a hierarchical file system; the functionality is required by the Java subsystem, event automation service, and various other components.
- Event Automation Service (EAS) and Message Adapter Service components of NetView; this allow communication to the distributed system from NetView for z/OS.
- Subsystem interface (SSI) address space with Program-to-Program Interface (PPI) function of NetView; without this, the automation manager and automation agent subsystem cannot communicate with NetView as the automation engine.
- Configured NetView and IBM Tivoli System Automation for z/OS.

The System Automation for z/OS V3.1 communication task INGPXDST has an initialization member INGXINIT in DSIPARM that we use to specify End-to-end Automation Adapter specific parameters. In our environment, we add the following parameters to this member:

- PPI=YES

  This parameter is used for the End-to-end Automation Adapter.
- PPIBQL=1500

  This parameter controls the buffer queue length for input communication.

4.4.2 Automated operator functions

The End-to-end Automation Adapter uses dedicated automated operator functions in the primary agent to execute requests and to forward events. The automated operator functions E2EOPER, and E2EOPR01 through E2EOPR*nn*, are optional for the End-to-end Automation Adapter configuration. If defined, they are used to execute requests from the End-to-end Automation Adapter.

You *must* define the automated operator function EVTOPER. It is used to forward events to the End-to-end Automation Adapter. If not defined, the initialization of the End-to-end Automation Adapter fails.

We use the customization dialogs to define the automated operator functions for the End-to-end Automation Adapter in entry type Auto Operators of the System Automation for z/OS V3.1 policy database.

Note that the names of the Automation Operators must match those defined in the DSIPARM data set. The following operator group had to be selected to our z/OS systems SC63 and SC66:

- EVT_PUBLISHER: Event publisher
- E2E_AUTOOPS: Automated functions for End-to-end Automation Adapter

Figure 4-7 shows the output of the DISPAOPS command from system SC63N showing the defined automated operator functions.

```
AOFK2SO                  SA z/OS  - Command Dialogs      Line  27   of 51
Domain ID   = SC63N     ---------- DISPAOPS ----------    Date = 10/05/06
Operator ID = TIVO03                                      Time = 15:09:55

           Automated
 System    Function       Primary    Status      Secondary   Status
 --------  ----------     -------    ------      ---------   ------
 SC63      EVTOPER        AUTEVT1     ACTIV      AUTEVT2     ACTIV
 SC63      E2EOPER        AUTE2E      ACTIV
 SC63      E2EOPR01       AUTE2E01    ACTIV
 SC63      E2EOPR02       AUTE2E02    ACTIV
 SC63      E2EOPR03       AUTE2E03    ACTIV
```

Figure 4-7 DISPAOPS result

4.4.3 Enabling the Event Automation Service

The Event Automation Service (EAS) can be started either with a job from an MVS system console, or from a UNIX System Service command shell. In our case study scenario, we automate the EAS start on our IBM Tivoli System Automation for z/OS V3.1. Our EAS resource name is called E2E_EAS and the MVS job name is called AOFAEVNT.

You can find a sample for starting EAS as a job located in SCNMUXMS as the member IHSAEVNT. The initialization files are assumed to be located in a data set allocated to DD name IHSSMP3. We follow the description within the member for configuration.

Perform the following updates to the sample IHSAEVNT for our AOFAEVNT started task:

- If you do not hardcode the default name IHSAINIT for the global initialization file, pass the name of your file using the parameter INITFILE.
- If you do not hardcode the default name IHSAMCFG for the message adapter configuration file, pass the name of your file using the parameter MSGCFG.
- Modify the DD statements to specify the data set names in your installation.

The startup parameters have to be provided in the form of initialization files, and we discuss configuration in the following sections:

- “Configure the Global Initialization File” on page 86
- “Configure the NetView message adapter service” on page 87

Configure the Global Initialization File

The default member name for the global initialization file is IHSAINIT. This is found in the IBM-supplied dataset SCNMUXMS. This member was copied into our user data set name NETVUSER.SCNMUXMS using System Automation for z/OS V3.1 job AOFAEVNT. We follow the description within the member for configuration.

- Make sure that the NetView message adapter service is also started when starting EAS. This is done by commenting out the following statement:

  ```
  * NOSTART TASK=MESSAGEA
  ```

- Specify the Program to Program Interface (PPI) called INGEV2E2 for the receiver:

  ```
  PPI=INGEVE2E
  ```

We change the IHSAINIT initialization file so that only the necessary tasks for our environment are started at initialization time. Example 4-1 shows the configuration in the NETVUSER.SCNMUXCL(IHSAINIT) member. We show here only the lines that are relevant to our scenario. All the other lines are commented out.

Example 4-1 Message adapter task

```
# PPI Receiver ID
PPI=INGEVE2E
# Tasks not started at initialization
NOSTART TASK=ALERTA
#NOSTART TASK=MESSAGEA
NOSTART TASK=EVENTRCV
NOSTART TASK=ALRTTRAP
```

```
NOSTART TASK=TRAPALRT
```

Configure the NetView message adapter service

The default member name for the message adapter service is IHSAMCFG in SCNMUXMS. We copy this member into our data set name NETVUSER.SCNMUXMS. We change this IHSAMCFG for the following:

- We specify the address 127.0.0.1 as the server host name.
- We keep the port number on which the End-to-end Automation Adapter listens as the default of 5529.

> **Note:** Port 5529 is typically used for the Tivoli Enterprise Console® incoming event port, which is the port that the Event Automation server uses.

- Set the connection mode in connection_oriented. This allows the connection to be established at initialization time and closed at End-to-end Automation Adapter termination time.
- We keep the default value of 4096 for the maximum event size.
- We specify the name of the NetView message adapter format file as INGMFMTE. The version of the file that is to be used by end-to-end automation is delivered in ING.SINGSAMP(INGMFMTE). No configuration is required.

The resulting IHSAMCFG in our environment is shown in Example 4-2.

Example 4-2 Sample IHSAMCFG

```
ServerLocation=127.0.0.1
ServerPort=5529
ConnectionMode=connection_oriented
BufferEvents=yes
BufferEventsLimit=0
BufferFlushRate=0
BufEvtPath=/etc/Tivoli/tec/cache_nv390msg
BufEvtMaxSize=64
BufEvtShrinkSize=8
BufEvtRdBlkLen=64
EventMaxSize=4096
AdapterFmtFile=INGMFMTE
FilterMode=out
```

An excerpt of the AOFAEVNT that we use is shown in Example 4-3.

Example 4-3 Excerpt of AOFAEVNT startup procedure

```
//EASGO PROC PROG=IHSAC000, ** EVENT/AUTOMATION SERVICE
// REG=32M, ** REGION SIZE IN K FOR MAIN TASK
//* TCPDATA='', ** SET FOR TCPIP.DATA EXPLICIT ALLOC
// MSGCFG=, ** MESSAGE ADAPTER CONFIG FILE
// ALRTCFG=, ** ALERT ADAPTER CONFIG FILE
// ERCVCFG=, ** EVENT RECEIVER CONFIG FILE
// TALRTCFG=, ** TRAP TO ALERT CONV. CONFIG FILE
// ALRTTCFG=, ** ALERT TO TRAP CONV. CONFIG FILE
// PPI=INGEVE2E, ** PPI RECEIVER ID
// INITFILE=, ** EVENT AUTOAMTION SVC INIT FILE
// OUTSIZE=, ** TRACE/ERROR WRAP SIZE IN KBYTES
// OELINE= ** SPECIFY OE-STYLE PARAMETERS
//*
//*********************************************************************
//*
//STEP1 EXEC PGM=&PROG,TIME=1440,REGION=&REG,
// PARM=('/MSGCFG=&MSGCFG ALRTCFG=&ALRTCFG ERCVCFG=&ERCVCFG*
// TALRTCFG=&TALRTCFG ALRTTCFG=&ALRTTCFG PPI=&PPI *
// INITFILE=&INITFILE OUTSIZE=&OUTSIZE &OELINE')
//*
//STEPLIB DD DSN=NETVIEW.SCNMUXLK,DISP=SHR
//IHSSMP3 DD DSN=NETVUSER.SCNMUXCL,DISP=SHR
// DD DSN=NETVIEW.SCNMUXCL,DISP=SHR
//IHSMSG1 DD DSN=NETVIEW.SCNMUXMS,DISP=SHR
//* EAS OUTPUT DATASETS
//IHSC DD SYSOUT=A
//IHSM DD SYSOUT=A
//IHSA DD SYSOUT=A
//IHSE DD SYSOUT=A
//IHST DD SYSOUT=A
//IHSL DD SYSOUT=A
//IHSCS DD SYSOUT=A
//IHSMS DD SYSOUT=A
//IHSAS DD SYSOUT=A
//IHSES DD SYSOUT=A
//IHSTS DD SYSOUT=A
//IHSLS DD SYSOUT=A
//IHSCSTD DD SYSOUT=A
//IHSASTD DD SYSOUT=A
//IHSMSTD DD SYSOUT=A
//IHSESTD DD SYSOUT=A
```

```
//IHSTSTD DD SYSOUT=A
//IHSLSTD DD SYSOUT=A
//SYSTERM DD SYSOUT=A
```

4.4.4 Customize the End-to-end Automation Adapter

We are now ready to configure the End-to-end Automation Adapter.

The End-to-end Automation Adapter SMP/E installation creates a default directory structure for the various files that are associated with the End-to-end Automation Adapter, as follows:

/usr/lpp/ing/adapter	This contains the executable files, for example, the End-to-end Automation Adapter start and stop scripts.
/usr/lpp/ing/adapter/config	This contains configuration files, for example, the master configuration file.
/usr/lpp/ing/adapter/data	Working files, for example, the cache and log files. This directory is initially empty.
/usr/lpp/ing/adapter/lib	This contains JAR files and DLLs for the End-to-end Automation Adapter.
/usr/lpp/ing/adapter/ssl	Security certificates. This directory is initially empty.

The files and structures created by our SMP/E installation were read only. A separate structure and files were created to copy SMP/E files that need to be updated by the installation or during the execution of the automation adapter. That file structure was created as follows:

/var/ing/	The user defined structure containing the End-to-end Automation Adapter start and stop scripts.
/var/ing/config	The user defined structure containing the configuration files.
/var/ing/data	The user defined structure containing the working files.
/var/ing/ssl	The user defined structure for security certificates as needed.

To customize the End-to-end Automation Adapter, we perform configurations on the following elements:

- "RACF authorization for z/OS UNIX privileges" on page 90

- "The End-to-end Automation Adapter shell script" on page 90
- "The link list" on page 92
- "The End-to-end Automation Adapter master configuration file" on page 92
- "The End-to-end Automation Adapter plug-in configuration file" on page 93
- "The JAAS configuration file" on page 94

RACF authorization for z/OS UNIX privileges

You can define profiles in the UNIXPRIV class to grant RACF authorization for certain z/OS UNIX privileges. These privileges are automatically granted to all users with z/OS UNIX superuser authority. By defining profiles in the UNIXPRIV class, you may specifically grant certain superuser privileges with a high degree of granularity to users who do not have superuser authority. This allows you to minimize the number of assignments of superuser authority at your installation and reduces your security risk.

Since our case study scenario is in a controlled environment, we decide to grant our RACF user ID with superuser authority.

Refer to *Security Configuration in a TCP/IP Sysplex Environment*, SG24-6527 for instructions on how to manipulate security privileges for your environment.

The End-to-end Automation Adapter shell script

If you want to use your own hierarchical file system (HFS) or a shared HFS, you have to modify the End-to-end Automation Adapter start script and its configuration files.

In our case scenario, we copy the ingadapter.sh USS shell script from /usr/lpp/ing/adapter/ingadapter to /var/ing/ingadapter for editing. We also copy the following configuration files from /usr/lpp/ing/adapter/config to var/ing/config.

- ing.adapter.jaas.properties
- ing.adapter.jlog.properties
- ing.adapter.jlogdir.properties
- ing.adapter.plugin.properties
- ing.adapter.properties
- ing.adapter.ssl.properties

We changed the following parameters in the ingadapter.sh script for our case study scenario. Entries that are not mentioned in the following list are not modified and are left with their default values.

- We defined the file structure to use for configuration files as follows:

  CONFIG_DIR=/var/ing

- We updated the ingadapter shell script to use the configuration file structure.

- We comment out the SSL_PASSWD parameter because we are not using the GenerateSampleKeys function.
- We set the file names and create the empty files for the standard output and error streams using the REDIRSTDOUT and REDIRSTDERR entries in the ingadapter.sh script to match the End-to-end Automation Adapter start up procedure script on z/OS (AOFAADPT). The entries in the AOFAADPT are as follows.

  ```
  //STDOUT DD PATH='/var/ing/data/stdout.log',
  //STDERR DD PATH='/var/ing/data/stderr.log',
  ```
- We specify the path to JAVA runtime in our installation.

Example 4-4 shows the customization section of ingadapter.sh. Changed attributes are marked in bold.

Example 4-4 Customized ingadapter.sh

```
#-----------------------------------------------------------------------
# Customization section begins here
#-----------------------------------------------------------------------
INSTALL_DIR=/usr/lpp/ing/adapter
CONFIG_DIR=/var/ing
#export STEPLIB="HLQ.SINGMOD1":$STEPLIB
#SSL_PASSW=passphrase

REDIRSTDOUT=/var/ing/data/stdout.txt
REDIRSTDERR=/var/ing/data/stderr.txt

export INGXDBUG=0,2,1

JAVA=/usr/lpp/java/J1.4/bin/java
JAVA_KEYTOOL=keytool
DATA_DIR=$CONFIG_DIR/data
SSL_DIR=$CONFIG_DIR/ssl
CONFIG_DIR=$CONFIG_DIR/config

ADAPTER_CONFIG=$CONFIG_DIR/ing.adapter${SUFFIX}.properties
JLOG_FILENAME_LOGPATH=ing.adapter.jlogdir.properties
JLOG_FILENAME=ing.adapter.jlog.properties
JLOG_CONFIG_DIR=$CONFIG_DIR
JAAS_CONFIG=$CONFIG_DIR/ing.adapter.jaas.properties
SSL_CONFIG=$CONFIG_DIR/ing.adapter.ssl.properties
#-----------------------------------------------------------------------
# Customization section ends here.
#-----------------------------------------------------------------------
```

The link list

We add the following libraries to the link list in our case study environment:

- SYS1.SCLBDLL2
- SYS1.SCEERUN
- SYS1.SCEERUN2
- SYS1.SCLBDLL

You can check that these libraries have been linked using the D PROG,LNKLST command.

The End-to-end Automation Adapter master configuration file

The master configuration file for the End-to-end Automation Adapter is specified in the ingadapter.sh script as $CONFIG_DIR/ing.adapter${SUFFIX}.properties. As we do not use a suffix in our environment, we have to configure the file /var/ing/config/ing.adapter.properties.

We change the following parameters in the End-to-end Automation Adapter master configuration file for our case study scenario. Entries that we do not mention here in the following list were not modified and were left with their default values.

- The IP address of our z/OS system on which the End-to-end Automation Adapter receives requests from the End-to-end Automation Management Component. This is specified in the `eez-remote-contact-hostname` entry.
- We set `eez-operator-authentication` to `false` because we are using the default IBM Tivoli System Automation for z/OS V3.1 JAAS login modules for authentication.
- We specify the fully qualified host name of the server on which the End-to-end Automation Management Component runs in our environment using the `eif-send-to-hostname` entry.

Example 4-5 shows our End-to-end Automation Adapter master configuration file. Changed attributes are marked in bold.

Example 4-5 End-to-end automation adapter master configuration file

```
# --- Adapter Configuration -----

eez-remote-contact-hostname   = 9.3.5.104
eez-remote-contact-port       = 2001
eez-remote-contact-over-ssl   = false
eez-operator-authentication   = false
eez-initial-contact           = true
eez-max-connections           = 3
eez-data-directory            = ./data
```

```
# --- EIF Configuration ---------

eif-cache                    = true
eif-cache-size               = 500
eif-retry-interval-seconds   = 30
eif-send-to-hostname         = peoria.itsc.austin.ibm.com
eif-send-to-port             = 2002
eif-receive-from-hostname    = 127.0.0.1
eif-receive-from-port        = 5529


# --- Plugin Configuration ------

plugin-configfile-sa4zos = ./config/ing.adapter.plugin.properties
```

The End-to-end Automation Adapter plug-in configuration file

The master configuration file contains the location of the End-to-end Automation Adapter plug-in configuration file, as shown in the last line of Example 4-5 on page 92. The plug-in configuration file is located in $EEZ_INST_HOME/ing.adapter.plugin.properties.

We change the following parameters in the End-to-end Automation Adapter plug-in configuration file. Entries that are not mentioned in the following list were not modified and were left with their default values. We change the number of elements in the PPI queue using the PPIBQL entry.

- Since our z/OS system resides in a remote data center, we change the value of the TIMEOUT entry.
- We decided to change the plugin-domain-name entry to SC63N (and SC66N) similar to our NetView domain names. Here you can decide your own naming strategy; System Automation for z/OS V3.1 uses the XCF sysplex group name.

Example 4-6 shows the End-to-end Automation Adapter plug-in configuration file for our case study scenario. Changed attributes are marked in bold.

Example 4-6 Content of the plugin configuration file

```
# --- Specific settings for the SA/Netview communication ---
# --- Modify these parameters to your needs ---
#GRPID = XX
PPIBQL = 3000
AUTOPFN = E2EOPER
TIMEOUT = 4447
CODEPAGE= Cp1047
# --- Domain name may be modified or omitted ---
plugin-domain-name = SC63N
# --- Do not modify these plugin settings ---
plugin-impl-class = com.ibm.ing.sam.INGXPlugin
plugin-impl-class-singleton = true
plugin-event-classes = SystemAutomation_Resource_Status_Change
SystemAutomation_Domain_Status_Change SystemAutomation_Resource_Conf
iguration_Change SystemAutomation_Request_Configuration_Change
SystemAutomation_Relationship_Configuration_Change
plugin-auto-start = true
```

The JAAS configuration file

The Java Authorization Authentication Services (JAAS) configuration file for the End-to-end Automation Adapter is specified in the ingadapter.sh script as follows:

```
JAAS_CONFIG=$CONFIG_DIR/ing.adapter.jaas.properties
```

This file specifies two entries for a login module that is required, because in our environment all incoming requests to the End-to-end Automation Adapter will be authenticated using JAAS. We have not changed this file and used all default values. See Example 4-7.

Example 4-7 Content of JAAS configuration

```
EEZAdapterDefaultLogin {
com.ibm.eez.adapter.EEZAdapterDefaultLoginModule required ;
};
EEZAdapterLogin {
com.ibm.security.auth.module.OS390LoginModule required;
};
```

4.4.5 Perform configuration for security

When the End-to-end Automation Management Component issues a request to the End-to-end Automation Adapter, there is always a user ID and password associated with the request. You need to consider both authentication and authorization aspects.

- Authentication

  In our environment, the End-to-end Automation Adapter performs authentication checks using RACF through the login modules specified in the JAAS configuration file. This decision is made at End-to-end Automation Adapter startup time in the master configuration file.

- Authorization

  In case the user ID authentication succeeds, the End-to-end Automation Adapter performs an authorization check for each of the commands in the request that are to be executed.

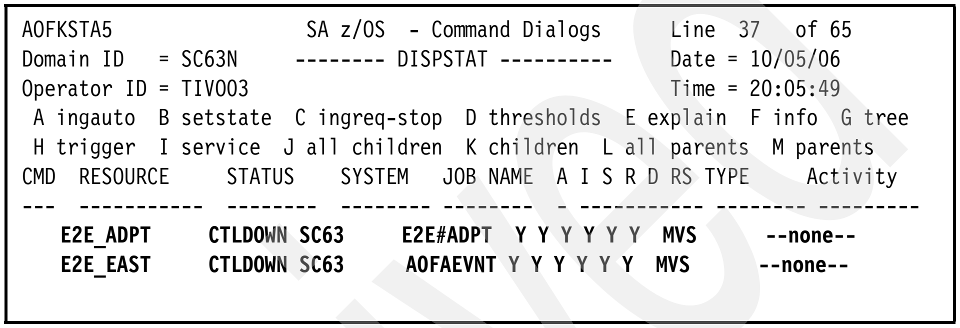
  The authorization check is done by an authorization user exit, which is an external REXX program that *must* be named AOFEXE2E. If no authorization user exit exists, the user ID associated with the request is considered to be authorized for each request.

We use the sample security exit that is provided by System Automation for z/OS V3.1 as the member AOFEXE2E in the sample data set SINGSAMP.

4.4.6 Verify startup of the Automation Adapter

Once we complete all the customization, we are ready to start the End-to-end Automation Adapter for z/OS.

Figure 4-8 displays the status of E2E_EAS and E2E_ADPT resources on NetView. The status of these resources are in CTLDOWN. We need to change the status of these resources to RESTART, starting with the E2E_ADPT first and then E2E_EAS second. This will start both resources in the proper order.



```
AOFKSTA5                 SA z/OS  - Command Dialogs      Line  37   of 65
Domain ID   = SC63N     -------- DISPSTAT ----------     Date = 10/05/06
Operator ID = TIV003                                     Time = 20:05:49
 A ingauto  B setstate  C ingreq-stop  D thresholds  E explain  F info  G tree
 H trigger  I service  J all children  K children  L all parents  M parents
CMD  RESOURCE     STATUS    SYSTEM   JOB NAME  A I S R D RS TYPE     Activity
---  -----------  --------  -------- --------  - ----------- -------- ---------
   E2E_ADPT     CTLDOWN SC63     E2E#ADPT  Y Y Y Y Y Y  MVS      --none--
   E2E_EAST     CTLDOWN SC63     AOFAEVNT Y Y Y Y Y Y  MVS      --none--
```

Figure 4-8 Status of resources E2E_EAS and E2E_ADPT

The sample console output on this start up is shown in Example 4-8.

Example 4-8 Starting end to end automation

```
$HASP100 AOFAADPT ON STCINRDR
IEF695I START AOFAADPT WITH JOBNAME AOFAADPT IS ASSIGNED TO USER
STC,GROUP SYS1
$HASP373 AOFAADPT STARTED
CNM493I INGMSG02 : 00000029 : AOCFILT AOFAADPT ACTIVMSG
JOBNAME=AOFAADPT
IEF403I AOFAADPT - STARTED - TIME=13.41.04 - ASID=0080 - SC63
INGX9704I Preparing the environment to start the automation adapter.
EEZA0100I The adapter has been started*
Initial contact was enabled and the connection to the management server
has been established*
EEZA0101I The adapter is active*
EEZA0111I The plug-in is starting: class com.ibm.ing.sam.INGXPlugin *
INGX9802I INGX9802I INGXLogger has successfully been initialized using
configuration file ing.adapter.jlog.properties from path /var/ing
/config.*
INGX9902I INGXPluginLogger has successfully been initialized using con
figuration file ing.adapter.jlog.properties from path
/var/ing/adapter/config.*
```

```
CNM493I INGMSG02 : 00003489 : ACTIVMSG UP=YES
EEZA0112I The plug-in has been started: class
com.ibm.ing.sam.INGXPlugin *
EEZA0102I The adapter is ready*
```

4.4.7 Solve timeout problems

Since we have relatively small z/OS systems, during the development of this IBM Redbook, we experienced timeout problems on the End-to-end Automation Management Component operations console when displaying IBM Tivoli System Automation for z/OS resource data from our automation domains.

The End-to-end Automation Adapter receives the requests from the End-to-end Automation Management Component operations console. These requests are then mapped to IBM Tivoli System Automation for z/OS commands that query the requested data from the End-to-end Automation Management Component automation manager. Before sending the query commands to the End-to-end Automation Management Component automation manager, the primary automation agent in z/OS checks the expiration time given to the requests. If the expiration time that remains is too short, the requested command will be rejected, resulting in error message ING249E that indicates that a task execution request timed out.

IBM Tivoli System Automation for z/OS V3.1 End-to-end Automation Adapter, SC33-8271 provides a mapping of end-to-end automation requests to System Automation for z/OS commands. Most of these commands contain a WAIT parameter. The value of the WAIT parameter is calculated as the difference between the time when the IBM Tivoli System Automation for z/OS command was issued within the NetView environment and the expiration time given to the end-to-end automation request.

The expiration time of an end-to-end automation request is determined by both:

- The time when the corresponding end-to-end automation request was issued from the End-to-end Automation Management Component operations console or the End-to-end Automation Management Component automation manager.
- The timeout in seconds defined by a environment variable defined in the IBM WebSphere Application Server on which the End-to-end Automation Management Component automation manager runs. The environment variable must be named `com.ibm.eez.aab.invocation-timeout-seconds` and must be set to a value in seconds. In our environment, we set the `com.ibm.eez.aab.invocation-timeout-seconds` to 2500 seconds.

5

Managing SYSPLEXes with the use of the Processor Operations feature

The chapter discusses the implementation of Processor Operations in our ITSO environment.

5.1 Processor Operations

The Processor Operations function of System Automation for z/OS helps operators manage more systems with greater efficiency. One operator, in one location, can initialize, configure, monitor, shut down, and recover multiple systems on both local and remote sites and respond to a variety of detected conditions. The operator, using one standard interface, can do all that across multiple types of systems. Automated routines for frequently used functions speed up the work of skilled operators and assist less experienced operators to become more productive.

Processor Operations supports the z9™, IBM System z, and S/390-CMOS processors. Other CMOS processors supporting Operations Command Facility (OCF) that are not part of the above processor families are supported with limited functionality. The target operating system for these processor can be any supported operating system, including z/OS, OS/390, MVS, VM, VSE, and Linux for IBM System z.

Remote IBM System z systems can now be controlled using Simple Network Management Protocol (SNMP) without any additional interface or networking other than an existing processor LAN.

Note: If you plan to use a z9 Hardware Management Console (HMC) in your Processor Operations configuration in order to monitor and control an IBM System z or S/390-CMOS processor (machine number 9672), the microcode of this HMC must be on the mandatory z9 level before it can be used with Processor Operations.

5.2 Planning the hardware interfaces

There are two different possibilities to gain access to a Central Processor Complex (CPC):

- Processor Operations (ProcOps)

  Processor Operations uses basic IP communication to access the CPC. System Automation for z/OS allows the possibility to use SNMP protocol as well as the already available SNA protocol.

- z/OS Base Control Program (BCP) internal interface

  The z/OS BCP internal interface is used by Extended Parallel Sysplex Automation, Geographical Diverse Parallel Sysplex (GDPS), and z/OS MSYS for Operations. It does not use a networking infrastructure; instead, it talks directly using API to the CPC. This interface is *not* a published interface.

The environment that we are working to implement is shown in Figure 5-1.

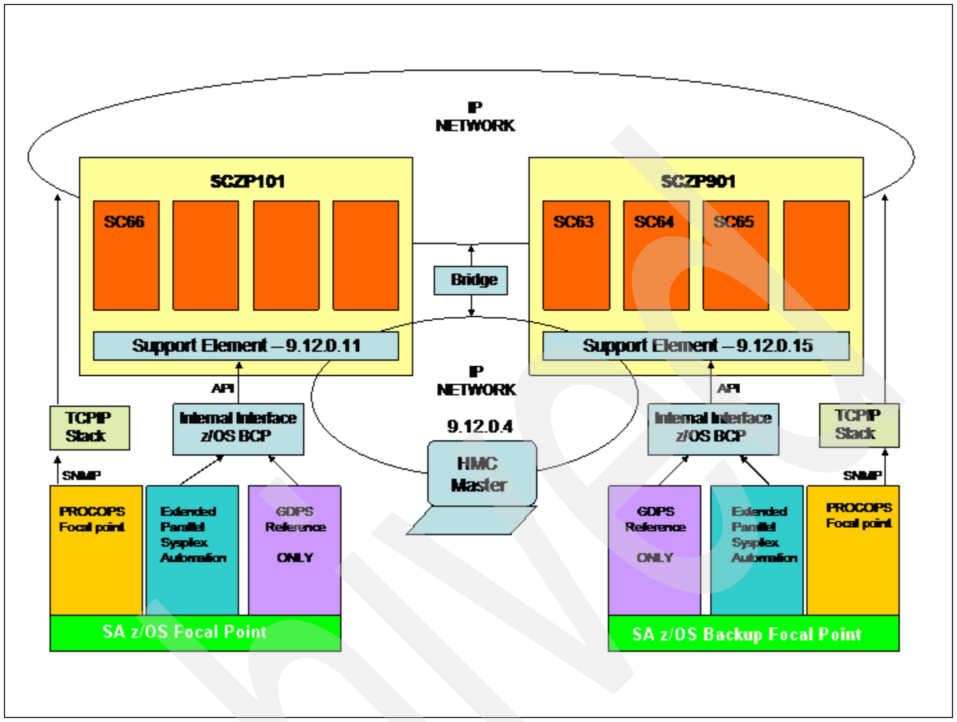

IP NETWORK
SCZP101
SC66
SCZP901
SC63
SC64
SC65
Bridge
Support Element – 9.12.0.11
Support Element – 9.12.0.15
IP NETWORK
9.12.0.4
HMC Master
API
API
TCPIP Stack
TCPIP Stack
Internal Interface z/OS BCP
Internal Interface z/OS BCP
SNMP
SNMP
PROCOPS Focal point
Extended Parallel Sysplex Automation
GDPS Reference ONLY
GDPS Reference ONLY
Extended Parallel Sysplex Automation
PROCOPS Focal point
SA z/OS Focal Point
SA z/OS Backup Focal Point

Figure 5-1 HMC configuration for ITSO environment

The assumption for the following sections are that System Automation for z/OS is already running and has an active automation profile.

5.3 Modifying automation policies

To enable Processor Operations, you must modify several automation policies:

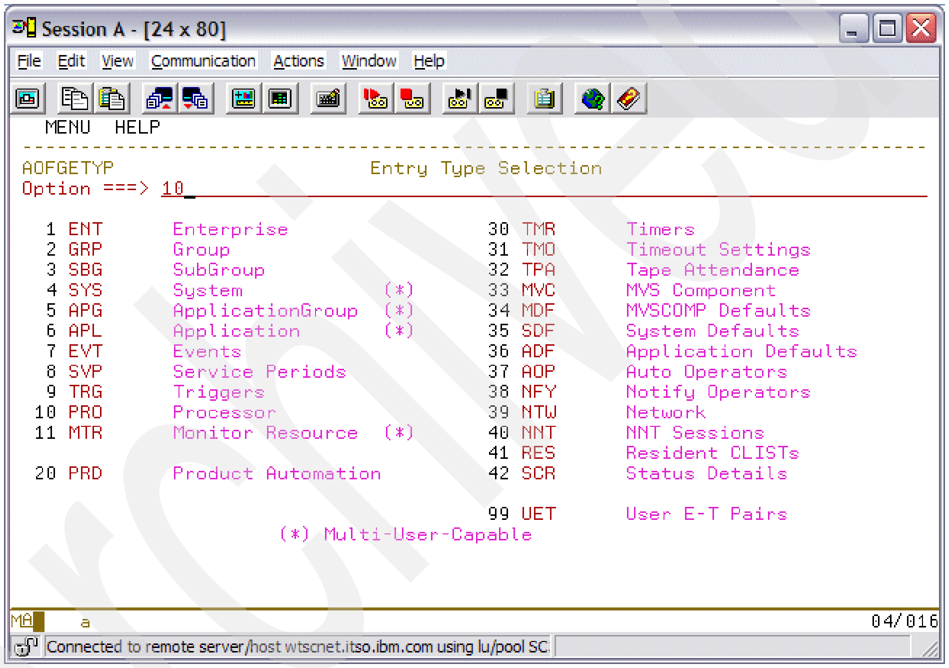
- 5.3.1, “Defining OCF-based processors” on page 101
- 5.3.2, “Connecting System to Processor” on page 108
- 5.3.3, “Enterprise Processor Operations information” on page 112

5.3.1 Defining OCF-based processors

This section describes how to define OCF-based processors to Processor Operations. In System Automation for z/OS, every connection to a CPC or an external coupling facility is described by a Processor entry. You can have more than one entry for the same processor, depending on the connection type; however, these entries must be defined with different entry names. The processor entry let you reach every LPAR or Coupling Facility.

The processor entry describes a connection mechanism, using internal or external interface, to a Support Element or HMC. The Support Element or HMC allows access for management of every operating system image running on these CPCs.

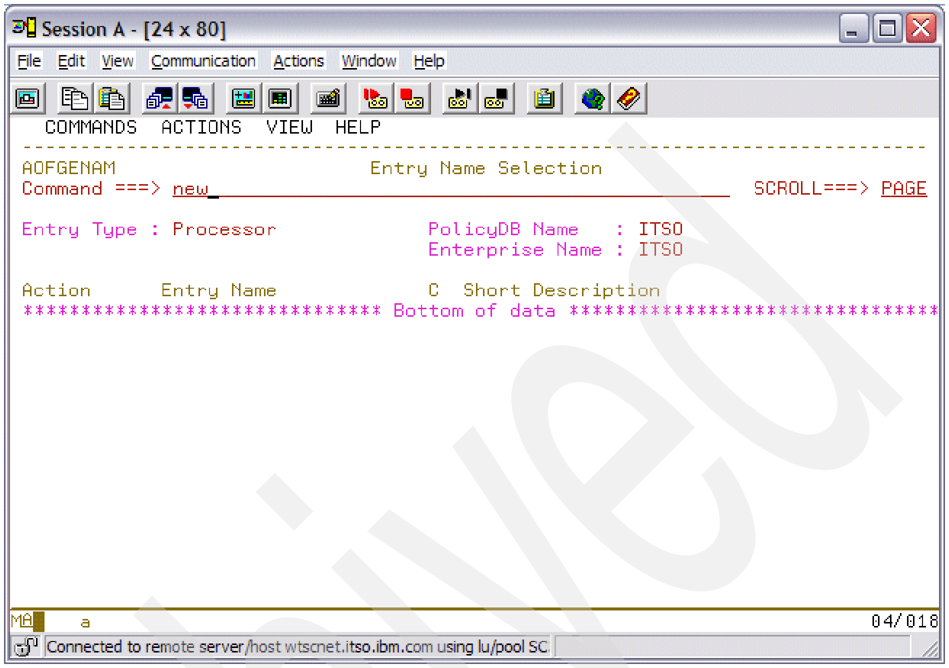
Figure 5-2 shows the automation policy entry type selection to define and modify automation policy entry.



```
  MENU  HELP
 ------------------------------------------------------------------------------
 AOFGETYP                      Entry Type Selection
 Option ===> 10_

   1 ENT      Enterprise                       30 TMR      Timers
   2 GRP      Group                            31 TMO      Timeout Settings
   3 SBG      SubGroup                         32 TPA      Tape Attendance
   4 SYS      System                (*)        33 MVC      MVS Component
   5 APG      ApplicationGroup      (*)        34 MDF      MVSCOMP Defaults
   6 APL      Application           (*)        35 SDF      System Defaults
   7 EVT      Events                           36 ADF      Application Defaults
   8 SVP      Service Periods                  37 AOP      Auto Operators
   9 TRG      Triggers                         38 NFY      Notify Operators
  10 PRO      Processor                        39 NTW      Network
  11 MTR      Monitor Resource      (*)        40 NNT      NNT Sessions
                                               41 RES      Resident CLISTs
  20 PRD      Product Automation               42 SCR      Status Details

                                               99 UET      User E-T Pairs
                        (*) Multi-User-Capable

MA     a                                                                 04/016
Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC
```

Figure 5-2 Entry Type Selection

To work with processor entry, select option 10. The result is shown in Figure 5-3.

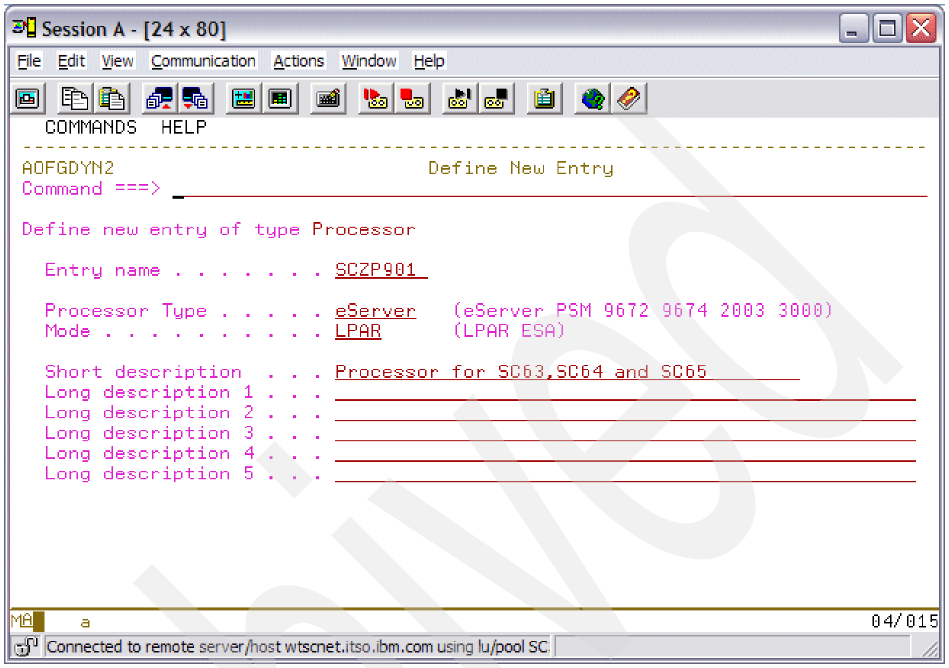


```
  COMMANDS  ACTIONS  VIEW  HELP
------------------------------------------------------------------------------
AOFGENAM                       Entry Name Selection
Command ===> new_                                            SCROLL===> PAGE

Entry Type : Processor               PolicyDB Name   : ITSO
                                     Enterprise Name : ITSO

Action       Entry Name              C  Short Description
******************************* Bottom of data ********************************
```

```
MA      a                                                               04/018
Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC
```

Figure 5-3 Entry Name Selection

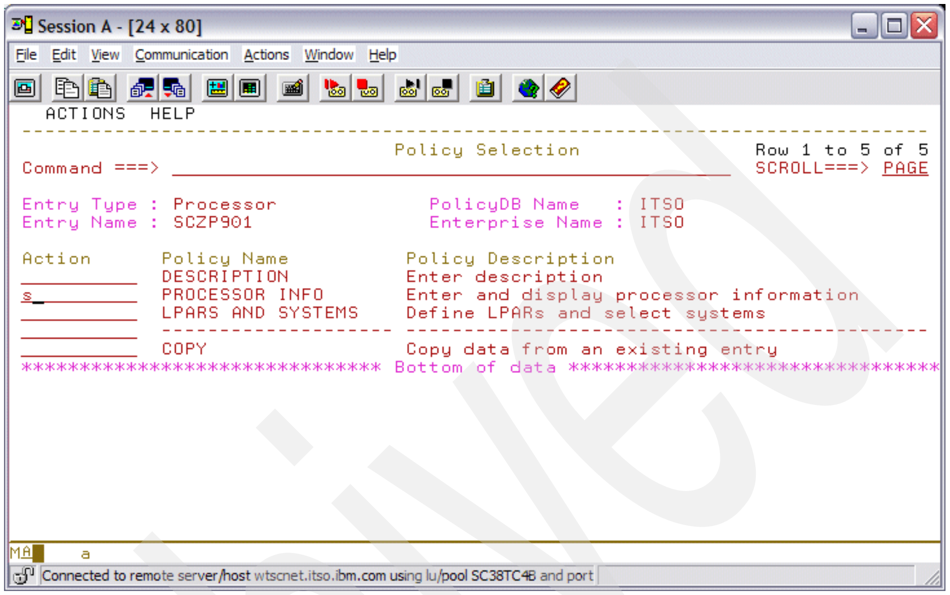
In the screen shown in Figure 5-3 on page 103, enter `NEW` to add a new processor definition. Figure 5-4 shows an example of the processor entry definition.



```
  COMMANDS   HELP
 ------------------------------------------------------------------------------
 AOFGDYN2                          Define New Entry
 Command ===> _

 Define new entry of type Processor

   Entry name . . . . . . . . SCZP901

   Processor Type . . . . . . eServer    (eServer PSM 9672 9674 2003 3000)
   Mode . . . . . . . . . . . LPAR       (LPAR ESA)

   Short description   . . . Processor for SC63,SC64 and SC65
   Long description 1 . . . .
   Long description 2 . . . .
   Long description 3 . . . .
   Long description 4 . . . .
   Long description 5 . . . .

MA      a                                                                 04/015
Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC
```

Figure 5-4   Define New Entry

We recommend using the hardware configuration definition (HCD) names of your processors as your processor names in System Automation. Press PF3 to exit and save. This brings you to the policy selection dialog shown in Figure 5-5.

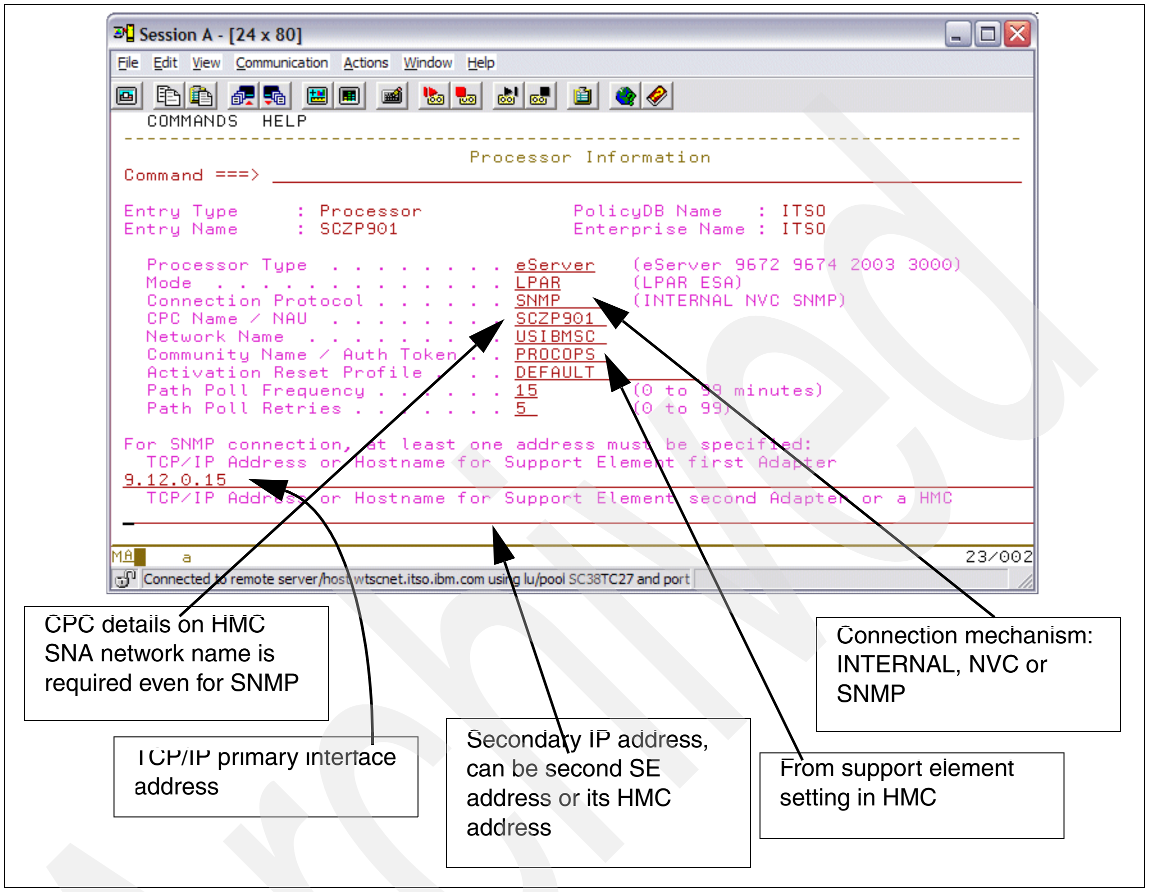


```
  ACTIONS  HELP
 ------------------------------------------------------------------------------
                                Policy Selection                 Row 1 to 5 of 5
 Command ===> ________________________________________________  SCROLL===> PAGE

 Entry Type : Processor                PolicyDB Name    : ITSO
 Entry Name : SCZP901                  Enterprise Name  : ITSO

 Action      Policy Name             Policy Description
 _________   DESCRIPTION             Enter description
 s________   PROCESSOR INFO          Enter and display processor information
 _________   LPARS AND SYSTEMS       Define LPARs and select systems
 _________   --------------------    ----------------------------------------
 _________   COPY                    Copy data from an existing entry
 ******************************* Bottom of data ********************************
```

```
MA      a
Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC38TC4B and port
```

Figure 5-5 Policy Selection - Processor Info

In Figure 5-5 on page 105, select the PROCESSOR INFO policy and define your processor information, as shown in Figure 5-6.

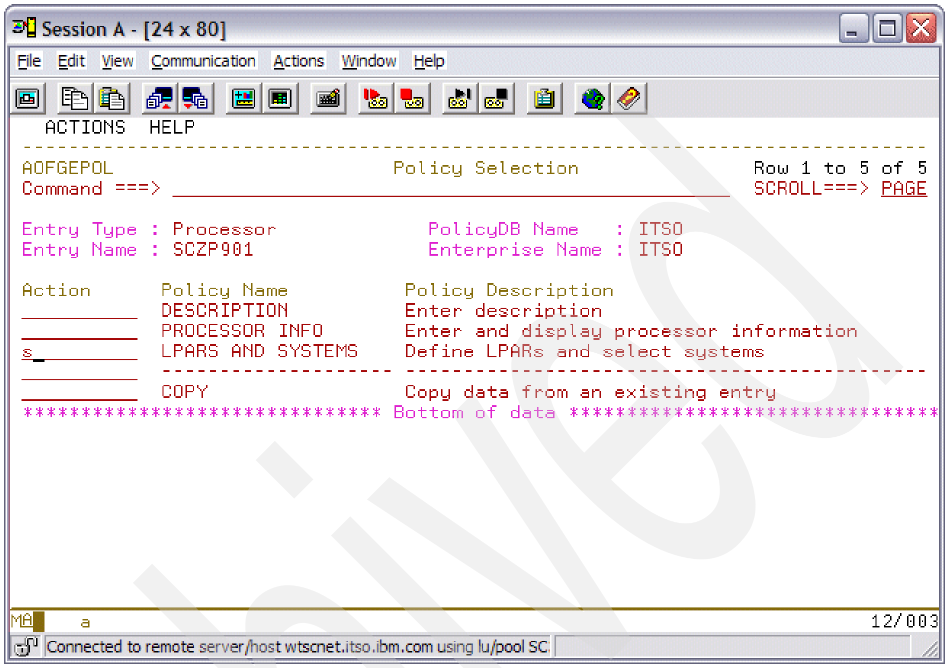


```
  COMMANDS   HELP
------------------------------------------------------------------------------
                              Processor Information
Command ===> ___________________________________________________________________

Entry Type      : Processor              PolicyDB Name    : ITSO
Entry Name      : SCZP901                Enterprise Name : ITSO

  Processor Type  . . . . . . . . eServer   (eServer 9672 9674 2003 3000)
  Mode  . . . . . . . . . . . . . LPAR      (LPAR ESA)
  Connection Protocol . . . . . . SNMP      (INTERNAL NVC SNMP)
  CPC Name / NAU  . . . . . . . . SCZP901
  Network Name  . . . . . . . . . USIBMSC
  Community Name / Auth Token . . PROCOPS
  Activation Reset Profile  . . . DEFAULT
  Path Poll Frequency . . . . . . 15        (0 to 99 minutes)
  Path Poll Retries . . . . . . . 5         (0 to 99)

For SNMP connection, at least one address must be specified:
  TCP/IP Address or Hostname for Support Element first Adapter
9.12.0.15
  TCP/IP Address or Hostname for Support Element second Adapter or a HMC

MA  a                                                                   23/002
Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC38TC27 and port
```

CPC details on HMC SNA network name is required even for SNMP
TCP/IP primary interface address
Secondary IP address, can be second SE address or its HMC address
From support element setting in HMC
Connection mechanism: INTERNAL, NVC or SNMP

Figure 5-6 Processor Information

Save the definition in Figure 5-6 on page 106. Back in the policy selection dialog, select the LPARS AND SYSTEMS policy, as shown in Figure 5-7.

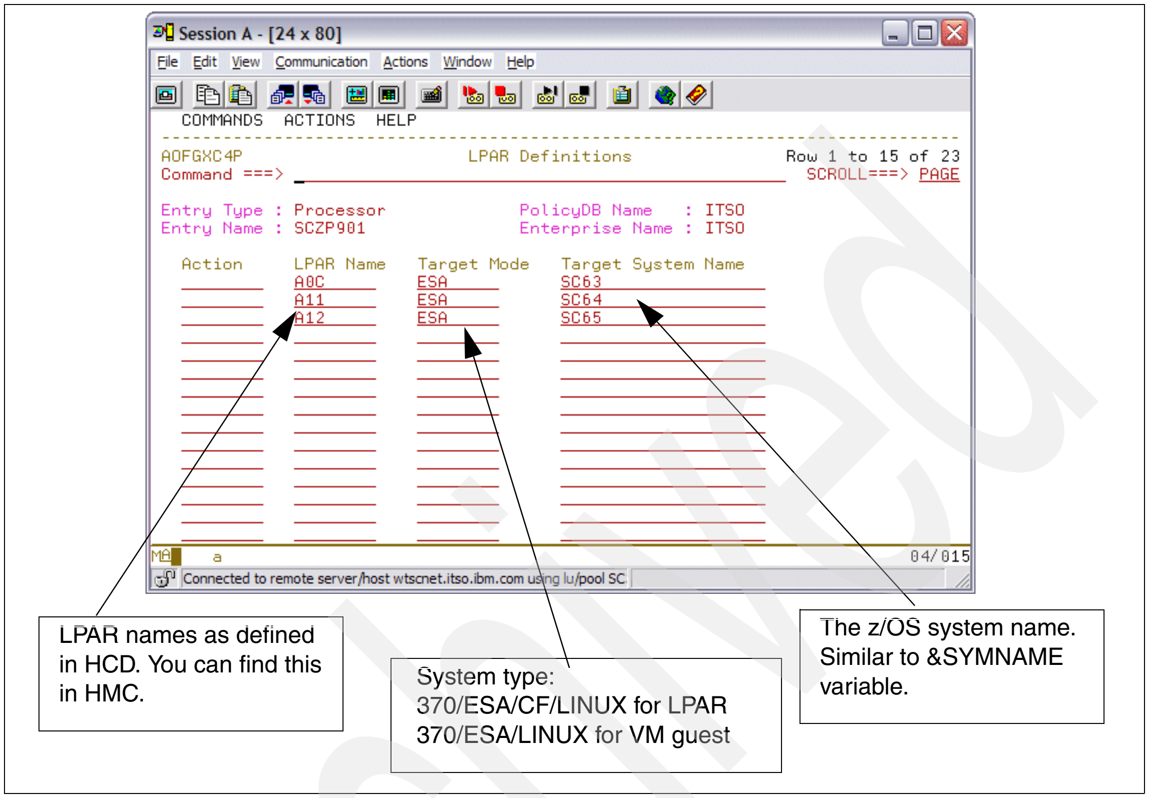


```
  ACTIONS  HELP
------------------------------------------------------------------------------
AOFGEPOL                        Policy Selection                Row 1 to 5 of 5
Command ===> ________________________________________________  SCROLL===> PAGE

Entry Type : Processor                 PolicyDB Name    : ITSO
Entry Name : SCZP901                   Enterprise Name  : ITSO

Action       Policy Name               Policy Description
             DESCRIPTION               Enter description
__________   PROCESSOR INFO            Enter and display processor information
s_________   LPARS AND SYSTEMS         Define LPARs and select systems
__________   --------------------      ----------------------------------------
__________   COPY                      Copy data from an existing entry
******************************* Bottom of data ********************************
```

```
MA      a                                                                 12/003
Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC
```

Figure 5-7 Policy Selection - LPAR and SYSTEMS

The policy definition for LPARs and Systems is shown in Figure 5-8.


Session A - [24 x 80]
File Edit View Communication Actions Window Help

```
  COMMANDS  ACTIONS  HELP
------------------------------------------------------------------------------
AOFGXC4P                    LPAR Definitions                  Row 1 to 15 of 23
Command ===> _                                               SCROLL===> PAGE

Entry Type : Processor                   PolicyDB Name   : ITSO
Entry Name : SCZP901                     Enterprise Name : ITSO

  Action     LPAR Name   Target Mode   Target System Name
             A0C         ESA           SC63
             A11         ESA           SC64
             A12         ESA           SC65
```

```
MA     a                                                                04/015
Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC
```

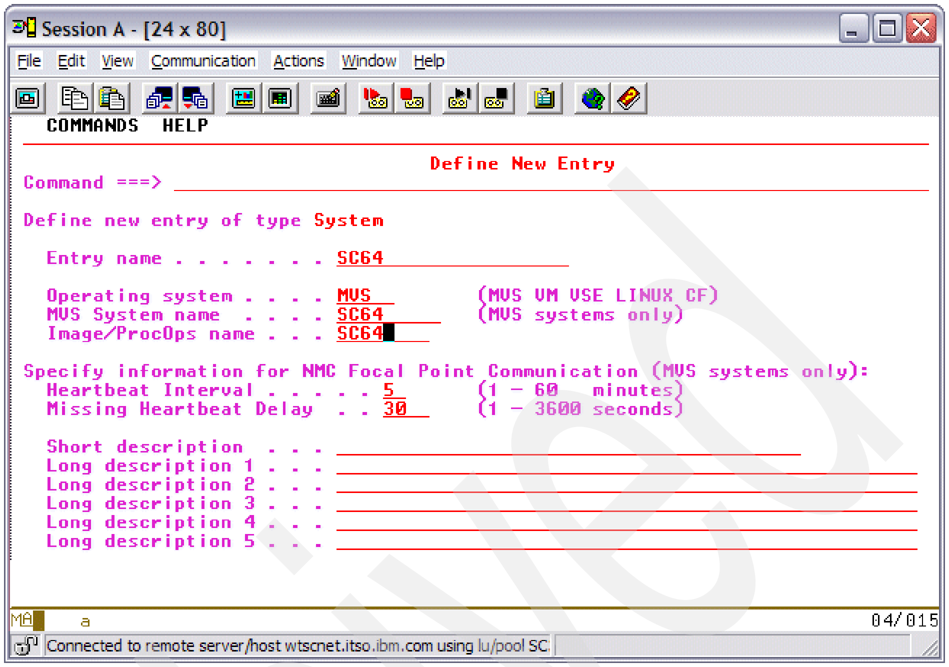
LPAR names as defined in HCD. You can find this in HMC.
System type: 370/ESA/CF/LINUX for LPAR 370/ESA/LINUX for VM guest
The z/OS system name. Similar to &SYMNAME variable.

Figure 5-8 LPAR Definition

You can now save the new processor definition. Press F3 to save and exit the processor definition.

5.3.2 Connecting System to Processor

This section shows the system definition that is connected to the processor entry. All Processor Operations target systems, including coupling facilities, must be defined using Automation Policy as a System object using the Dialog option 4 (SYS). In the dialog 4 from the Entry type Selection dialog shown in Figure 5-2 on page 102, the system definition can be created by issuing the NEW command from the entry name selection dialog. The new system entry dialog is shown in Figure 5-9 on page 109.



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

Define new entry of type System

  Entry name . . . . . . . . SC64_______________

  Operating system . . . . . MVS__       (MVS VM VSE LINUX CF)
  MVS System name  . . . . . SC64____    (MVS systems only)
  Image/ProcOps name . . . . SC64____

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

  Short description  . . . ________________________________________
  Long description 1 . . . __________________________________________________
  Long description 2 . . . __________________________________________________
  Long description 3 . . . __________________________________________________
  Long description 4 . . . __________________________________________________
  Long description 5 . . . __________________________________________________


MA      a                                                               04/015
Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC
```

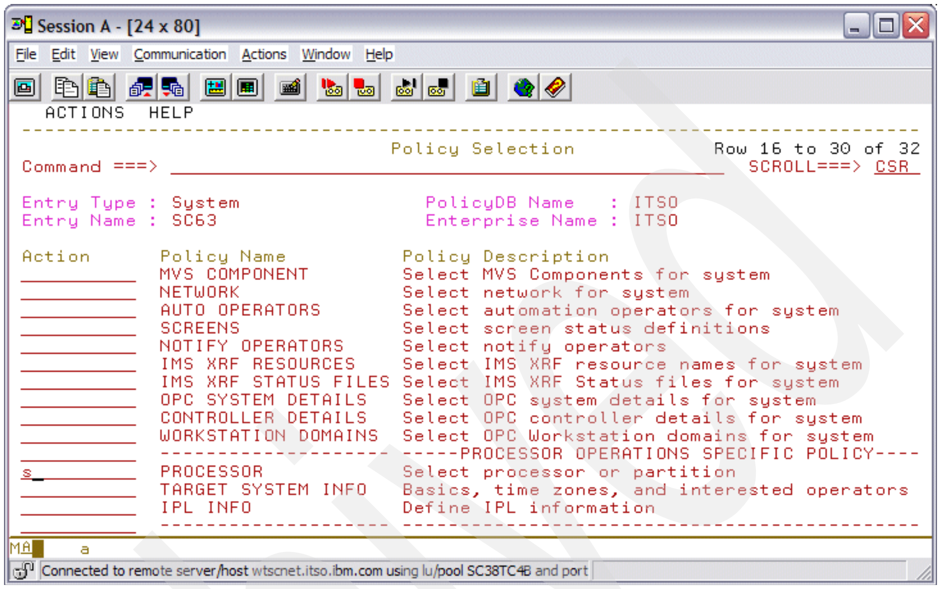
Figure 5-9 System definition

As shown in Figure 5-9, a system may has several name identifiers that may or may not be the same, such as:

- System Automation for z/OS entry name
- Actual z/OS system name
- Processor Operations name

To avoid confusion, all these names should be defined consistently. If possible, use the same name for all these entries. Press PF3 to save this.

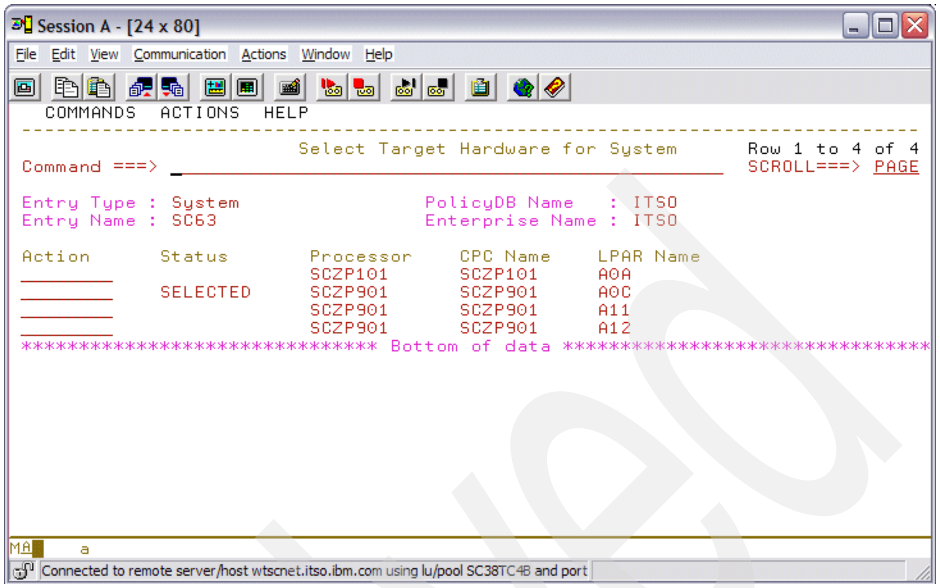
After having specified target system names and operating system type, select PROCESSOR policy to choose a previously defined processor for your system, as shown in Figure 5-10.



```
  ACTIONS  HELP
------------------------------------------------------------------------------
                             Policy Selection                Row 16 to 30 of 32
Command ===> _________________________________________________  SCROLL===> CSR

Entry Type : System                    PolicyDB Name    : ITSO
Entry Name : SC63                      Enterprise Name  : ITSO

Action      Policy Name               Policy Description
            MVS COMPONENT             Select MVS Components for system
            NETWORK                   Select network for system
            AUTO OPERATORS            Select automation operators for system
            SCREENS                   Select screen status definitions
            NOTIFY OPERATORS          Select notify operators
            IMS XRF RESOURCES         Select IMS XRF resource names for system
            IMS XRF STATUS FILES      Select IMS XRF Status files for system
            OPC SYSTEM DETAILS        Select OPC system details for system
            CONTROLLER DETAILS        Select OPC controller details for system
            WORKSTATION DOMAINS       Select OPC Workstation domains for system
            --------------------      -----PROCESSOR OPERATIONS SPECIFIC POLICY----
s           PROCESSOR                 Select processor or partition
            TARGET SYSTEM INFO        Basics, time zones, and interested operators
            IPL INFO                  Define IPL information
            --------------------      ---------------------------------------------
MA     a
Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC38TC4B and port
```

Figure 5-10   Policy Selection - Processor

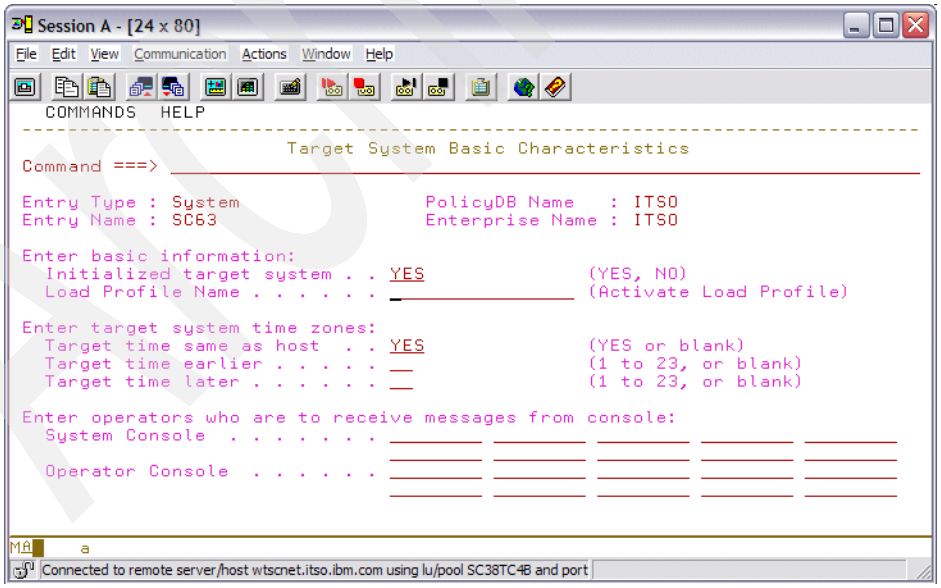
You will find that your system has already selected a processor, if the dialog finds that a processor has your system name in its LPAR and system name list (Figure 5-7 on page 107). If that is not the case, you must select a processor from the list in Figure 5-11 on page 111.



```
  COMMANDS  ACTIONS  HELP
 ------------------------------------------------------------------------------
                        Select Target Hardware for System       Row 1 to 4 of 4
 Command ===> _                                                 SCROLL===> PAGE

 Entry Type : System                  PolicyDB Name   : ITSO
 Entry Name : SC63                    Enterprise Name : ITSO

 Action      Status      Processor    CPC Name    LPAR Name
                         SCZP101      SCZP101     A0A
 ________    SELECTED    SCZP901      SCZP901     A0C
 ________                SCZP901      SCZP901     A11
 ________                SCZP901      SCZP901     A12
 ******************************* Bottom of data ********************************

MA      a
Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC38TC4B and port
```

Figure 5-11   Select Target Hardware for System

Select the TARGET SYSTEM INFO policy from the Policy Selection dialog, similar to Figure 5-10 on page 110.



```
  COMMANDS  HELP
 ------------------------------------------------------------------------------
                       Target System Basic Characteristics
 Command ===> __________________________________________________________________

 Entry Type : System                  PolicyDB Name   : ITSO
 Entry Name : SC63                    Enterprise Name : ITSO

 Enter basic information:
   Initialized target system . . YES              (YES, NO)
   Load Profile Name . . . . . . _________        (Activate Load Profile)

 Enter target system time zones:
   Target time same as host  . . YES              (YES or blank)
   Target time earlier . . . . . __               (1 to 23, or blank)
   Target time later . . . . . . __               (1 to 23, or blank)

 Enter operators who are to receive messages from console:
   System Console  . . . . . . . ________ ________ ________ ________ ________
                                 ________ ________ ________ ________ ________
   Operator Console  . . . . . . ________ ________ ________ ________ ________
                                 ________ ________ ________ ________ ________

MA      a
Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC38TC4B and port
```

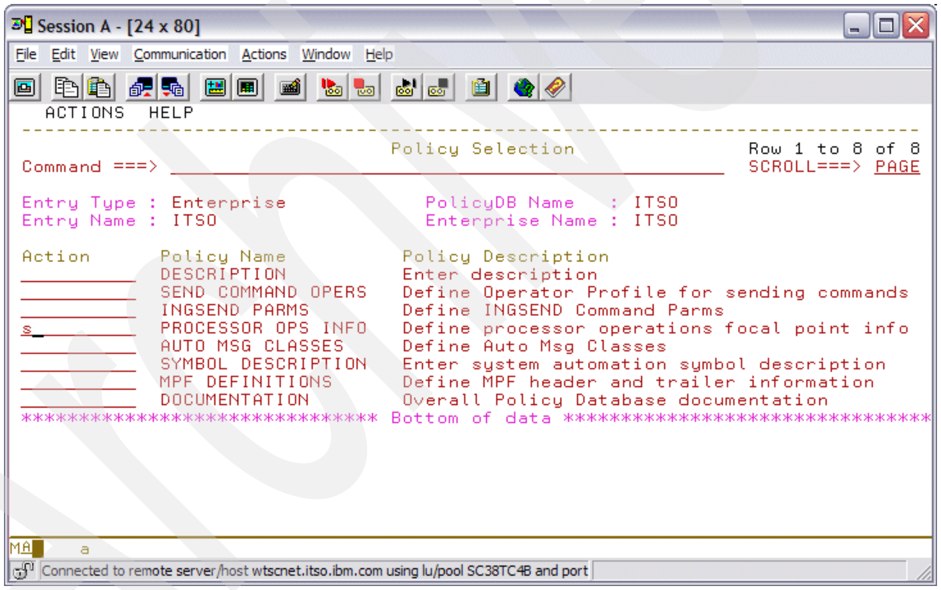
Figure 5-12   MVS Target System Basic Characteristics

Decide here if you want Processor Operations to automatically start its connection to this system when it is started. If the Processor Operations FP system runs in the same time zone as the target system, select it or otherwise define the time difference. If you want to predefine operators to receive messages from this system, fill in the interested operator lists.

Note: Processor Operations has the ISQXDST command, which lets you manage this list more dynamically. This command is invoked from the NetView session.

5.3.3 Enterprise Processor Operations information

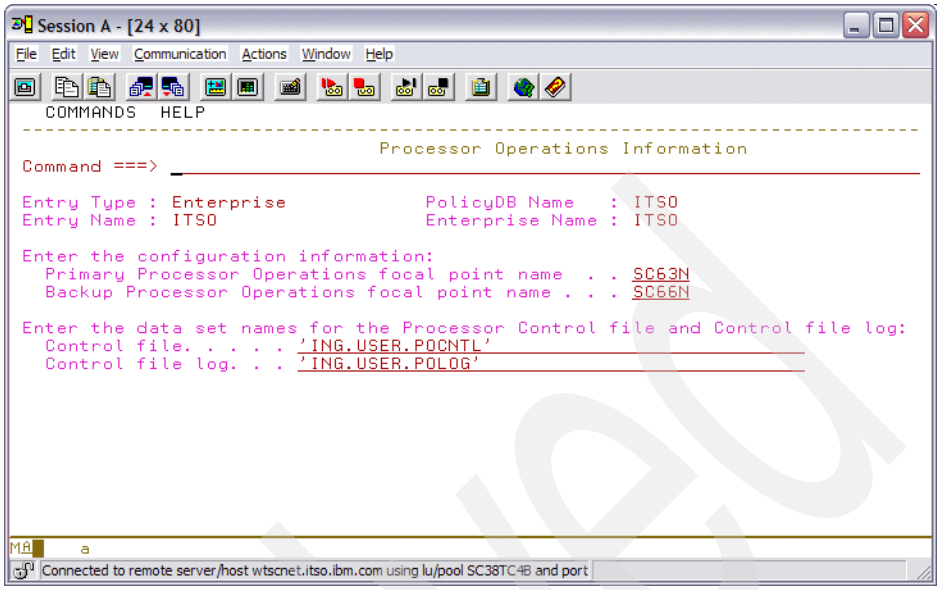
From the Enterprise definition in your automation policy (selection 1 from the Entry Type Selection dialog), you must define the Processor Operation information policy, as shown in Figure 5-13.



```
  ACTIONS  HELP
 ------------------------------------------------------------------------------
                                Policy Selection                Row 1 to 8 of 8
 Command ===> ________________________________________________ SCROLL===> PAGE

 Entry Type : Enterprise              PolicyDB Name    : ITSO
 Entry Name : ITSO                    Enterprise Name : ITSO

 Action      Policy Name            Policy Description
 __________  DESCRIPTION            Enter description
 __________  SEND COMMAND OPERS     Define Operator Profile for sending commands
 __________  INGSEND PARMS          Define INGSEND Command Parms
 s_________  PROCESSOR OPS INFO     Define processor operations focal point info
 __________  AUTO MSG CLASSES       Define Auto Msg Classes
 __________  SYMBOL DESCRIPTION     Enter system automation symbol description
 __________  MPF DEFINITIONS        Define MPF header and trailer information
 __________  DOCUMENTATION          Overall Policy Database documentation
 ******************************* Bottom of data ********************************

MA     a
Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC38TC4B and port
```

Figure 5-13 Policy Selection - Processor OPS Info

In the Processor Operations information policy in Figure 5-14 on page 113, define the System Automation for z/OS domain names of your System Automation for z/OS that you use as the primary and backup focal points.



```
  COMMANDS  HELP
------------------------------------------------------------------------------
                              Processor Operations Information
Command ===> _

Entry Type : Enterprise               PolicyDB Name    : ITSO
Entry Name : ITSO                     Enterprise Name  : ITSO

Enter the configuration information:
  Primary Processor Operations focal point name  . . SC63N
  Backup Processor Operations focal point name . . . SC66N

Enter the data set names for the Processor Control file and Control file log:
  Control file. . . . . 'ING.USER.POCNTL'
  Control file log. . . 'ING.USER.POLOG'
```

MA a
Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC38TC4B and port

Figure 5-14 Processor Operation Information

The control file and control file log datasets (ISQCNTL and ISQLOG) are used during the System Automation for z/OS Processor Operations build. You can use the data sets ING.USER.POCNTL and ING.USER.POLOG that you create in 2.3, “Creating System Automation for z/OS dialog screens” on page 21.

5.4 Preparing the hardware

For each CPC support element, verify that the System Automation for z/OS Processor Operations hardware requirements, such as microcode level is met. Refer to *IBM Tivoli System Automation for z/OS V3.1 Planning and Installation*, SC33-8261.

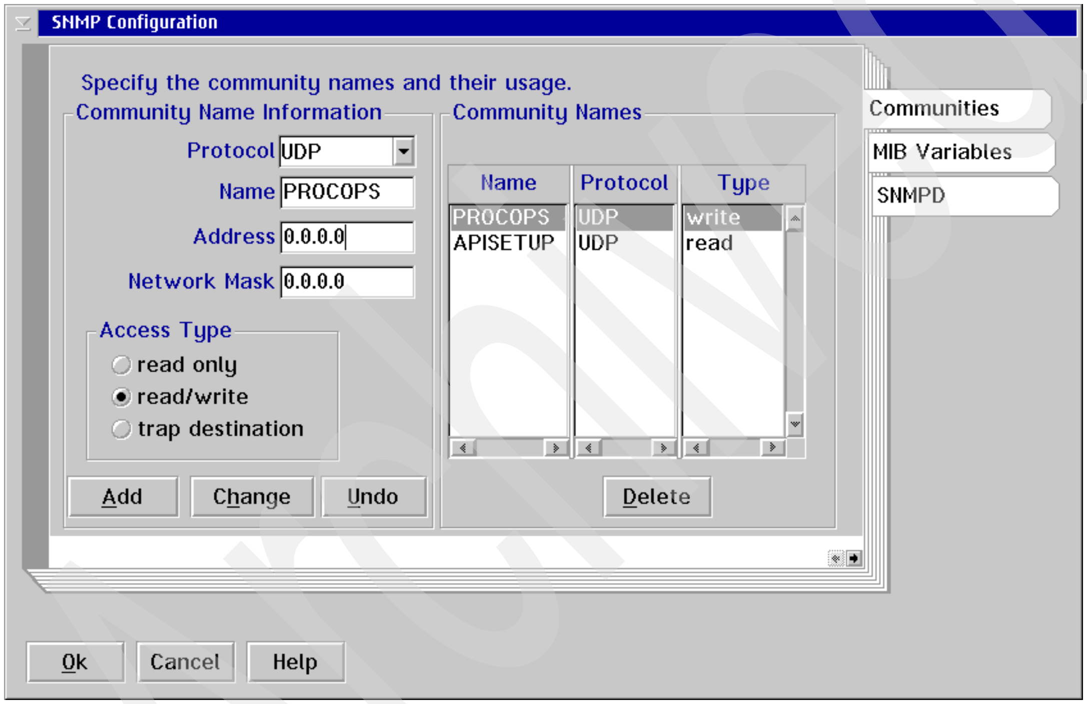
5.4.1 Preparing the support element

For each support element or HMC machine, perform the required customization task to activate the SNMP communication and the z900 API required for this communication, as described in Chapter 9, “Installing SA z/OS on Host Systems“, of the *IBM Tivoli System Automation for z/OS V3.1 Planning and Installation*, SC33-8261.

During customization, you must specify the SNMP community names; these names are used for the System Automation for z/OS dialog processor definitions.

Community names have to be specified in order to use the BCP internal interface transport, the TCP/IP SNMP transport for Processor Operations, or both. For this task, you need to be logged on in Access Administrator mode on your CPC's Support Element. If the Support Element has multiple network adapters, the SNMP API must be defined to use adapter 0 (primary network adapter), even if that adapter is not being used for network connection.

We use two community names, as suggested by the installation manual, one for API access with a read only access and the other with a write access for Processor Operations. The PROCOPTS community definition is shown in Figure 5-15.



Specify the community names and their usage.

Community Name Information

Protocol: UDP

Name: PROCOPS

Address: 0.0.0.0

Network Mask: 0.0.0.0

Access Type: read only [✓ read/write] trap destination

Community Names

Name	Protocol	Type
PROCOPS	UDP	write
APISETUP	UDP	read

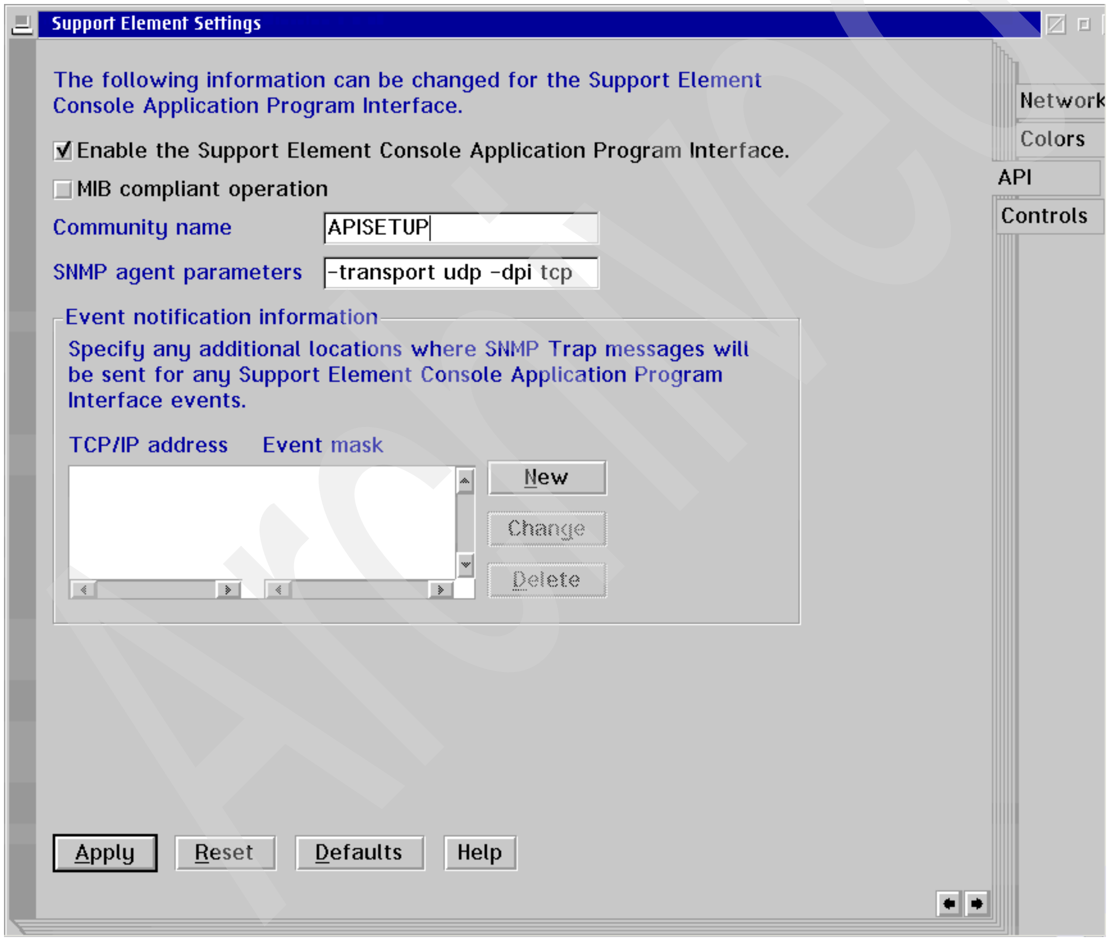
Figure 5-15 SNMP definitions for the API

For additional SNMP and API configuration information, refer to Chapter 6, "Configuring the Data Exchange APIs", in *zSeries 900 Application Programming Interface*, SB10-7030.

5.4.2 Enable the API and set the community name

In order to use the BCP internal interface or the SNMP interface, the Support Element API function needs to be enabled. This is achieved using the API tab of the Support Element Settings notebook window, as shown in Figure 5-16. Be sure to use the appropriate SNMP community name.

Note: This is only valid for an OS/2® based Support Element. The settings must be left blank for the new Linux-based HMCs.


Support Element Settings

The following information can be changed for the Support Element Console Application Program Interface.

[✓] Enable the Support Element Console Application Program Interface.

[ ] MIB compliant operation

Community name: APISETUP

SNMP agent parameters: -transport udp -dpi tcp

Event notification information

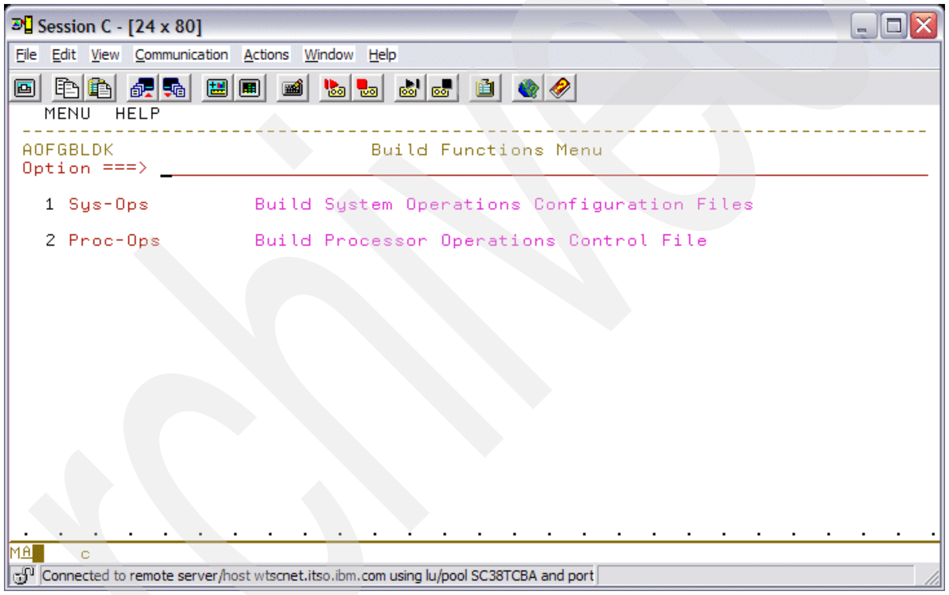
Specify any additional locations where SNMP Trap messages will be sent for any Support Element Console Application Program Interface events.

Figure 5-16 SNMP definitions for the enable API

5.5 Processor Operations build facility

This section describes how to build the Processor Operations definitions. The System Automation for z/OS processor control file contains information specific to your System Automation for z/OS configuration and is used to start Processor Operations. The control file is built from informtion about the Enterprise, Groups, SubGroups, Systems, Processors, and Communication Tasks already entered in the Policy database. Any change to these definitions requires the control file to be rebuilt.

From the Policy database selection screen, enter BUILD next to the PolicyDB. Figure 5-17 appears, which allows you to build the Proc-Ops control file.

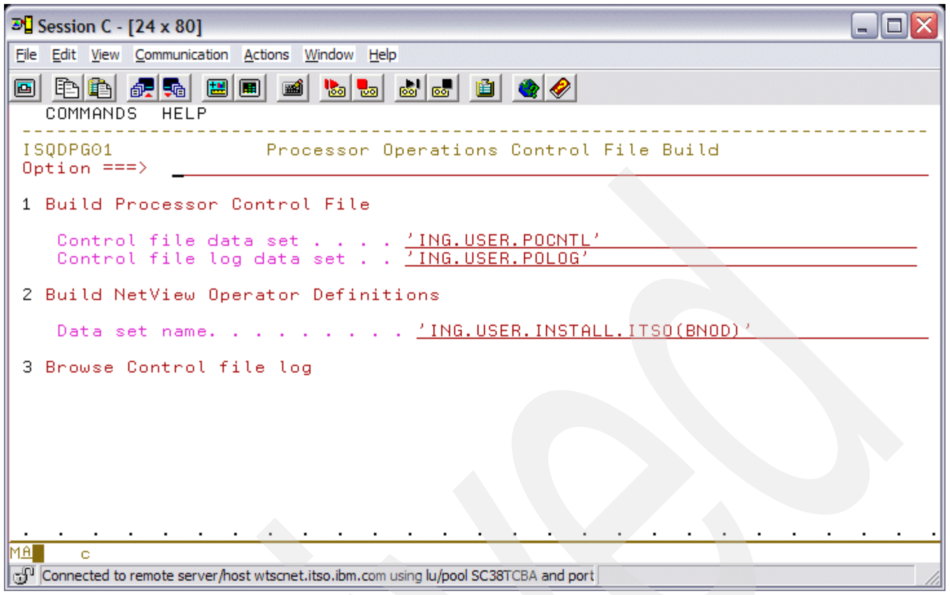


```
  MENU  HELP
 ------------------------------------------------------------------------------
 AOFGBLDK                          Build Functions Menu
 Option ===> _

   1  Sys-Ops          Build System Operations Configuration Files

   2  Proc-Ops         Build Processor Operations Control File
```

```
MA       c
Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC38TCBA and port
```

Figure 5-17 Build Functions Menu

From the Build Functions Menu, select option 2. Figure 5-18 on page 117 shows an example of the screen ISQDPG01 that is displayed. If the control file and control file log were previously specified, they will automatically appear here; otherwise, they will need to be entered.



```
  COMMANDS  HELP
------------------------------------------------------------------------------
ISQDPG01             Processor Operations Control File Build
Option ===> _

1 Build Processor Control File

   Control file data set . . . . 'ING.USER.POCNTL'
   Control file log data set . . 'ING.USER.POLOG'

2 Build NetView Operator Definitions

   Data set name. . . . . . . . . 'ING.USER.INSTALL.ITSO(BNOD)'

3 Browse Control file log
```

```
MA      c
Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC38TCBA and port
```

Figure 5-18 Processor Operations Control File Build

Select option 1 to generate the control file. If any errors are detected, they will be recorded in the control file log.

Select option 2 to build a sample NetView operator definitions file. The file can be used as an include for the NetView DSIOPF member. The operators defined in this member would be various System Automation for z/OS automated task user IDs. The number of target control task operators (ISQCMxxx) should match the number of current processors that are defined with a connection protocol other than INTERNAL. The number of message monitor task operators (ISQBTxxx) should also match the number of target control tasks.

Option 3 will enable you to browse this data set to determine what errors, if any, have occurred. Press PF3 to exit and save.

Important: The following applies:

- Proc-Ops Build will only select entries with connection protocol NVC and SNMP.
- Sys-Ops Build will only select entries with connection type INTERNAL.

5.6 Starting and stopping Processor Operations

Processor Operations can either be started manually using operator commands in System Automation for z/OS, or automatically through definitions in the automation policy.

5.6.1 Manual operation of Processor Operations

Important: Even though it is possible to have Processor Operations started on multiple systems at any one time, only the focal point system will be communicating to the Support Elements in the enterprise.

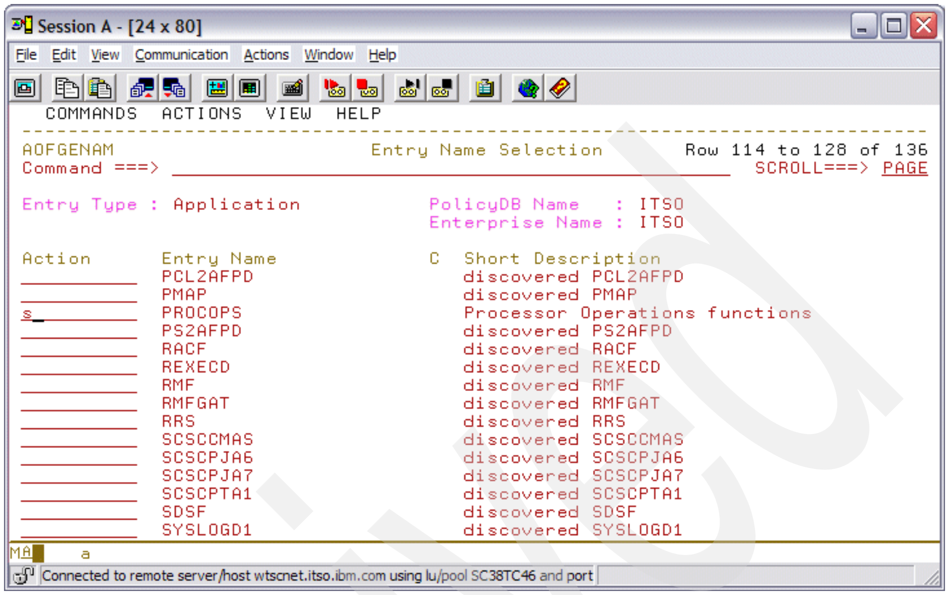
The ISQSTART <controldsn> command starts Processor Operations. It establishes the processor operations environment with System Automation for z/OS. When this command is entered on the NetView Control Command Facility (NCCF) screen, you must specify, as a parameter, the control file where you built the Processor Operations definitions. See Figure 5-18 on page 117.

You can use the ISQSTOP command to stop Processor Operations from NCCF.

5.6.2 Automating Processor Operations

You can automate the start of Processor Operations using System Automation for z/OS policy to start when System Automation for z/OS is started. A sample Processor Operations application is provided. You can add the *PROCOPS sample database into your policy database. This provides a new application definition under option 6 from the Entry Name Selection dialog.

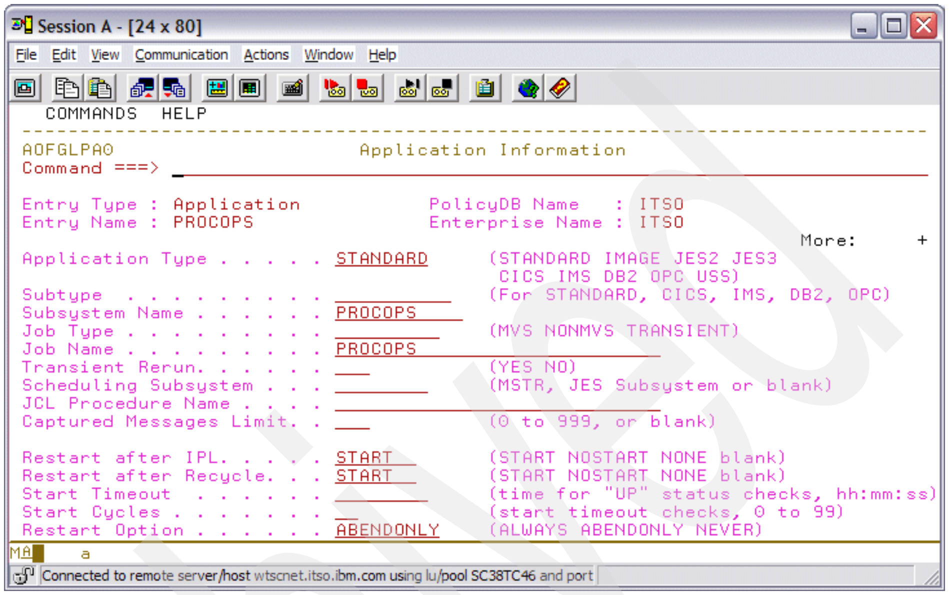
Figure 5-19 shows the PROCOPS applications that have already been defined.



```
  COMMANDS  ACTIONS  VIEW  HELP
------------------------------------------------------------------------------
AOFGENAM                      Entry Name Selection          Row 114 to 128 of 136
Command ===> ________________________________________________      SCROLL===> PAGE

Entry Type : Application                 PolicyDB Name   : ITSO
                                         Enterprise Name : ITSO

Action       Entry Name                  C   Short Description
__________   PCL2AFPD                        discovered PCL2AFPD
__________   PMAP                            discovered PMAP
s_________   PROCOPS                         Processor Operations functions
__________   PS2AFPD                         discovered PS2AFPD
__________   RACF                            discovered RACF
__________   REXECD                          discovered REXECD
__________   RMF                             discovered RMF
__________   RMFGAT                          discovered RMFGAT
__________   RRS                             discovered RRS
__________   SCSCCMAS                        discovered SCSCCMAS
__________   SCSCPJA6                        discovered SCSCPJA6
__________   SCSCPJA7                        discovered SCSCPJA7
__________   SCSCPTA1                        discovered SCSCPTA1
__________   SDSF                            discovered SDSF
__________   SYSLOGD1                        discovered SYSLOGD1
MA      a
Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC38TC46 and port
```

Figure 5-19 Entry Name selection - Application

When you select this application and then select AUTOMATION INFO policy, you will get the screen shown in Figure 5-20. Enter YES for Start on IPL and Start on Recycle.

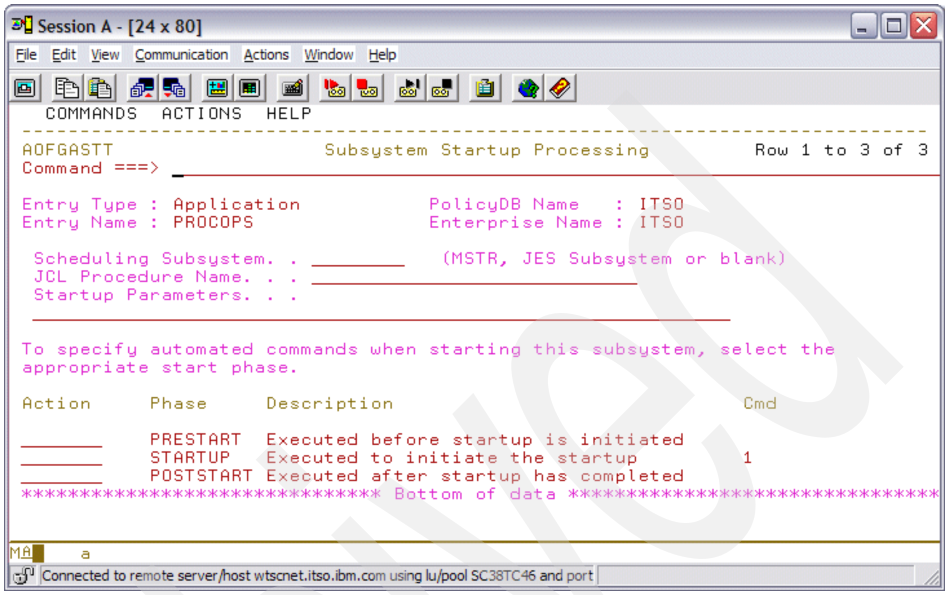


```
  COMMANDS  HELP
------------------------------------------------------------------------------
AOFGLPA0                      Application Information
Command ===> _
                                                                               
Entry Type : Application               PolicyDB Name    : ITSO
Entry Name : PROCOPS                   Enterprise Name  : ITSO
                                                                More:      +
Application Type . . . . . STANDARD      (STANDARD IMAGE JES2 JES3
                                          CICS IMS DB2 OPC USS)
Subtype  . . . . . . . . . ________      (For STANDARD, CICS, IMS, DB2, OPC)
Subsystem Name . . . . . . PROCOPS
Job Type . . . . . . . . . ________      (MVS NONMVS TRANSIENT)
Job Name . . . . . . . . . PROCOPS
Transient Rerun. . . . . . ___           (YES NO)
Scheduling Subsystem . . . ________      (MSTR, JES Subsystem or blank)
JCL Procedure Name . . . . ____________________________
Captured Messages Limit. . ___           (0 to 999, or blank)

Restart after IPL. . . . . START         (START NOSTART NONE blank)
Restart after Recycle. . . START         (START NOSTART NONE blank)
Start Timeout  . . . . . . ________      (time for "UP" status checks, hh:mm:ss)
Start Cycles . . . . . . . __            (start timeout checks, 0 to 99)
Restart Option . . . . . . ABENDONLY     (ALWAYS ABENDONLY NEVER)
MA      a
```

Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC38TC46 and port

Figure 5-20 Application Information

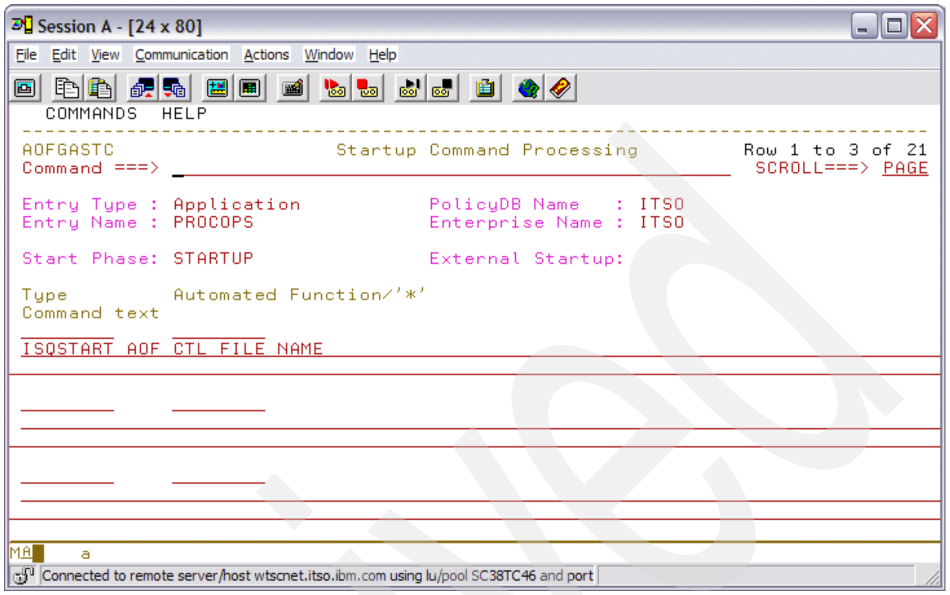
Press PF3 to exit and save. Now select the STARTUP policy in the policy selection screen. Figure 5-21 shows the startup policy.



```
  COMMANDS  ACTIONS  HELP
------------------------------------------------------------------------------
AOFGASTT                    Subsystem Startup Processing          Row 1 to 3 of 3
Command ===> _

Entry Type : Application                 PolicyDB Name   : ITSO
Entry Name : PROCOPS                     Enterprise Name : ITSO

 Scheduling Subsystem. . ________   (MSTR, JES Subsystem or blank)
 JCL Procedure Name. . . ____________________________
 Startup Parameters. . .
 ____________________________________________________________

To specify automated commands when starting this subsystem, select the
appropriate start phase.

Action     Phase     Description                                 Cmd

_______    PRESTART  Executed before startup is initiated
_______    STARTUP   Executed to initiate the startup           1
_______    POSTSTART Executed after startup has completed
******************************* Bottom of data ********************************

MA       a
Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC38TC46 and port
```

Figure 5-21 Subsystem Startup Processing

When you select STARTUP command, Figure 5-22 shows the start command.


Session A - [24 x 80]
File Edit View Communication Actions Window Help

```
 COMMANDS  HELP
------------------------------------------------------------------------------
AOFGASTC                    Startup Command Processing           Row 1 to 3 of 21
Command ===> _                                                   SCROLL===> PAGE

Entry Type : Application               PolicyDB Name    : ITSO
Entry Name : PROCOPS                   Enterprise Name  : ITSO

Start Phase: STARTUP                   External Startup:

Type         Automated Function/'*'
Command text

ISQSTART AOF CTL FILE NAME

MA      a
```

Connected to remote server/host wtscnet.itso.ibm.com using lu/pool SC38TC46 and port

Figure 5-22 Startup Command Processing

When you do a BUILD for your Automation Control File (ACF), you will notice the message in the BUILD Report member $BLDRPT is similar to Example 5-1.

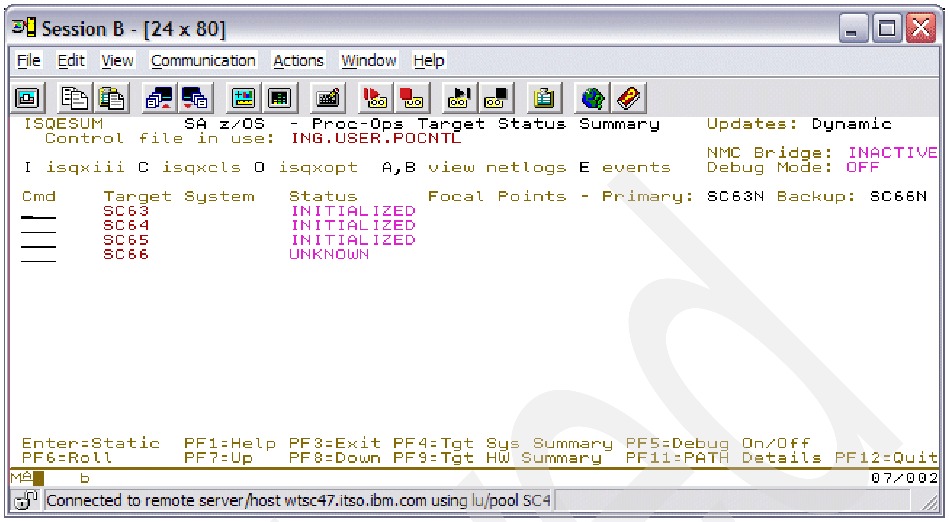
Example 5-1 Message for Processor Operations automation

```
Processor Control File "ING.USER.POCNTL" will be used for automated
startup of the PROCESSOR_OPERATIONS application
```

Later, when System Automation for z/OS is recycled or the Automation Control File (ACF) is reloaded, Processor Operations will be automatically started, monitored, and stopped, much the same as any other application defined to System Automation for z/OS.

5.6.3 Managing processors using Processor Operations

Once Processor Operations is started, enter the ISQXDST command on the NCCF screen to view the status of the Processor Operations managed systems. Figure 5-23 on page 123 shows the managed system status dialog.


Session B - [24 x 80]

File Edit View Communication Actions Window Help

```
ISQESUM       SA z/OS   - Proc-Ops Target Status Summary     Updates: Dynamic
  Control file in use: ING.USER.POCNTL
                                                             NMC Bridge: INACTIVE
I isqxiii C isqxcls O isqxopt  A,B view netlogs E events     Debug Mode: OFF

Cmd     Target System    Status      Focal Points - Primary: SC63N Backup: SC66N
___     SC63             INITIALIZED
___     SC64             INITIALIZED
___     SC65             INITIALIZED
___     SC66             UNKNOWN











Enter=Static  PF1=Help PF3=Exit PF4=Tgt Sys Summary PF5=Debug On/Off
PF6=Roll      PF7=Up   PF8=Down PF9=Tgt HW Summary  PF11=PATH Details PF12=Quit
MA      b                                                                 07/002
```

Connected to remote server/host wtsc47.itso.ibm.com using lu/pool SC4

Figure 5-23 Processor Operations Target Status Summary

If Processor Operations has not yet established communications with the target systems using the ISQXIII command, the status will be set to `UNKNOWN`, not `INITIALIZED`. Ensure that the indication for updates is set to `Dynamic` in the upper right corner of the dialog, which will allow the Status column to change dynamically. Pressing the Enter key will toggle between static and dynamic updates.

For a detailed list of all Processor Operations commands, see *IBM Tivoli System Automation for z/OS V3.1 Operator's Commands*, SC33-8265.

5.7 Security with RACF

This section discusses additional RACF security that can be implemented to ensure authorized access to the processor hardware layer. For processor operations SNMP connections and for the Parallel Sysplex enhancements functions that use the BCP internal interface, a security product such as RACF must be used to define the required resources and grant access to these resources for the authorized NetView users and autotasks.

5.7.1 Allowing NetView to use the BCP internal interface

Before you can use the enhanced sysplex functions of System Automation for z/OS for processor automation, access to the hardware resource must be authorized by the Security Access Facility (SAF). The resource is defined as a FACILITY class resource, hence the FACILITY class must be active. We also recommend that the FACILITY class to be RACLISTed to ensure checking performance. RACLISTed classes are cached in memory.

To enable the RACF's FACILITY class, you can run the command shown in Example 5-2.

Example 5-2 Command for enabling RACF's FACILITY class

```
SETROPTS CLASSACT(FACILITY)
SETROPTS RACLIST(FACILITY)
```

To define the HSAET32 hardware resource, run the RDEFINE FACILITY HSA.ET32OAN.HSAET32 UACC(NONE) command. The default access is NONE, so specific access must be given to the user that starts the System Automation for z/OS started task. The command to do that is PERMIT HSA.ET32OAN.HSAET32 CLASS(FACILITY) ID(stcuser) ACC(READ), assuming STCUSER is the user ID associated with your System Automation for z/OS started task.

Note: The FACILITY class profile names are predefined in System Automation for z/OS.

5.7.2 Access to the processor and LPAR

Each processor (CPC) defined in your System Automation for z/OS policy data base must have a corresponding resource profile defined with your SAF product. The format of the CPC resources are:

- HSA.ET32TGT.netid.nau

  The netid.nau part of the resource name corresponds with the netid.nau definition of the CPC entry specified in the customization dialog (see Figure 5-6 on page 106). The period between netid and nau is part of the resource name.

- HSA.ET32TGT.netid.nau.lpar

  For LPAR protection, define a resource with the netid.nau.lpar specification. The LPAR names are shown in Figure 5-8 on page 108.

To define a CPC resource in RACF, issue the RDEFINE FACILITY HSA.ET32TGT.USIBMSC.SCZP901 UACC(NONE) command.

The CPC with netid USIBMSC and nau SCZP901 is defined as a resource in the RACF class facility with a universal access attribute of NONE. You can use a wildcard character to specify that the resource be more generic if that is suitable for your environment.

Different users requires different access level to these resources. The following lists the access levels and their meaning for the CPC and LPAR resources:

- READ: Retrieve, get configuration information from the CPC
- WRITE: Update, set configuration information of the CPC
- CONTROL: Issue operations management commands of the CPC

5.7.3 Defining the CPC access lists

Several System Automation for z/OS autotasks need to be authorized to access the CPCs that are defined in the customization dialog. The following System Automation for z/OS w autotasks need to be authorized with access level CONTROL for all defined CPCs and all its LPARs:

- The XCF autotasks.
- The autotasks defined with `SYN %AOFOPXCFOPER%` in automation table member AOFMSGSY.
- The HW interface autotasks AUTHWxxx.
- The AUTXCFxx autotasks plus the additional ones from %AOFOPXCFOPER% are used internally once INGCF drain or INGCF enable is invoked by an authorized user. IXC102A message automation is also performed by these autotasks.
- The autotasks used for the HW interface initialization and communication also need to be authorized. Use access level CONTROL for the AUTHWxxx autotasks in your environment.

To permit access to a CPC resource in RACF, issue the PERMIT HSA.ET32TGT.USIBMSC.SCZP901 CLASS(FACILITY) ID(AUTXCF) ACC(CONTROL) command. The XCF autotask AUTXCF gets access level CONTROL for the CPC resource USIBMSC.SCZP901.

Similarly for a generic LPAR resource access, issue the PERMIT HSA.ET32TGT.USIBMSC.SCZP901.* CLASS(FACILITY) ID(AUTXCF) ACC(CONTROL) command. The XCF autotask AUTXCF gets access level CONTROL for all the logical partitions under the SCZP901.

6

Migrating to System Automation for z/OS

This chapter discusses migration to System Automation for z/OS. The topics discussed are:

6.1 Software Migration Project Office

This section introduces the Software Migration Project Office (SMPO) group. SMPO is a group of IBM specialists who are experienced in converting ISV products to IBM solutions. We also provide an overview of all the current SMPO offerings. In this section, the following topics are described:

- 6.1.1, “SMPO overview” on page 128
- 6.1.2, “Migration services offerings” on page 129

6.1.1 SMPO overview

New implementation of IBM and Tivoli products from other software vendors are primarily based on the following important considerations:

- Integrated software solutions that are reliable, robust and scalable.
- Software solutions that are affordable and have terms and conditions that are flexible and predictable.

These new implementations often involved the IBM Software Migration Project Office (SMPO) for migration assistance. Since 1993, the SMPO has been helping new software implementation projects migrate into some of the industry's leading database, z/OS, and systems management from IBM and Tivoli.

SMPO migration services offerings are backed up with a dedicated team of more than 90 migration specialists. This helps to ensure that your migration is performed professionally and successfully. To date, SMPO has performed more than 3,500 migrations and have a client satisfaction rating of 95 percent!

The SMPO provides expertise in migration solutions for z/OS based systems management, DB2, CICS tools, and security, including assistance with software portfolio management and cost containment.

The SMPO is a team of specialists with many years of practical experience migrating clients from other products. They utilize a proven methodology to ensure success gained from years of experience. The team focuses on the needs as expressed by the client.

6.1.2 Migration services offerings

The SMPO's migration service offerings have a flexible, modular structure that can be customized to meet a client's specific requirements. Our migration services offerings include:

- Migration Assessment

  Helps identify the migration issues unique to a specific environment. A migration assessment estimates the resources, time, and cost needed for a migration project.

- Migration Planning Service

  A migration specialist leads a joint client and IBM team that develops a migration plan detailing the tasks, target dates, and people assignments.

- Conversion Tools Services

  For products with large amounts of operational data, conversion tools may be used to automate the translation from one format to another, thereby reducing the time and errors involved in the conversion.

- Migration Consulting Services

  Migration specialists can provide guidance throughout the migration project. Typically, consulting services are provided during the technical analysis, testing and cut over phases of the project. Services can be provided onsite, through phone or remote dial-up.

- Migration Implementation Services

  Assistance is provided in the actual implementation of migration services includes:

- Product installation and customization
- Implementation of new function
- Exit design and coding
- Testing, test planning and validation
- Operations skills transfer
- Project management

To learn more about the products the SMPO migrates, please go to:

http://www.ibm.com/software/solutions/softwaremigration

6.2 A typical migration process

A typical automation migration process consists of several phases:

- 6.2.1, “Product implementation” on page 130
- 6.2.2, “Conversion” on page 132
- 6.2.3, “Production roll-out” on page 133
- 6.2.4, “Additional steps” on page 134

6.2.1 Product implementation

The implementation of System Automation for z/OS can be summarized in the following steps. Some consideration is also provided in Chapter 2, “System Automation for z/OS V3R1 configuration” on page 19.

- Software installation.

  The first step in the migration process is to install IBM Tivoli NetView for z/OS and System Automation for z/OS using SMP/E. You can find detailed informtion about this process from the IBM Tivoli NetView and System Automation for z/OS program directories.

  It is important to check for the latest APARs to ensure known issues with solutions are not propagated into your environment. You can select these APARs from the Preventive Service Planning (PSP) bucket information library. You can search the PSP bucket at

  `https://techsupport.services.ibm.com/server/390.psp390`

- Allocate datasets.

  Custom datasets must be allocated outside of SMP/E processing. Ensure the USS directories and files are created and defined as well. Additional customization dialog datasets must also be defined.

- Customize PARMLIB members.
  - Update SCHEDxx
  - Update MPFLSTxx
  - Update LPALSTxx
  - Update LNKLSTxx
  - Update IEFSSNxx

- Customize PROCLIB members.

  It is useful for operations and general efficiency of the PROCLIB dataset to have a common procedure library across systems. You can do this using system symbolics when allocating datasets. When allocating datasets for

System Automation for z/OS and NetView, keep this in mind when establishing naming standards.

- Customize IBM Tivoli NetView for z/OS.
  - Customize NetView Alert information.
  - Customize DSIPARM data; It can be advantageous in a sysplex environment to establish a plexwide DSIPARM dataset as well as a system specific DSIPARM. Doing so can promotes standardization while still having the flexibility for necessary system differences.
  - Modifying NetView DSIPARM.
    - Definition for an Automation Network (Optional).
    - AOFOPFGW Modifications.
    - Update DSIOPFU with local operator definitions.
    - Update security table if using NetView security.
    - Customize NetView Style Sheet.
  - Customize NetView CNM1NETV - Main Menu screen.
- Customizing the Automation Manager
  - Customizing HSAPRMxx: If using multiple logical sysplexes within a physical sysplex, update the GRPID field accordingly. Remember to also update the INGXINIT member of the SA agent as well within DSIPARM.
  - ARM instrumentation of the Automation Manager.
  - Security considerations.
  - Customizing the component trace.
  - Customizing the system logger.
- Install ISPF dialog screens.
  - Allocate libraries for the dialogs.
  - Invoking the ISPF dialogs.
  - Verify the ISPF dialog planning and installation.
  - Verify the number of available REXX environments.
- Defining automation policy.
  - Create policy definitions; You can initially create a limited Policy with only a few tasks defined for verification purposes.
  - Build the control files.
  - Distribute system operations control files.

- Define Host-to-Host communications.
  - Customize the SYS1.VTAMLST data set.
  - Perform VTAM definitions.
  - Add additional VTAM statements.
- Define security, for both IBM Tivoli NetView for z/OS and System Automation for z/OS.
- Customize the Status Display Facility (SDF); This is an optional step. This provides a simple 3270-based layout of your automated applications. In conjunction with SDF, the NetView Web Application and the NetView Management Console can also be configured for a graphical based interface view of your environment.
- Automate System Operations startup. After the scheduled IPL of the system, start Automation NetView on all images and verify cross system communications.
- Provide operator training.

6.2.2 Conversion

After the System Automation for z/OS has been installed and running, conversion may start. The conversion consists of:

- Classify current automation by type.
- Identify and eliminate obsolete automation.
- Convert message automation.
  - Convert message suppression automation to MPF or NetView Message Revision Table entries.
  - Convert message automation to System Automation for z/OS policy or NetView Message Automation Table (MAT) with REXX processing as needed.
- Convert REXX programs as needed; some tasks include:
  - Implement e-mail notification by implementing SMTP.
  - Automation programs that cannot be defined by a simple System Automation for z/OS policy must be converted to REXX program.
- Test message automation.
  - Construct test scenarios.
  - Test message automation.
  - Sign-off message automation test results.

- Convert command based automation.
  - Classify commands, if it is a custom operator command or z/OS command.
  - Custom commands automation can be converted to REXX.
  - Install and activate NetView MVS Command Exit.
- Convert Timer automation.
  - Add Timers to Automation Policy.
  - Test Timer automation REXX programs.
- Finalize automation conversion.
  - Review automation conversion.
  - Sign-off on automation conversion.
- Formal product training.

6.2.3 Production roll-out

When all the conversion has been implemented and tested, we are ready for the production roll-out:

- Establish infrastructure for System Automation for z/OS implementation.
  - Define all necessary shared NetView datasets.
  - Define Automation Policy for production LPARs.
  - Create Server Group for Automation Managers.
  - Determine need for NetView graphics.
- Implement System Automation for z/OS in production.
  - Propagate PARMLIB changes to images.
  - Propagate PROCLIB changes to images.
  - Propagate product libraries.
  - Disable current automation product.
  - IPL images.
  - Start Automation Manager.
  - Start System Automation for z/OS agents.
  - Verify System Automation for z/OS is operational.
  - System Automation for z/OS automation and operation verification.

6.2.4 Additional steps

There are some additional steps that must also be performed, such as:

- Update systems operation procedures
- Convert automation security to RACF or other local SAF product
- Production roll-out wrap-up

6.3 General migration tool

The original idea for these general migration tools came about during 2005 when the SMPO was asked to provide some ideas on helping reduce the total hours that are required when we perform migrations from non-IBM automation products to System Automation for z/OS. The System Automation for z/OS development lab chose not to participate in the tool development, so we were given the go ahead to code them.

The tools are written in REXX, making extensive use of both NetView PIPEs as well as NetView 3270 VIEW screens. They do not require an System Automation for z/OS NetView to run them, so they could be run in either a standard NetView or an System Automation for z/OS NetView, whichever is available. However, the minimum NetView required is V5R1, since the code uses the new EXTEND function of VIEW that was introduced in that release with APAR OA06590 (PTF UA09844 available on April 27, 2004).

Note: This new function is only documented in that APAR and the new NetView V5R2 manuals but was never described in the V5R1 library.

These tools are the first release of what is intended to be a growing amount of software that will help you with your migrations from non-IBM automation software to System Automation for z/OS. Feel free to e-mail `mailto:andrew.mcintyre@us.ibm.com` with any enhancement requests you might like to see and they can be considered for the next release.

The discussion includes:

- 6.3.1, “Installation” on page 135
- 6.3.2, “User interface” on page 135
- 6.3.3, “Auto Discovery of Started Tasks” on page 137
- 6.3.4, “Migration source code control” on page 139

6.3.1 Installation

The tools consists of the following members:

SAMSYSC	Compiled REXX program
SAMSYSP	Screens member

Both members should be put into your local REXX library. This is usually the first one in your DSICLD concatenation in the NetView Started Task Procedure.

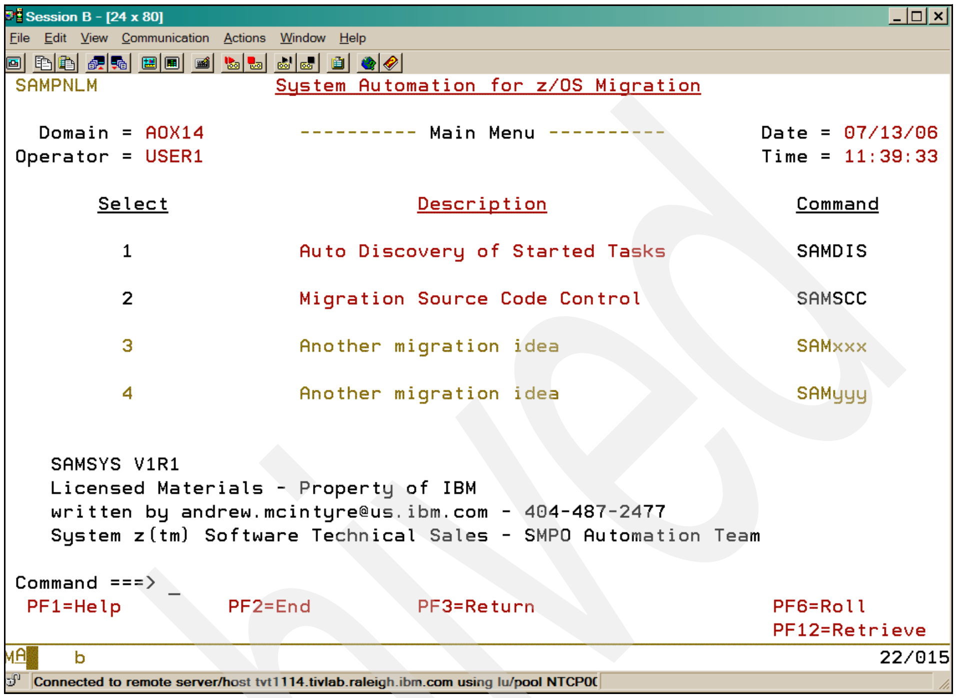
Note: The screens member does not need to be installed in your CNMPNL1 concatenation because it is loaded dynamically by the REXX code when you run it.

The program code allocates various files, including both existing and new datasets. All new datasets DCB parameters are always modeled after the first dataset in your DSICLD concatenation. You will need to ensure that your security software allows NetView to create these new datasets.

6.3.2 User interface

The migration tools use the standard NetView 3270 interface, including complete use of both the PF6 ROLL as well as the PF12 RETRIEVE functions. All of the usable PF keys are shown at the bottom of each screen.

Once the two members are put in the DSICLD concatenation, log on to NetView and issue SAMSYSC and press Enter to display the screen shown in Figure 6-1.



```
SAMPNLM                 System Automation for z/OS Migration

  Domain = AOX14          ---------- Main Menu ----------        Date = 07/13/06
Operator = USER1                                                  Time = 11:39:33

       Select                  Description                      Command

         1                Auto Discovery of Started Tasks          SAMDIS

         2                Migration Source Code Control            SAMSCC

         3                Another migration idea                   SAMxxx

         4                Another migration idea                   SAMyyy


   SAMSYS V1R1
   Licensed Materials - Property of IBM
   written by andrew.mcintyre@us.ibm.com - 404-487-2477
   System z(tm) Software Technical Sales - SMPO Automation Team

Command ===> _
 PF1=Help         PF2=End          PF3=Return                    PF6=Roll
                                                                 PF12=Retrieve
MA      b                                                                22/015
Connected to remote server/host tvt1114.tivlab.raleigh.ibm.com using lu/pool NTCP0(
```

Figure 6-1 Main menu screen

All the standard screen items are similar to a typical NetView application, such as the NetView domain, operator ID, date, and time. This migration tool consists of the following functions:

- Auto Discovery of Started Tasks.

  Started task discovery in a large SYSPLEX environment can be a very tedious process. The tool allows automatic collection of running tasks for all members of the SYSPLEX. We discuss this in 6.3.3, “Auto Discovery of Started Tasks” on page 137.

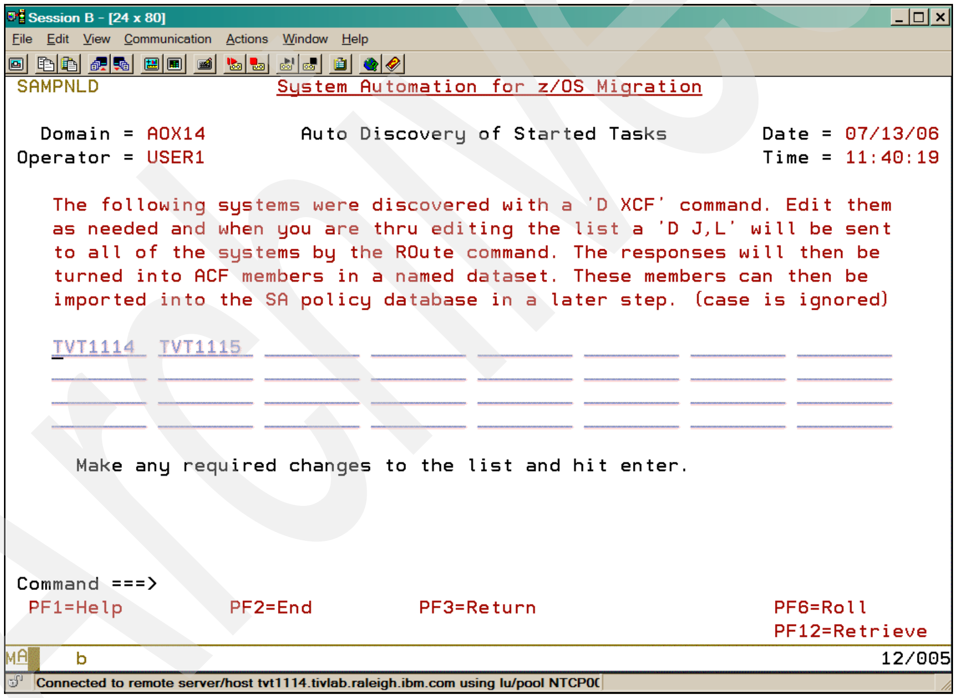
- Migration Source Code Control.

  In a large migration project with a sizeable amount of automation code, it is sometimes hard to pinpoint exactly the progress of the conversion. This tool allows you to manage this progress using a control file that stores the progress status of the migration effort. We discuss this in 6.3.4, “Migration source code control” on page 139.

6.3.3 Auto Discovery of Started Tasks

In handling started task management, you have to key in all of the job names of the various started tasks running on each system in the SYSPLEX for the initial System Automation for z/OS policy setup phase. This is one of the most time consuming steps in setting up System Automation for z/OS and this tool automates the entire process. The tool will also set up the connection in System Automation for z/OS policy between the job names and the systems they were found running on. You will still have to key in relationships and shut down commands but the time consuming and error prone work of keying in hundreds (sometimes thousands) of job names is now taken care of.

To start the process, select 1 from the initial menu (or you could issue SAMDIS from the command line). You will then see the screen in Figure 6-2, with the system names from your SYSPLEX automatically filled in.

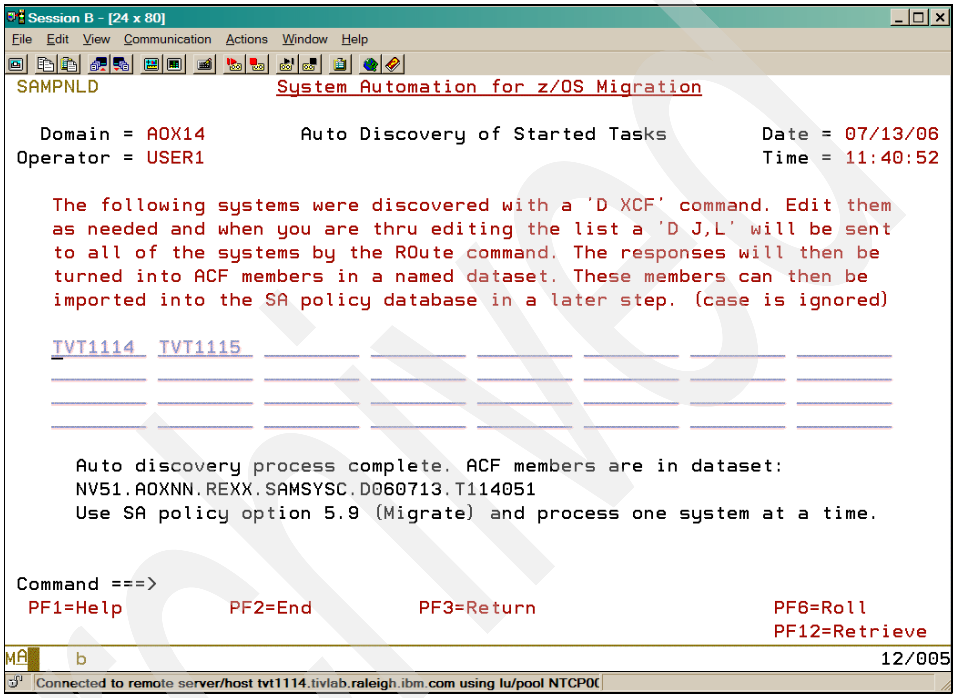


```
SAMPNLD                 System Automation for z/OS Migration

  Domain = AOX14         Auto Discovery of Started Tasks          Date = 07/13/06
Operator = USER1                                                  Time = 11:40:19

   The following systems were discovered with a 'D XCF' command. Edit them
   as needed and when you are thru editing the list a 'D J,L' will be sent
   to all of the systems by the ROute command. The responses will then be
   turned into ACF members in a named dataset. These members can then be
   imported into the SA policy database in a later step. (case is ignored)

   TVT1114_ TVT1115_ ________ ________ ________ ________ ________ ________
   ________ ________ ________ ________ ________ ________ ________ ________
   ________ ________ ________ ________ ________ ________ ________ ________
   ________ ________ ________ ________ ________ ________ ________ ________

     Make any required changes to the list and hit enter.



Command ===>
 PF1=Help         PF2=End         PF3=Return                    PF6=Roll
                                                                PF12=Retrieve
MA      b                                                             12/005
Connected to remote server/host tvt1114.tivlab.raleigh.ibm.com using lu/pool NTCP0(
```

Figure 6-2 Auto Discovery of Started Tasks screen

At this point, you can alter the system list however you wish. For example, you could delete one or more of the systems if you did not want to discover their started tasks at this time. One reason you might want to do this is that you may have different peak times from system to system when there are the most started tasks to discover. So you could discover some of the systems now and some later, whenever you wanted to do them.

Once you are satisfied with the list, press Enter for the automatic discovery to start. The text on the screen describes how the discovery process actually works. You will then see progress messages for each system flash by and when all the systems have been processed, you will see a screen like Figure 6-3.



```
SAMPNLD                     System Automation for z/OS Migration

  Domain = AOX14          Auto Discovery of Started Tasks          Date = 07/13/06
Operator = USER1                                                   Time = 11:40:52

   The following systems were discovered with a 'D XCF' command. Edit them
   as needed and when you are thru editing the list a 'D J,L' will be sent
   to all of the systems by the ROute command. The responses will then be
   turned into ACF members in a named dataset. These members can then be
   imported into the SA policy database in a later step. (case is ignored)

   TVT1114  TVT1115  ________ ________ ________ ________ ________ ________
   ________ ________ ________ ________ ________ ________ ________ ________
   ________ ________ ________ ________ ________ ________ ________ ________
   ________ ________ ________ ________ ________ ________ ________ ________

     Auto discovery process complete. ACF members are in dataset:
     NV51.AOXNN.REXX.SAMSYSC.D060713.T114051
     Use SA policy option 5.9 (Migrate) and process one system at a time.


Command ===>
 PF1=Help         PF2=End         PF3=Return                    PF6=Roll
                                                                PF12=Retrieve
MA     b                                                              12/005
Connected to remote server/host tvt1114.tivlab.raleigh.ibm.com using lu/pool NTCP0(
```

Figure 6-3 Auto Discovery of Started Tasks screen

Take note of the dataset name where the results of the discovery have been stored. This dataset name is based on the first dataset in your DSICLD concatenation with the added tokens of the date and time of the discovery as D*yymmdd* and T*hhmmss*. Each system discovered has its own member in this dataset where the member name is the name of the system.

The generated member can then be imported into a System Automation for z/OS policy. Use the following process:

1. Log on to TSO/ISPF and get into the System Automation for z/OS policy screens.
2. Start with an empty policy created from the EMPTY policy sample.

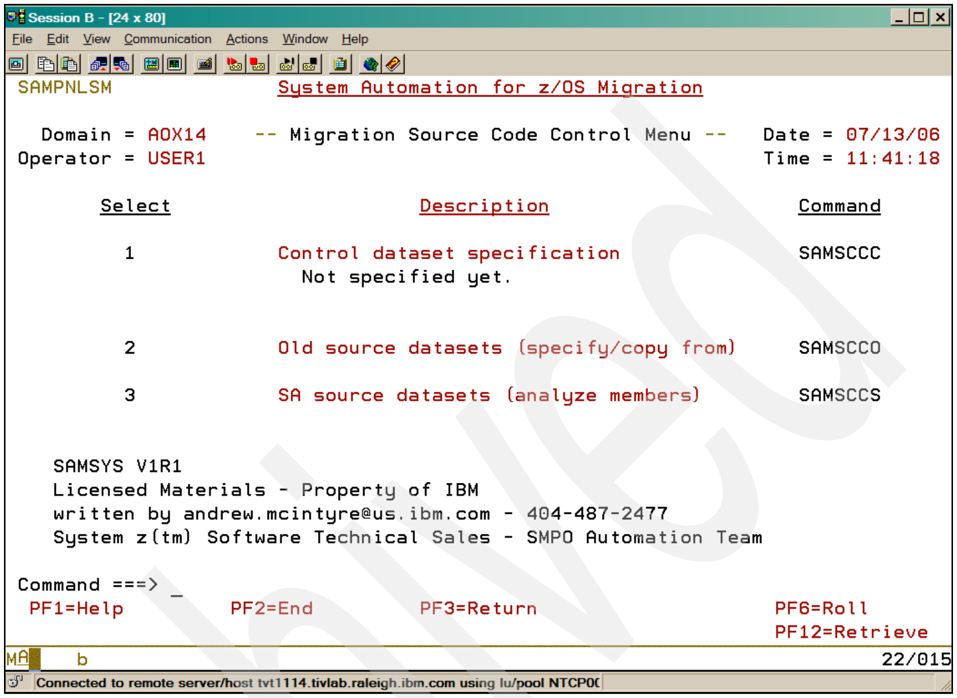
Note: If you do not start with the EMPTY sample, any existing entries of the same types with the same names of what was discovered will be overlaid with nulls (including any relationships, shutdown commands, and other fields), so be careful and always start with an empty policy.

3. Go to option 5.9 (Migrate) and specify one system name at a time and migrate the corresponding member name for that system into System Automation for z/OS policy. The System Automation for z/OS policy types that will be created are GRP, SYS, APG, and APL.
4. The Migrate option in the 5.9 menu only allows you to migrate one system at a time. When you are finished, you will have the job names of all the started tasks that were running on all the systems you discovered stored in your System Automation for z/OS policy.
5. Now you can add relationships, shutdown commands, and do any other policy work you need to do.

6.3.4 Migration source code control

This tool is for those larger migrations where either the client's project management, or sometimes the IBM project manager, wants to know *exactly* how far along the REXX migration has proceeded. This tool will avoid the necessity to keep up with the REXX migration progress manually, usually with a large, hard to maintain spreadsheet or project plan. Instead, it automates the analysis of what status the entire set of REXX migrations are in so that everyone can tell at a glance how far along the project has progressed.

To start the process, key in 2 from the initial menu or you could issue SAMSCC from the command line. You will then see the sub menu screen in Figure 6-4.

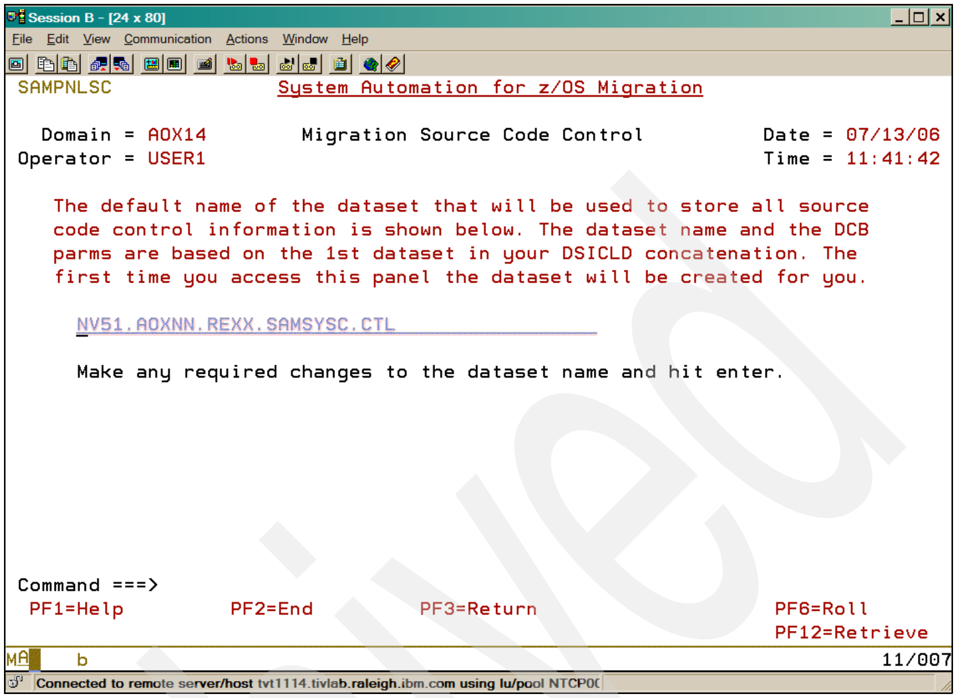


```
SAMPNLSM                   System Automation for z/OS Migration

  Domain = AOX14     -- Migration Source Code Control Menu --    Date = 07/13/06
Operator = USER1                                                 Time = 11:41:18

       Select                     Description                     Command

         1             Control dataset specification              SAMSCCC
                         Not specified yet.


         2             Old source datasets (specify/copy from)    SAMSCCO

         3             SA source datasets (analyze members)       SAMSCCS


   SAMSYS V1R1
   Licensed Materials - Property of IBM
   written by andrew.mcintyre@us.ibm.com - 404-487-2477
   System z(tm) Software Technical Sales - SMPO Automation Team

Command ===> _
 PF1=Help         PF2=End         PF3=Return                    PF6=Roll
                                                                PF12=Retrieve
MA      b                                                                22/015
Connected to remote server/host tvt1114.tivlab.raleigh.ibm.com using lu/pool NTCP0(
```

Figure 6-4 Migration Source Code Control Menu screen

This source code control tool is somewhat larger in scope than the auto discovery tool and that is the reason it has a sub menu available. You can also bypass this menu and go directly to any of the sub functions by using the commands shown on the screen, such as SAMSCCC, SAMSCCO, and SAMSCCS. These sub menus also represent the sequence of tasks that you need to perform for the source code control.

1. To specify the control dataset, just key in 1 on the sub menu screen (or use the SAMSCCC command) to specify the control dataset name. You get the screen in Figure 6-5 on page 141 with a proposed dataset name already filled in, that you can change however you wish.

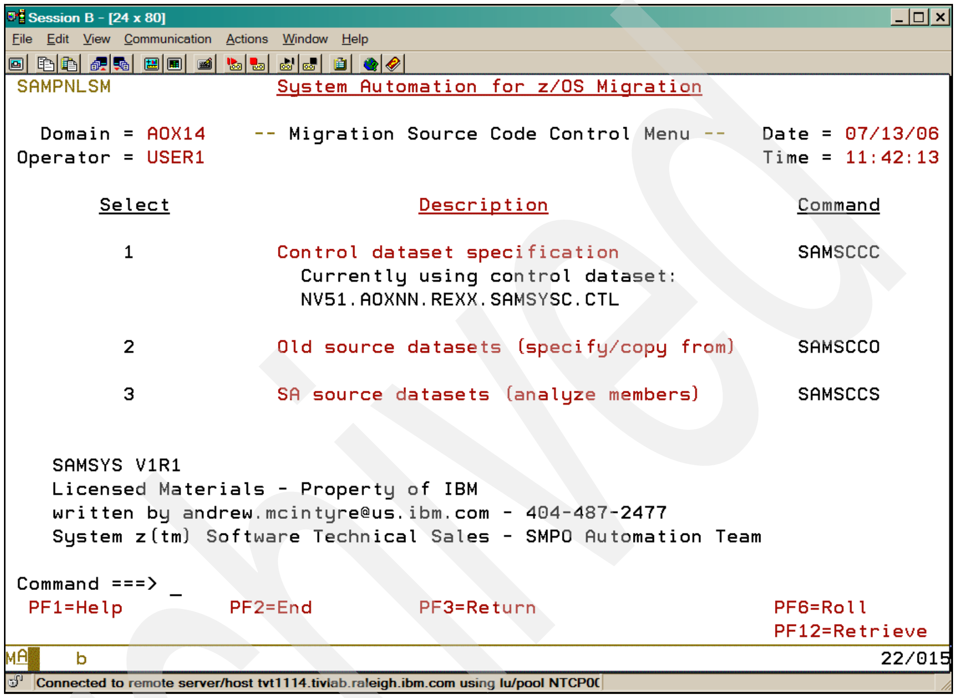


```
SAMPNLSC                 System Automation for z/OS Migration

  Domain = AOX14        Migration Source Code Control          Date = 07/13/06
Operator = USER1                                               Time = 11:41:42

   The default name of the dataset that will be used to store all source
   code control information is shown below. The dataset name and the DCB
   parms are based on the 1st dataset in your DSICLD concatenation. The
   first time you access this panel the dataset will be created for you.

     NV51.AOXNN.REXX.SAMSYSC.CTL

     Make any required changes to the dataset name and hit enter.







Command ===>
 PF1=Help         PF2=End          PF3=Return                    PF6=Roll
                                                                 PF12=Retrieve
```

Figure 6-5 Migration Source Code Control - control dataset screen

2. Press Enter and follow the prompts and you end up either allocating a new control dataset or using an already existing one. The messages that you see are self explanatory as to what is taking place.

Be aware that you can specify and switch between any number of control datasets. The tool stores this dataset name in a global variable in the NetView save/restore database. When you have successfully created a control dataset, the initial sub menu for this tool will show the dataset name currently being used, as in the screen shown in Figure 6-6.



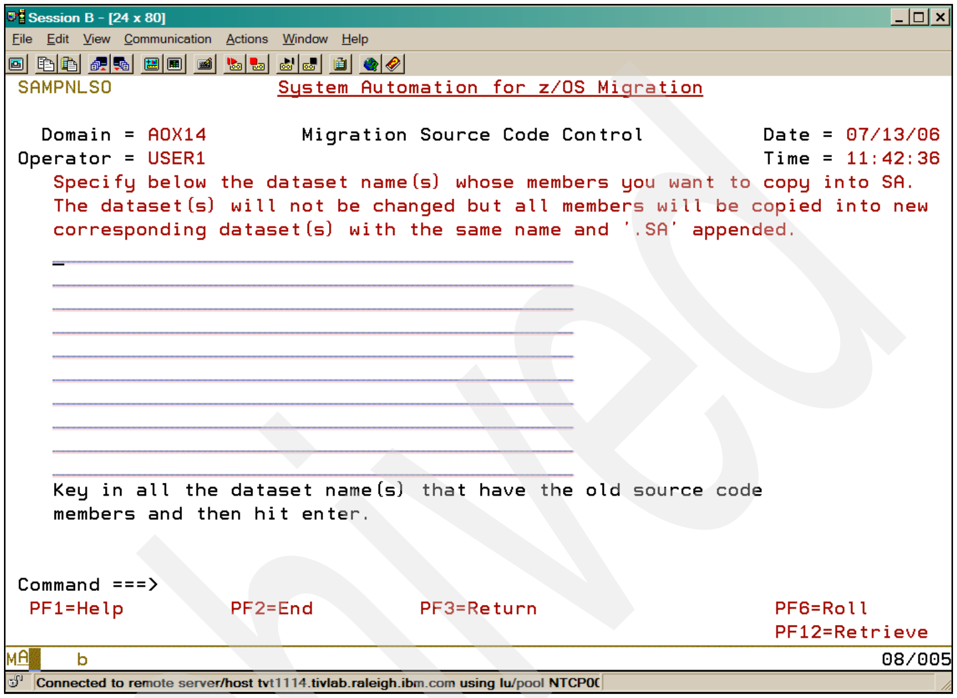
```
SAMPNLSM                  System Automation for z/OS Migration

  Domain = AOX14     -- Migration Source Code Control Menu --    Date = 07/13/06
Operator = USER1                                                 Time = 11:42:13

       Select                      Description                     Command

         1               Control dataset specification             SAMSCCC
                           Currently using control dataset:
                           NV51.AOXNN.REXX.SAMSYSC.CTL

         2               Old source datasets (specify/copy from)   SAMSCCO

         3               SA source datasets (analyze members)      SAMSCCS


   SAMSYS V1R1
   Licensed Materials - Property of IBM
   written by andrew.mcintyre@us.ibm.com - 404-487-2477
   System z(tm) Software Technical Sales - SMPO Automation Team

Command ===> _
 PF1=Help         PF2=End         PF3=Return                     PF6=Roll
                                                                 PF12=Retrieve
MA     b                                                                22/015
Connected to remote server/host tvt1114.tivlab.raleigh.ibm.com using lu/pool NTCP0(
```

Figure 6-6 Migration Source Code Control Menu screen with control dataset

3. The initial control dataset contains two members: #README and MIGREYEC. You can use TSO/ISPF to take a look at these two members. There are comments in both members with explanations as to their use.
4. Edit the MIGREYEC member if you want to use your own categories of REXXs. These category names are the key to this tool and allow you to keep track of your various REXX members in up to six different groups. The tool uses the first four characters of the names as headings on a later screen (that is why “copyed” is intentionally misspelled).

   Once you have settled on your category names in MIGREYEC, either using the suggestions or editing the member and changing them to whatever you want, you are ready to proceed with the next step.

5. Back on the sub menu screen in Figure 6-6 on page 142, key in 2 or use the SAMSCCO command to specify the old source datasets that contain your various REXX members. You will see the screen shown in Figure 6-7.



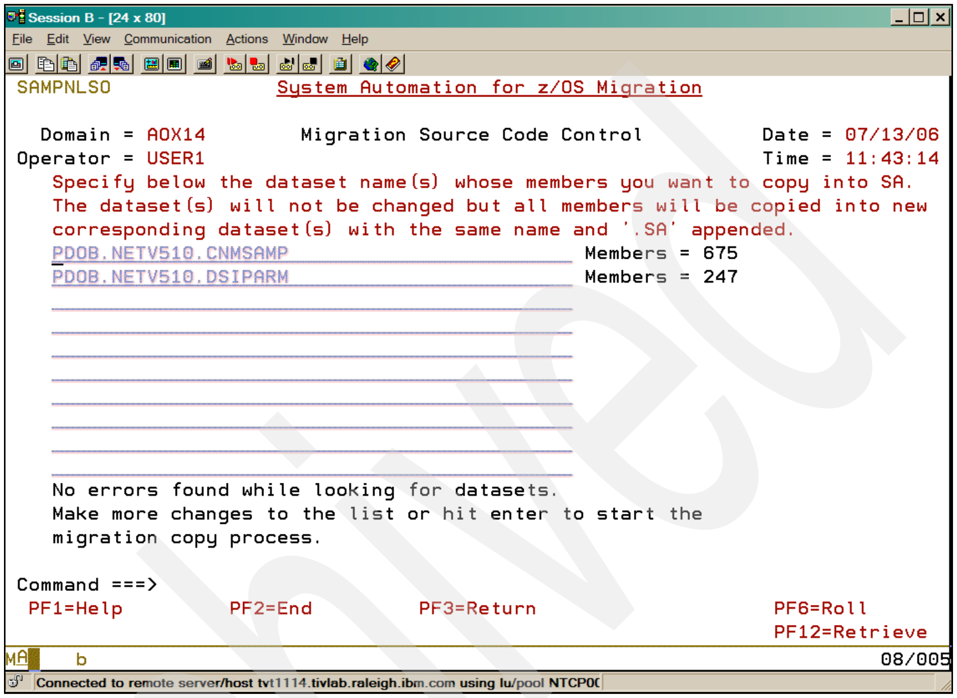
```
SAMPNLSO                      System Automation for z/OS Migration

  Domain = AOX14         Migration Source Code Control            Date = 07/13/06
Operator = USER1                                                  Time = 11:42:36
   Specify below the dataset name(s) whose members you want to copy into SA.
   The dataset(s) will not be changed but all members will be copied into new
   corresponding dataset(s) with the same name and '.SA' appended.

   _____________________________________________
   _____________________________________________
   _____________________________________________
   _____________________________________________
   _____________________________________________
   _____________________________________________
   _____________________________________________
   _____________________________________________
   _____________________________________________
   _____________________________________________
   Key in all the dataset name(s) that have the old source code
   members and then hit enter.


Command ===>
 PF1=Help         PF2=End         PF3=Return                     PF6=Roll
                                                                 PF12=Retrieve
MA      b                                                               08/005
Connected to remote server/host tvt1114.tivlab.raleigh.ibm.com using lu/pool NTCP0(
```

Figure 6-7 Migration Source Code Control - old dataset screen

6. You can key in up to ten dataset names that contain your original REXX source code. These datasets will not be changed in any way. They will simply have all of their members copied into new datasets that use the same name but with ".SA" appended to the end.

   If you have more than ten datasets, then you must specify additional control datasets and key in the old datasets in groups of ten each. This should not be a problem for most migrations.

7. Once you have keyed in your old dataset names and had the tool validate them, you will then see a screen similar to Figure 6-8 (the dataset names are just examples).

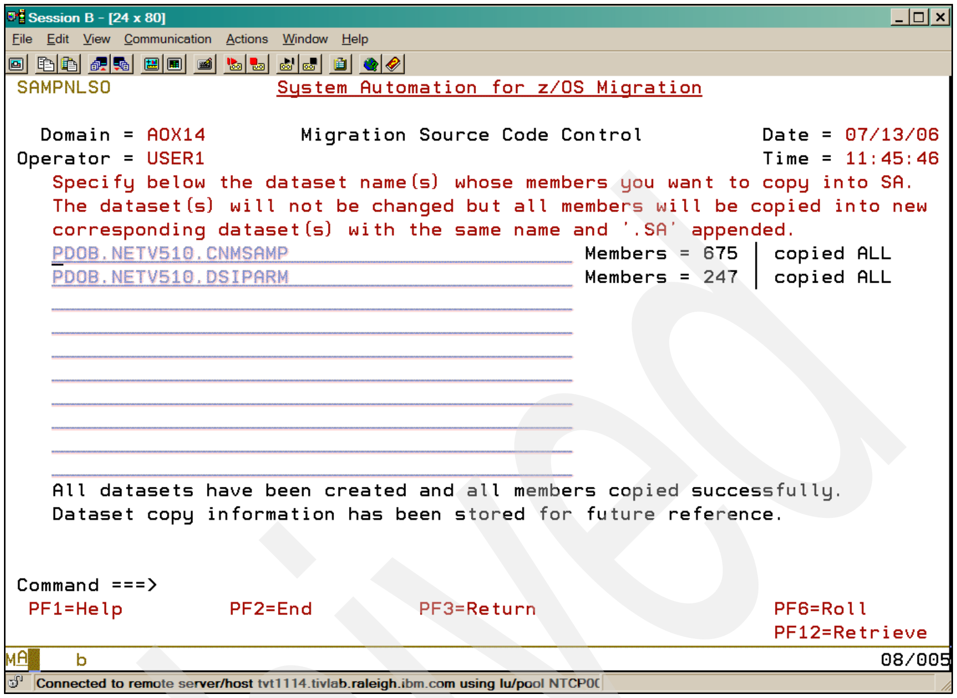


```
SAMPNLSO                  System Automation for z/OS Migration

  Domain = AOX14        Migration Source Code Control              Date = 07/13/06
Operator = USER1                                                   Time = 11:43:14
   Specify below the dataset name(s) whose members you want to copy into SA.
   The dataset(s) will not be changed but all members will be copied into new
   corresponding dataset(s) with the same name and '.SA' appended.
   PDOB.NETV510.CNMSAMP                          Members = 675
   PDOB.NETV510.DSIPARM                          Members = 247









   No errors found while looking for datasets.
   Make more changes to the list or hit enter to start the
   migration copy process.

Command ===>
 PF1=Help          PF2=End          PF3=Return                    PF6=Roll
                                                                  PF12=Retrieve
MA     b                                                                08/005
Connected to remote server/host tvt1114.tivlab.raleigh.ibm.com using lu/pool NTCP0C
```

Figure 6-8 Migration Source Code Control - verified old dataset screen

8. In Figure 6-8, it shows the total number of members found in each dataset. Now you can press Enter one final time to start the actual copy process. During this process, the first line of the MIGREYEC member that you validated earlier will be placed at the top of each source member. Depending on the number of datasets and the total number of members, this process may take a few minutes. A progress count is provided as the copy is taking place so you can see that everything is working successfully.

   After the copy process has completed, you will see a screen like Figure 6-9 on page 145 showing that the copy processing was successful. If there were any errors, you will see them as well on the same line as the dataset name.



```
SAMPNLSO                 System Automation for z/OS Migration

  Domain = AOX14         Migration Source Code Control          Date = 07/13/06
Operator = USER1                                                Time = 11:45:46
   Specify below the dataset name(s) whose members you want to copy into SA.
   The dataset(s) will not be changed but all members will be copied into new
   corresponding dataset(s) with the same name and '.SA' appended.
   PDOB.NETV510.CNMSAMP                        Members = 675 | copied ALL
   PDOB.NETV510.DSIPARM                        Members = 247 | copied ALL

   All datasets have been created and all members copied successfully.
   Dataset copy information has been stored for future reference.


Command ===>
 PF1=Help         PF2=End         PF3=Return                      PF6=Roll
                                                                   PF12=Retrieve
MA     b                                                                 08/005
Connected to remote server/host tvt1114.tivlab.raleigh.ibm.com using lu/pool NTCP0(
```

Figure 6-9 Migration Source Code Control - old dataset screen

At this point, you have created a complete copy of your existing REXX libraries, but with ".SA" appended to the dataset name and each member now having a new comment line inserted at the top.

9. You can now proceed with the actual migration, that is, editing and testing the various REXXs. You *must* edit the datasets residing in the *new* datasets with the suffix SA. As you perform this work, uncomment (take the * off) the word that corresponds to the status you want this REXX to have. For instance, the line below indicates that this member is in a `copyed` status:

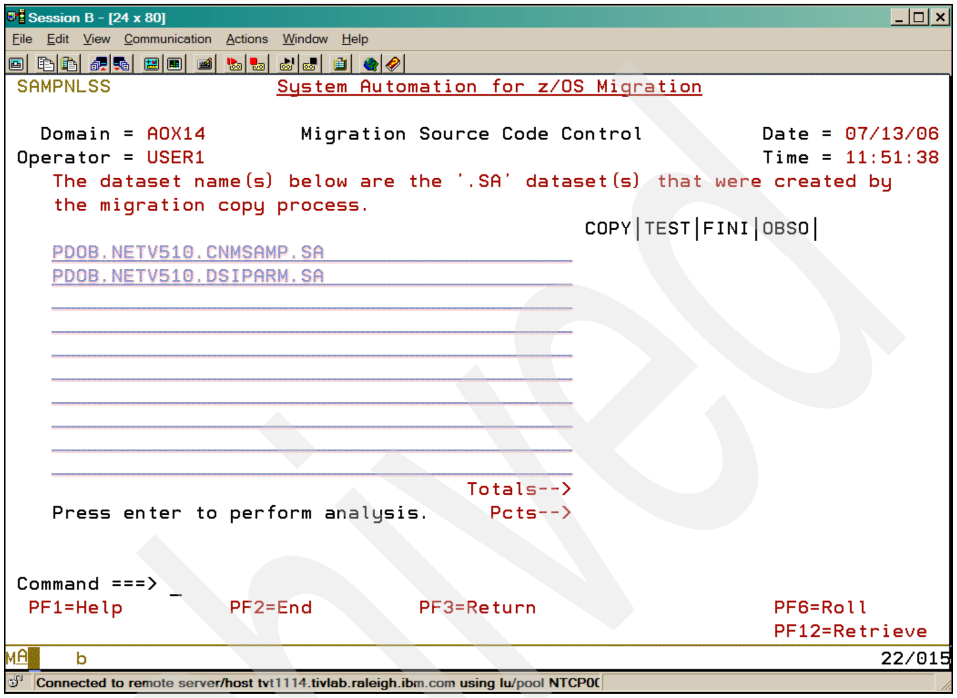
```
/* SA migration ---> copyed *testing *finished *obsolete <--- */
```

However, once you have started testing the REXX, you should edit the member to read:

```
/* SA migration ---> copyed  testing *finished *obsolete <--- */
```

Note: The rightmost, uncommented word is the status that is used during the analysis stage. In other words, there is no reason to comment (put an * back on) the earlier word as it leaves that particular status.

10. To get the overall progress of the migration, at any time just key in 3 on the sub menu screen or use the SAMSCCS command to analyze the new source datasets that contain your various REXX members. You will see Figure 6-10.

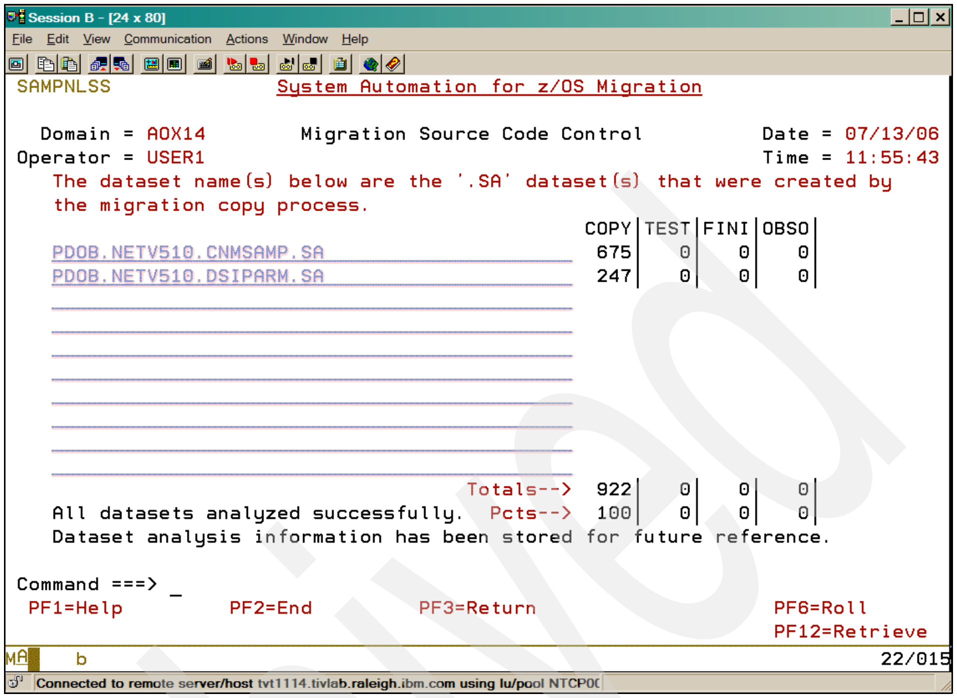


```
 SAMPNLSS                    System Automation for z/OS Migration

   Domain = AOX14         Migration Source Code Control            Date = 07/13/06
 Operator = USER1                                                  Time = 11:51:38
    The dataset name(s) below are the '.SA' dataset(s) that were created by
    the migration copy process.
                                                 COPY|TEST|FINI|OBSO|
    PDOB.NETV510.CNMSAMP.SA
    PDOB.NETV510.DSIPARM.SA










                                       Totals-->
    Press enter to perform analysis.     Pcts-->


 Command ===> _
  PF1=Help         PF2=End         PF3=Return                   PF6=Roll
                                                                PF12=Retrieve
MA     b                                                               22/015
Connected to remote server/host tvt1114.tivlab.raleigh.ibm.com using lu/pool NTCP0(
```

Figure 6-10 Migration Source Code Control - empty dataset progress screen

Notice that the new dataset names have been filled in for you and the category names you originally specified have been used as headings on this screen. We are only showing four categories, but there is enough space for six.

11. Press Enter for the tool to analyze all of your new source datasets. You can see the progress of the analysis as it reads each member. You will eventually see the screen shown in Figure 6-11 on page 147.



```
SAMPNLSS                    System Automation for z/OS Migration

  Domain = AOX14          Migration Source Code Control           Date = 07/13/06
Operator = USER1                                                  Time = 11:55:43
   The dataset name(s) below are the '.SA' dataset(s) that were created by
   the migration copy process.
                                                    COPY|TEST|FINI|OBSO|
   PDOB.NETV510.CNMSAMP.SA                           675|   0|   0|   0|
   PDOB.NETV510.DSIPARM.SA                           247|   0|   0|   0|









                                          Totals-->  922|   0|   0|   0|
   All datasets analyzed successfully.  Pcts-->  100|   0|   0|   0|
   Dataset analysis information has stored for future reference.

Command ===> _
 PF1=Help          PF2=End            PF3=Return                    PF6=Roll
                                                                    PF12=Retrieve
MA     b                                                                  22/015
```

Figure 6-11 Migration Source Code Control - dataset progress screen report

Anytime you go back into the SAMSCCS part of the tool, it redisplays the information from the last analysis. Then when you run it again, it overlays this data with the latest results.

The `Pcts` line is the completion percentages that you can provide to your project management anytime they want it. Now you do not have to manually keep up with the status of all those REXXs anymore! The tool does it for you!

6.4 Migration Unicenter CA-OPS/MVS

The Unicenter CA-OPS/MVS Web page can be viewed at:

http://www3.ca.com/solutions/Product.aspx?ID=3102

As an automation product, Unicenter CA-OPS/MVS provides event management and automation management for z/OS. It uses a mechanism called System State Manager (SSM) to achieve resource status. It integrates with other Unicenter products for graphical analyzer and UNIX System Services automation. It also provides an screen based automation definition called EasyRule.

The migration to System Automation for z/OS from CA-OPS/MVS should be done in phases. The first phase should include adding all system and application information into the policy database. This will provide the policy base for the remaining automation processes.

The migration process should take full advantage of the features and functions of System Automation for z/OS whenever possible. Of course, the amount of time available to complete the migration and the knowledge and experience in System Automation for z/OS will be factors in how the migration process is done.

In this section, we discuss the common methodologies used when migrating from CA-OPS/MVS to System Automation for z/OS. We will describe the methods for migrating the following features:

- 6.4.1, "System State Manager (SSM) tables and rule migration" on page 148
- 6.4.2, "Message rules" on page 148
- 6.4.3, "Command rules" on page 149
- 6.4.4, "Timer rules" on page 149
- 6.4.5, "Screen scraping rules" on page 150
- 6.4.6, "REXX migration" on page 151

6.4.1 System State Manager (SSM) tables and rule migration

System State Manager (SSM) application entries can be added to System Automation for z/OS policy during the initial phase of the migration project. This will allow the task management portion of System Automation for z/OS to run in monitor mode.

In monitor mode, System Automation for z/OS monitors and updates the status's for all objects defined to policy. With the Automation flags set to NO, status information and actions defined in policy can be monitored and verified against SSM actions without initiating duplicate automation.

An examination of the current message and REXX libraries should occur prior to the migration of the rules. All obsolete automation should be identified and excluded from the migration process.

6.4.2 Message rules

Message rules are entered into the System Automation for z/OS policy database. This is manifested in MESSAGE/USER DATA policy section for application or for MVS components. In the Message Processing screen, you specify how System Automation for z/OS should react if the specified message appears. The appropriate response should be documented in the Action column. Depending on the action, System Automation for z/OS displays a follow-on screen where you further specify the automated reaction to the message.

For each of the automated responses, an entry in the NetView automation table is required that specifies which routine System Automation for z/OS should be used to process your specified action. System Automation for z/OS is able to automatically build the NetView automation table entries for all CMD, REPLY, AUTO, and OVR actions as well as for certain USER and CODE actions.

The following actions are available:

- CMD: Issues commands as a response to a message.
- REP: Issues replies as a response to a WTOR issued by the application.
- CODE: Checks for certain codes within the message. If the codes are contained in the message, you can enter a value that is returned to the calling REXX.
- USER: Defines user-specified actions if the current message requires more specific automation than provided by the generic routines.
- AUTO: You can use the associated screen to specify messages that indicate the status of a resource. Once messages have been defined, they are automatically added to the NetView automation table.
- OVR: Allows you to override a default NetView automation table entry generated for a message.

6.4.3 Command rules

Two different techniques will suffice to handle these rules. If it is a uniquely named REXX command, where it does not replace a standard MVS command, then the REXX will be migrated into the System Automation for z/OS REXX library. It can be invoked from the MVS console by prefixing the command name with the subsystem command designator.

For commands that are replacing or rejecting standard MVS or JES commands, the NetView Command Management feature must be used. See the member CNMEMCXY in the CNMCLST dataset, which provides examples of using the NetView Command Management feature.

6.4.4 Timer rules

CA-OPS/MVS timer rules can be migrated to either the TIMERS entry section of policy or for timers linked to a specific application into the APPLICATION entry section using INGTIMER. For more complex time specification, you can use NetView CHRON timers.

- TIMERS entry type

  This dialog can be used to define timers that are scheduled to issue commands or REXX at certain times. These timers can be triggered at a particular day and time, after a certain day, or to repeat at certain intervals.

- INGTIMER

  This command links timer commands to subsystems. This ensures that the timer is only active when the subsystem is active. When the subsystem terminates, the timer commands are automatically purged.

- CHRON

  The CHRON command enables you to run commands at timed intervals and provides the following capabilities:

  – Repeat a command at regular intervals during a portion of the day, for example, once every hour from 8:00 a.m. to 12:00 noon.
  – Specify which days of the week a repeating command does or does not execute, for example, the third Tuesday of any month.
  – Specify which calendar days a repeating command does or does not execute. The dates are classified as holidays, and define any other classes you prefer.
  – Schedule commands that run at a specific time of day on multiple or specific days.

6.4.5 Screen scraping rules

Screen scraping rules should be eliminated whenever possible. Screens can often change, requiring the code to change also. However, there are times when this is the only automation technique available.

The VET command (and PIPE stage) is used to read data from, and subsequently write data to, a virtual screen using VOST. When used as a first stage in a pipe, VET obtains data from the screen in one of the following forms:

- Row
- Field
- Message

Row and field form data is returned in message BNH150I. If a issued command does not result in a full screen being presented on the virtual screen, the message displayed on the screen is returned to VET in message form. If the application running returns a message and a full screen, both the message and full-screen data are returned to VET.

When VET is used as a command or as a subsequent pipe stage, it writes data to the virtual screen. Data is written in the form of simple text where one line is written for each input-capable field on the virtual screen.

An example of full screen automation is contained in member CNMS1098 in the CNMSAMP dataset.

6.4.6 REXX migration

Many REXX programs can be eliminated with policy entries. For the REXX members that must be maintained, the majority of code can be migrated as is. The primary difference is migrating the CA-OPS/MVS REXX functions or extensions.

For functions that are used extensively, it may be beneficial to re-write the command for use by System Automation for z/OS. For example, the OPSVALUE function in CA-OPS/MVS is often used extensively for the updating and retrieving of global variables. We can emulate this using the NetView global variables using a module, such as the GLBVALUE program in REXX. For migration purposes, the OPSVALUE references can then simply be changed to GLBVALUE to complete the migration of that portion of REXX code.

Writing the code to emulating functions can be time consuming. You must weigh the cost of writing the emulation code against re-writing the original REXX code to use System Automation for z/OS functions.

Example 6-1 shows the structure of the GLBVALUE sample that can replace the OPSVALUE function.

Example 6-1 GLBVALUE program

```
/*********************************************************************/
/*                                                                   */
/*   NAME       - GLBVALUE                                           */
/*   FUNCTION   - A FUNCTION THAT MANIPULATES GLOBAL VARIABLES.      */
/*                CALL PARAMETERS ARE:                               */
/*                P1 - VARIABLE NAME                                 */
/*                P2 - OPTION.  VALID OPTIONS ARE:                   */
/*                     'A' - INCREMENT A GLOBAL BY VALUE IN P3.      */
/*                     'V' - RETURNS VALUE OF GLOBAL, OR NULLS.      */
/*                     'L' - RETURNS A LIST OF GLOBAL VARIABLES      */
/*                           THAT BEGIN WITH THE NAME SPECIFIED      */
/*                     'E' - RETURNS 'I' IF VARIABLE EXISTS, OR 'N'  */
/*                           IF IT DOES NOT.                         */
/*                     'F' - RETURNS 'I' IF VARIABLE EXISTS, OR 'N'  */
/*                           IF IT DOES NOT. (SAME AS 'E')           */
/*                     'R' - REMOVES A VARIABLE, OR VARIABLES IF     */
/*                           VALUE IS GENERIC (SUCH AS GLV.XX.*).    */
/*                     'S' - RETURNS A LIST OF ALL GLOBAL VARIABLES  */
/*                          THAT BEGIN WITH THE NAME SPECIFIED      */
/*                    'U' - UPDATES THE VALUE OF THE VARIABLE TO    */
/*                          THE VALUE SPECIFIED IN P3.              */
/*               P3 - VALUE (FOR A AND U ONLY)                      */
```

6.5 Migrating BMC MAINVIEW AutoOPERATOR

The AutoOPERATOR product was originally sold by Boole and Babbage until BMC bought them in 1999. The AutoOPERATOR product can be viewed at:

http://www.bmc.com/products/proddocview/0,2832,19052_0_28229_8571,00.html

It consists of various automation modules that allows automation management of a z/OS environment. SMPO is usually involved with the z/OS, CICS, and IMS module migration.

The product is integrated with the MAINVIEW suite of monitoring software. However, we run into very little MAINVIEW related performance alerts that are being handled by AutoOPERATOR automation. Usually, the products are managed by separate groups in an organization and are rarely directly involved with each other.

With the migration of AutoOPERATOR to System Automation for z/OS, we recommend also using the IBM Tivoli OMEGAMON family of monitoring using the Tivoli Enterprise Portal and its out of the box (or client setup) automated situations to handle whatever performance related automation MAINVIEW was doing.

The vast majority of automation that is being run in a z/OS shop, such as message and command traps, z/OS started task management, and home grown custom REXX code, is almost always completely confined to the AutoOPERATOR environment and its features, and this is what we spend the majority of our time migrating to System Automation for z/OS.

Table 6-1 shows the BMC products and the IBM Tivoli software equivalents.

Table 6-1 Product mapping

BMC product	**IBM/Tivoli product**	**Comments**
AutoOPERATOR for z/OS	System Automation for z/OS	Base SA
AutoOPERATOR for CICS	System Automation for z/OS (CICS feature)	Included with base sa
AutoOPERATOR for IMS	System Automation for z/OS (IMS feature)	Included with base sa
MAINVIEW SYSPROG Services (originally known as RESOLVE)	System Automation for z/OS + standard z/OS commands (+ possibly OMEGAMON XE on z/OS for some commands)	Most of the RESOLVE commands can be handled with the latest z/OS commands
MAINVIEW (monitoring)	IBM Tivoli OMEGAMON XE	

6.5.1 IMFEXEC migration tool from the SMPO

Migrating REXX automation code is always one of the most time consuming aspects of any automation migration. The SMPO has developed a specific run time tool to help migrate any custom REXX code a client might have. For the AutoOPERATION client, SMPO issues a REXX subroutine called IMFEXEC, which performs a similar function with the AutoOPERATOR function calls. This allows the client's REXX to remain unchanged as much as possible and to continue to invoke IMFEXEC as it always has when moved to the System Automation for z/OS environment. This saves a lot of time in migrating, editing, and testing REXX code.

The various IMFEXEC subfunctions are handled by the SMPO IMFEXEC module in the same way they were under AutoOPERATOR, except now they use only System Automation for z/OS and NetView for z/OS functions. The module currently consists of over 600 lines of REXX code and the source code is provided during any SMPO automation migration engagement.

The subfunctions that are currently handled, with their supported parameters, are described below:

- CMD: Issues an MVS command.

```
/* CMD ---------------------------- */
/* handles non-quoted (no response) */
/*             quoted (responses)   */
/*     COUNT|LINES(0) (no response) */
/*     COUNT|LINES(n) (responses)   */
/*           # prefix (drops)       */
/*      and variables IMFCC = 0     */
/*                    IMFRC = 0     */
/*                    IMFNOL        */
/*                    LINEnnnn      */
/* errors out on . BBI commands     */
```

- SELECT: Runs a REXX program.

```
/* SELECT ------------------------- */
/* handles EXEC only                */
/*      and variables IMFCC = 0     */
/*                    IMFRC = 0     */
/* errors out on not EXEC           */
```

- MSG: Logs a message to the NetView netlog.

```
/* MSG ---------------------------- */
/* handles quoted and non quoted    */
/*      and variables IMFCC = 0     */
/*                    IMFRC = 0     */
```

- WTO: Writes a message to a console.

```
/* WTO ---------------------------- */
/* handles quoted (required)        */
/*         ROUTCDE(n..nn)           */
/*            DESC(n..nn)           */
/*      and variables IMFCC = 0     */
/*                    IMFRC = 0     */
/*                    IMFWTDOM      */
```

- ALERT: Writes an alert message to either (or both) the Status Display Facility (SDF) or the Tivoli Enterprise Portal Event Console.

```
/* ALERT --------------------------- */
/* handles alert key (first parm)   */
/* quoted alert text (required)     */
/*  FUNCTION|FUN(ADD|DELETE) (only) */
/*               ADD is the default */
/*  PRIORITY|PRI                    */
/*     (CRITICAL|CRIT)       red    */
/*     (HIGH)                red    */
/*     (MAJOR)               pink   */
/*     (MINOR)               yellow */
/*     (WARNING|WARN)        dkblue */
/*     (INFORMATIONAL|INFO)  ltblue */
/*     (CLEARING)            green  */
/*  QUEUE|QUE(MAIN|queue name)      */
/*      and variables IMFCC = 0     */
/*                    IMFRC = 0     */
```

- SUBMIT: Submits a batch job.

```
/* SUBMIT -------------------------- */
/* handles quoted (required)        */
/*      and variables IMFCC = 0     */
/*                    IMFRC = 0     */
```

- WAIT: Pauses the execution.

```
/* WAIT ---------------------------- */
/* handles non quoted time in secs  */
/*      and variables IMFCC = 0     */
/*                    IMFRC = 0     */
/* errors out on NAME parm          */
```

- VGET: Copies one or more variables from the NetView pool to the REXX function pool.

```
/* VGET ---------------------------- */
/* handles single var - no parens   */
/*  sing or mult vars - parens      */
/*         LOCAL   - task           */
/* default SHARED  - common         */
/*         PROFILE - common (saved) */
/*      and variables IMFCC = 0     */
/*                    IMFRC = 0     */
/* errors out on INTO parm          */
```

- VPUT: Copies one or more variables from the REXX function pool to the NetView pool.

```
/* VPUT --------------------------- */
/* handles single var - no parens   */
/*  sing or mult vars - parens      */
/*         LOCAL   - task           */
/* default SHARED  - common         */
/*         PROFILE - common (saved) */
/*      and variables IMFCC = 0     */
/*                    IMFRC = 0     */
/* errors out on FROM parm          */
```

- VDEL: Deletes one or more variables from the NetView pool.

```
/* VDEL --------------------------- */
/* handles single var - no parens   */
/*  sing or mult vars - parens      */
/*            pattern - asterisk    */
/*              (last byte only)    */
/*         LOCAL   - task           */
/* default SHARED  - common         */
/*         PROFILE - common (saved) */
/*      and variables IMFCC = 0     */
/*                    IMFRC = 0     */
```

6.5.2 Future enhancements

During our next AutoOPERATOR migration, we intend on adding the following enhancements to our IMFEXEC tool:

- VDCL to declare a variable list, as well as the VGET INTO and the VPUT FROM parameters.
- VLST to return a list of variables.
- VENQ and VDEQ to lock and unlock a resource, as well as their automatic use on SHARED and PROFILE variables.

Other possibilities include some of the RES (Resolve) parameters as well as the CICS and IMSTRAN subfunctions, as well as any other IMFEXEC subfunctions that might be highly used at the client and that would help speed up the migration or make the migrated code more reliable.

Abbreviations and acronyms

ACF	Advanced Communication Function
AIX	Advanced Interactive eXecutive
AON	Automated Operation Network
APAR	Authorized Program Analysis Request
APF	Authorized Programming Facility
API	Application Programming Interface
ARM	Automatic Restart Manager
CBPDO	Custom Build Product Delivery Options
CD-ROM	Compact Disc Read Only Memory
CICS	Customer Information Control System
CMOS	Complementary Metal Oxide Semiconductor
CPC	Central Processing Complex
DASD	Direct Access Storage Device
DB2	Database 2™
DCB	Data Control Block
DFSMS	Data Facility System Managed Storage
EAS	Event Automation Services
EIF	Event Integration Facility
ESCON	Enterprise System Connection
FICON	Fiber Connection
GDPS	Geographically Dispersed Parallel Sysplex
HFS	Hierarchical File System
HLQ	High Level Qualifier
HMC	Hardware Management Console
HSM	Hierarchical Storage Manager
HTTP	Hyper Text Transfer Protocol
I/O	Input Output
IBM	International Business Machines Corporation
IMS	Information Management System
IPL	Initial Program Load
ISPF	Interactive System Productivity Facility
ITSO	International Technical Support Organization
JAAS	Java Authentication and Authorization Services
JAR	Java Archive
JCL	Job Control Language
JES	Job Entry Subsystem
JRE	Java Runtime Environment
KDE	K Desktop Environment
LAN	Local Area Network
LPAR	Logical Partition
MCS	Multiple Console Support
MVS	Multiple Virtual Storage
NCCF	NetView Command Control Facility
NCP	Network Communication Program
NLDM	Network Logical Data Manager
NMC	Network Management Console

NPDA	Network Problem Determination Application
OPC	Operation Planning and Control
PPI	Program-to-program Interface
PPT	Program Property Table
PTF	Program Temporary Fix
RACF	Resource Access Control Facility
REXX	Restructured EXtended eXecutor
RMF	Resource Measurement Facility
RODM	Resource Object Data Model
SAF	System Authorization Facility
SDK	Software Development Kit
SDSF	System Display and Search Facility
SLES	SUSE Linux Enterprise Server
SMP/E	System Modification Program Extended
SMPO	Software Migration Project Office
SMS	System Managed Storage
SMTP	Simple Mail Transfer Protocol
SNA	System Network Architecture
SNMP	Simple Network Management Protocol
SSI	Subsystem Interface
SVC	Supervisory Call
SYSPLEX	System Complex
TAF	Terminal Access Facility
TCP/IP	Transmission Control Protocol Internet Protocol
TEC	Tivoli Enterprise Console
TSCF	Target System Control Facility
TSO	Time Sharing Option
UDB	Universal Database
URL	Universal Resource Locator
USS	UNIX System Services
VSE	Virtual Storage Environment
VTAM	Virtual Telecommunication Access Method
WTO	Write to Operator
WTOR	Write to Operator Reply
XCF	Cross-system Coupling Facility
XML	eXtensible Markup Language

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this IBM Redbook.

IBM Redbooks

For information about ordering these publications, see "How to get IBM Redbooks" on page 160. Note that some of the documents referenced here may be available in softcopy only.

- *IBM Tivoli OMEGAMON XE V3.1.0 Deep Dive on z/OS*, SG24-7155
- *End-to-end Automation with IBM Tivoli System Automation for Multiplatforms*, SG24-7117
- *Automation Using Tivoli NetView OS/390 V1R3 and System Automation OS/390 V1R3*, SG24-5515

Other publications

These System Automation for z/OS publications are also relevant as further information sources:

- *IBM Tivoli NetView Installation: Configuring Additional Components*, SC31-8874
- *IBM Tivoli System Automation for Multiplatforms V2.1.1 End-to-End Automation Management User's Guide and Reference*, SC33-8211
- *IBM Tivoli System Automation for Multiplatforms V2.1.1 Release Notes*, SC33-8214
- *IBM Tivoli System Automation for z/OS CICS Automation Programmer's Reference and Operator's Guide*, SC33-8267
- *IBM Tivoli System Automation for z/OS Customizing and Programming*, SC33-8260
- *IBM Tivoli System Automation for z/OS Defining Automation Policy*, SC33-8262
- *IBM Tivoli System Automation for z/OS IMS Automation Programmer's Reference and Operator's Guide*, SC33-8268

- *IBM Tivoli System Automation for z/OS Messages and Codes*, SC33-8264
- *IBM Tivoli System Automation for z/OS Operator's Commands*, SC33-8265
- *IBM Tivoli System Automation for z/OS Planning and Installation*, SC33-8261
- *IBM Tivoli System Automation for z/OS Programmer's Reference*, SC33-8266
- *IBM Tivoli System Automation for z/OS TWS Automation Programmer's Reference and Operator's Guide*, SC23-8269
- *IBM Tivoli System Automation for z/OS User's Guide*, SC33-8263
- *IBM Tivoli System Automation for z/OS End-to-End Automation Adapter*, SC33-8271
- *Tivoli NetView for z/OS Installation: Configuring Additional Components*, SC31-8874
- *Tivoli NetView for z/OS V5R2 Installation: Getting Started*, SC31-8872

Online resources

These Web sites are also relevant as further information sources:

- IBM Preventive Service Planning search site

  `https://techsupport.services.ibm.com/server/390.psp390`
- IBM Tivoli Systems Automations for z/OS pages

  `http://www-306.ibm.com/software/tivoli/products/system-automation-390/`
  `http://publib.boulder.ibm.com/tividd/td/IBMTivoliSystemAutomationforzOS3.1.html`
- IBM Tivoli NetView for z/OS page

  `http://www-306.ibm.com/software/tivoli/products/netview-zos/`
  `http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?toc=/com.ibm.itnetviewforzos.doc/toc.xml`

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

IBM Redbooks

IBM Tivoli System Automation for z/OS Enterprise Automation

(0.2"spine)
0.17"<->0.473"
90<->249 pages

IBM Tivoli System Automation for z/OS Enterprise Automation

Enterprise class automation on IBM System z

Migration from other automation platforms

Support for end-to-end automation

This IBM Redbook provides an overview of IBM Tivoli System Automation for z/OS concepts and new features. We also discuss some considerations on migrating non-IBM products to IBM Tivoli System Automation for z/OS.

The discussion is primarily aimed at technical professionals that are looking to understand the IBM Tivoli System Automation for z/OS new features and find migration options.

This IBM Redbook explains some important features of IBM Tivoli System Automation for z/OS, including using the Processor Operations feature, integration with IBM Tivoli OMEGAMON Classic, and integration with End-to-end automation feature of the IBM System Automation for Multiplatform.

We will also discuss the available services from the Software Migration Project Office that provide migration services for mainframe based products, including IBM Tivoli System Automation for z/OS, in this new implementation of IBM Tivoli System Automation for z/OS. We also provide some overview of the migration procedures from non-IBM products.

SG24-7308-00 ISBN 0738494631