

System Automation for z/OS
Version 4.Release 1

End-to-End Automation



Edition Notes

This edition applies to IBM® System Automation for z/OS Version 4, Release 1 (Program Number 5698-SA4), an IBM licensed program, and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters.

This edition replaces SC34-2750-01.

© **Copyright International Business Machines Corporation 1996, 2017.**

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

Contents

Figures.....	vii
Tables.....	ix
About this publication.....	xi
Who should use this publication.....	xi
Where to find more information.....	xi
The System Automation for z/OS Library.....	xi
Related Product Information.....	xii
Accessibility.....	xii
Using assistive technologies.....	xii
Keyboard navigation of the user interface.....	xii
Summary of Changes.....	xiii
New Information.....	xiii
Changed Information.....	xiv
Deleted Information.....	xv
Chapter 1. Introducing End-to-End Automation, Agent, and Adapter.....	1
Service Management Unite (SMU) Server.....	1
End-to-End Automation on z/OS.....	2
SA z/OS End-to-End Automation Adapter.....	3
Communication.....	4
Universal Automation Adapter (UAA).....	6
Chapter 2. Concepts of End-to-End Automation.....	7
Overview of the Architecture.....	7
Cross Domain Automation.....	8
Multiple E2E Automation Domains.....	9
Using E2E Automation.....	9
Defining DMN and REF objects.....	10
Defining cross-domain relationships.....	10
Loading the E2E Automation configuration.....	10
Activating E2E Automation.....	10
Showing that E2E Automation is activated.....	10
Status of an automation domain.....	10
Status of a resource reference.....	11
Operational information.....	11
Commands supporting REF and DMN.....	11
Conflicting Goals of Different SA z/OS Automation Domains.....	13
User Credentials of the End-to-End Automation Manager.....	14
Chapter 3. Prerequisites and Environment Setup.....	15
Setup Java Runtime.....	15
Setup Environment Variables.....	15
Setup Install-Root and Custom-Root Directory.....	16
Default USS File System.....	16
User-Defined USS File System for the Automation Adapter.....	16
Shared USS File System.....	17
Setup TCP/IP Stack Name in a TCP/IP Multi-Stack Environment (OA58444).....	18

Support of MVS System Symbols in Configuration Files.....	18
---	----

Chapter 4. Customizing the End-to-End Agent..... 19

Copy and Customize the Start Procedure.....	20
Copy and Customize the Environment Configuration File.....	20
Create the Credentials Configuration File.....	22
Set Up the SSL Configuration.....	23
Customize the SSL Configuration File.....	23
Create the Keystore File.....	24
Import the Public Key From the Keystore into the Truststore.....	24
Configuring the End-to-End Automation Adapter for Use by the End-to-End Agent.....	24

Chapter 5. Operating the End-to-End Agent..... 25

Starting the End-to-End Agent.....	25
Stopping the End-to-End Agent.....	25
Self-Terminating the End-to-End Agent.....	25
Cleaning Up the End-to-End Agent.....	26
Recycling the End-to-End Agent.....	26
Checking If the End-to-End Agent Is Running.....	26
Dependencies of the End-to-End Agent.....	26
Making the End-to-End Agent Highly Available.....	26

Chapter 6. Quick Startup of End-to-End Automation Adapter..... 29

Overview of Quick Startup.....	29
SA z/OS Automation Policy.....	30
SA z/OS Initialization Member INGXINIT.....	30
SA z/OS Sample JCL Procedure INGXADPT.....	30
SA z/OS USS Configuration Files (Automatically)	31
SA z/OS USS Configuration Files (Manually).....	31
NetView SSI JCL Procedure.....	32
NetView Event/Automation Service (E/AS).....	32
End-to-End Automation Adapter Startup.....	32
Verifying End-to-End Automation Adapter.....	33
Following Up.....	33

Chapter 7. Enabling NetView Event/Automation Service (E/AS)..... 35

Starting the E/AS Address Space.....	35
Customize the Sample Procedure.....	35
Configuring the Global Initialization File (Default: IHSAINIT).....	36
Configuring the NetView Message Adapter Service (Default: IHSAMCFG).....	36

Chapter 8. Customizing System Automation Agent..... 37

SA z/OS Initialization Member INGXINIT.....	37
SA z/OS Automated Operator Functions.....	37
SA z/OS Advanced Automation Options.....	38

Chapter 9. Customizing USS and TCP/IP..... 41

End-to-End Automation Adapter.....	41
Prerequisites for USS.....	41
Customizing USS.....	41
Customize the Link List.....	42
End-to-End Automation Agent.....	43
Prerequisites for USS.....	43

Chapter 10. Customizing End-to-End Automation Adapter..... 45

Configuration Files.....	45
--------------------------	----

The Automation Adapter Environment Configuration File.....	45
The Automation Adapter Master Configuration File.....	46
The Automation Adapter Plug-in Configuration file.....	48
The SSL Configuration File.....	49
How to Encrypt the Keystore Password.....	50
How to Create Keystore and Truststore for Use by SSL.....	51
The JAAS Configuration File.....	51
Chapter 11. Operating the End-to-End Automation Adapter.....	53
How to Use the Automation Adapter.....	53
USS Script.....	53
JCL Procedure Using BPXBATCH.....	55
SA z/OS Defined Resource.....	55
Command INGE2E.....	55
Chapter 12. Security Consideration for the End-to-End Automation Adapter.....	57
Authentication.....	57
Authority checking by the automation adapter.....	57
Authorization checking by user exit.....	57
Authorization checking by System Automation security.....	59
Chapter 13. Setting Up E2E Automation RACF Keyrings (OA58444).....	61
E2E Adapter RACF Keyring.....	61
Define E2EAdapterCA.....	61
Export E2EAdapterCA.....	62
Import SMUCA.....	63
Keytool – SMU.....	63
Create and Export SMU Public Client CA into E2E Adapter.....	63
Import E2EAdapter Public Server CA into SMU.....	64
E2E Agent RACF Keyring.....	65
Define E2EAgentCA.....	65
Connect E2EAgentCA public key to E2E Adapter Keyring.....	66
Connect E2EAdapterCA public key to E2E Agent Keyring.....	67
Configure the E2E Automation SSL Properties File.....	67
SMU Server (Linux).....	67
E2E Agent (z/OS).....	68
E2E Adapter (z/OS).....	68
Additional RACF Profile Access for E2EUSER.....	69
Chapter 14. Problem Determination of the End-to-End Automation Adapter.....	71
The Java log file.....	71
Diagnosis of the SA z/OS tasks for the adapter.....	71
End-to-end automation adapter diagnosis commands.....	72
Problems with selecting the primary agent.....	72
Problems with the program-to-program interface.....	73
Problems with NetView event/automation service.....	73
Examples.....	73
Problems with consumer registration.....	74
Link list errors.....	74
TCP/IP bind socket problem.....	75
Timeout problems.....	75
USS problems.....	76
USS Shared File System.....	76
CHECK_HEALTH function failure.....	76
Automation domain remaining unavailable after error recovery.....	76
Problem Determination of the End-to-End Automation Agent.....	77
The Java Log File.....	77

End-to-End Agent Diagnosis Commands.....	77
End-to-End Agent Loses Connection Information.....	77
Mixed SA Environment with End-to-End Agent.....	77
Appendix A. Using IBM Tivoli System Automation Application Manager.....	79
Appendix B. High Availability of the End-to-End Automation Adapter.....	81
Limitations to the automation.....	81
Defining automation policy.....	81
Application definitions for the NetView event/automation service (E2EEAS).....	81
Application definitions for the end-to-end automation adapter (E2EADPT).....	81
Application group definitions for the automation adapters (E2EADPT and E2EADPT_X).....	85
Automation purpose and rules.....	85
Non-disruptive recovery from errors.....	86
Resynchronization of the automation adapter.....	86
Recycling the automation adapter on the same system.....	86
Restarting the automation adapter or failover to a backup system.....	87
Automated applications.....	87
Grouping considerations.....	87
Appendix C. Configuration file parameters.....	89
Appendix D. SA z/OS specific implementation of the end-to-end automation resource data model.....	93
Resource data displayed by the WebUI.....	93
Domain object.....	93
Node object.....	94
Resource object.....	95
Requests issued by the SMU and the end-to-end automation manager.....	97
Overwriting commands.....	99
Request priority.....	99
Disregarding the hold flag.....	100
INGE2E.....	100
Example verification output.....	103
Status mapping.....	104
Introduction.....	104
Status mapping steps.....	105
Appendix E. Messages.....	109
End-to-end automation adapter messages.....	109
End-to-end automation adapter plug-in messages.....	122
Appendix F. Notices.....	143
Trademarks.....	144
Terms and conditions for product documentation.....	144
Index.....	147

Figures

- 1. SA z/OS Automation Adapter Connected to SMU..... 2
- 2. End-to-End Automation on z/OS..... 3
- 3. INGAMS Shows E2E Capability of PAM and AGENT on SA z/OS V4.1..... 4
- 4. End-to-End Automation Adapter Communication..... 5
- 5. Cross Domain Relationship..... 8
- 6. Multiple E2E Managers for Distinct Sets of Automation Domains..... 9
- 7. End-to-end automation adapter communication..... 30
- 8. Sample automation adapter master configuration file, ing.adapter.properties..... 46
- 9. Communication Paths of the End-to-End Automation Components..... 47
- 10. Example automation adapter plug-in configuration file, ing.adapter.plugin.properties..... 48
- 11. Example SSL configuration file, ing.adapter.ssl.properties..... 50
- 12. E2E Automation RACF Keying settings..... 61
- 13. DISPAOPS command response..... 72
- 14. INGAMS command dialog panel..... 72
- 15. Event/automation service STATUS information..... 73
- 16. Event/automation service QSTATS information..... 74
- 17. INGEVOBS LIST command response..... 74
- 18. Automation Adapter Connected to IBM Tivoli System Automation Application Manager..... 80
- 19. E2E automation adapter policy relationships..... 87

Tables

1. System Automation for z/OS library.....	xi
2. Commands Supporting REF and DMN.....	11
3. Files in the Customization Path.....	19
4. Parameters for inge2eagt.properties.....	20
5. E2E Sample Policy - E2E Agent Component.....	27
6. Automation adapter advanced automation options.....	38
7. Environment Variables of the Automation Adapter.....	45
8. Information Messages for AUDITMSG=true parameter.....	49
9. Master configuration file (ing.adapter.properties) parameters.....	89
10. Plug-in configuration file (ing.adapter.plugin.properties) parameters.....	91
11. SSL configuration file (ing.adapter.ssl.properties) parameters.....	92
12. Mapping of end-to-end automation requests to SA z/OS commands.....	97
13. SA z/OS automation manager status mapping.....	105
14. Status mapping: second step if the SA z/OS compound status reports a problem.....	106

About this publication

This publication describes the end-to-end automation adapter for z/OS and how it enables end-to-end automation and how it connects to Service Management Unite.

Who should use this publication

This publication is intended for:

- System programmers—primarily responsible for the installation, configuration, and customization of the automation adapter, but might also make use of the command interface
- Operators

Where to find more information

The System Automation for z/OS Library

Table 1 on page xi shows the information units in the System Automation for z/OS library. These manuals can be downloaded from [IBM Documentation](#).

Title	Form Number	Description
<i>Get Started Guide</i>	SC27-9532	This book is intended for SA z/OS® beginners. It contains the information about early planning, configuring the product, making it secure, customizing your automation environment, and the basic operational tasks that you perform on a daily basis.
<i>Planning and Installation</i>	SC34-2716	Describes SA z/OS new capabilities and how to plan, install, configure, and migrate SA z/OS.
<i>Customizing and Programming</i>	SC34-2715	Describes how to adapt the standard installation, add new applications to automation, write your own automation procedures, and add new messages for automated applications.
<i>Defining Automation Policy</i>	SC34-2717	Describes how to define and maintain the automation policy.
<i>User's Guide</i>	SC34-2718	Describes SA z/OS functions and how to use SA z/OS to monitor and control systems.
<i>Messages and Codes</i>	SC34-2719	Describes the problem determination information of SA z/OS, including messages, return codes, reason codes, and status codes.
<i>Operator's Commands</i>	SC34-2720	Describes the operator commands available with SA z/OS, including their purpose, format, and specifics of how to use them.
<i>Programmer's Reference</i>	SC34-2748	Describes the programming interfaces of SA z/OS and the definitions for the status display facility (SDF).

Table 1. System Automation for z/OS library (continued)

Title	Form Number	Description
<i>End-to-End Automation</i>	SC34-2750	Describes the end-to-end automation adapter for z/OS and how it enables end-to-end automation and how it connects to Service Management Unite Automation.
<i>Service Management Unite Automation Installation and Configuration Guide</i>	SC27-8747	Describes how to plan, install, set up, configure, and troubleshoot Service Management Unite Automation.
<i>Product Automation Programmer's Reference and Operator's Guide</i>	SC34-2714	Describes how to customize and operate product automation components (CICS, Db2, and IMS automation) with SA z/OS to provide a simple and consistent way to monitor and control all of the CICS, Db2, and IMS regions, both local and remote, within your organization.
<i>TWS Automation Programmer's and Operator's Reference Guide</i>	SC34-2749	Describes how to customize and operate TWS Automation.

Related Product Information

For information that supports System Automation for z/OS, visit the z/OS library in IBM Documentation (<https://www.ibm.com/docs/en/zos>).

Accessibility

Accessibility features help users with physical disabilities, such as restricted mobility or limited vision, to use software products successfully. System Automation for z/OS supports several user interfaces. Product functionality and accessibility features vary according to the interface.

The major accessibility features in this product enable users in the following ways:

- Use assistive technologies such as screen reader software and digital speech synthesizer, to hear what is displayed on screen. Consult the product documentation of the assistive technology for details on using those technologies with this product and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Magnify what is displayed on screen.

The product documentation includes the following features to aid accessibility:

- All documentation is available to both HTML and convertible PDF formats to give the maximum opportunity for users to apply screen-reader software
- All images in the documentation are provided with alternative text so that users with vision impairments can understand the contents of the images.

Using assistive technologies

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

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Vol 1* for information about accessing TSO/E and ISPF interfaces. These

guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

Summary of Changes

This document contains information previously presented in System Automation for z/OS V3R5.0 End-to-End Automation Adapter.

The following section lists important or major changes to this document.

New Information

OA58444 Enhancements (Jan 2020)

OA58444 is a compatibility APAR for SMU 1.1.7, and it also introduces the following enhancements to E2E Automation:

- System Automation V4.1 now supports network isolation for E2E Automation components via multiple TCP/IP stacks. See [“Setup TCP/IP Stack Name in a TCP/IP Multi-Stack Environment \(OA58444\)”](#) on page 18.
- End-to-end automation configuration now supports system symbols in the configuration files. See [“Support of MVS System Symbols in Configuration Files”](#) on page 18.
- The SSL configuration file can now be customized using an alternative way, that is, creating certificates residing in RACF. See [Chapter 13, “Setting Up E2E Automation RACF Keyrings \(OA58444\),”](#) on page 61.
- The **eez-functional-authentication** parameter is added into the automation adapter master configuration file. This parameter can switch on or off E2E adapter RACF checking for requests sent from the SMU functional user ID.
- The E2E adapter and agent are enhanced to support TLS version 1.2. **E2E_SSL_VERSION** is added into the configuration file `ingadapter.properties`. **INGAGT_SSL_VERSION** is added into the configuration file `inge2eagt.properties`.
- You can add user-specific Java system properties by specifying the new **E2E_USRJP** parameter in the automation adapter environment configuration file and the new **INGAGT_USRJP** parameter in the automation agent environment configuration file.
- You can add user-specific classpath by specifying the new **E2E_USRCP** parameter in the automation adapter environment configuration file and the new **INGAGT_USRCP** parameter in the automation agent environment configuration file.

New Chapters

The following new chapters are added:

- [Chapter 2, “Concepts of End-to-End Automation,”](#) on page 7
- [Chapter 4, “Customizing the End-to-End Agent,”](#) on page 19
- [Chapter 5, “Operating the End-to-End Agent,”](#) on page 25
- [Chapter 9, “Customizing USS and TCP/IP,”](#) on page 41
- [“Problem Determination of the End-to-End Automation Agent”](#) on page 77

Configuration Assistant enhancements

The Configuration Assistant is enhanced to configure the connection of SAplices to Service Management Unite (SMU) and end-to-end automation (E2E automation adapter and E2E agent).

If you want to use Configuration Assistant to perform the configuration tasks in the following chapters, please refer to "Chapter 9. Base SA z/OS Configuration Using the Configuration Assistant" in *IBM System Automation for z/OS Planning and Installation*.

- [Chapter 4, “Customizing the End-to-End Agent,”](#) on page 19
- [Chapter 6, “Quick Startup of End-to-End Automation Adapter,”](#) on page 29

- [Chapter 7, “Enabling NetView Event/Automation Service \(E/AS\),” on page 35](#)
- [Chapter 10, “Customizing End-to-End Automation Adapter,” on page 45](#)

End-to-End Automation on z/OS

End-to-End Automation on z/OS is added to [Chapter 1, “Introducing End-to-End Automation, Agent, and Adapter,” on page 1.](#)

Environment setup of end-to-end agent

Environment setup of end-to-end agent is added to [Chapter 3, “Prerequisites and Environment Setup,” on page 15.](#)

Parameter E2E_DROPCON

Parameter E2E_DROPCON is added to [“The Automation Adapter Environment Configuration File” on page 45.](#)

New topic Command INGE2E

[“Command INGE2E” on page 55](#) is added to [Chapter 11, “Operating the End-to-End Automation Adapter,” on page 53.](#)

New parameters in INGE2E command

New parameters RESET, STOP_E2EAGT, CLEANUP_E2EAGT, and DEBUG_E2EAGT are added to [“INGE2E” on page 100.](#)

Changed Information

Chapter 1. Introducing End-to-End Automation, Agent and Adapter

This chapter is updated to include how the automation adapter works with IBM Service Management Unite and end-to-end agent. See [Chapter 1, “Introducing End-to-End Automation, Agent, and Adapter,” on page 1.](#)

Figure 9. End-to-end automation adapter communication

This figure is updated to include end-to-end agent. See [Figure 7 on page 30.](#)

Figure 10. Sample automation adapter master configuration file, `ing.adapter.properties`

This figure is updated to include `eif-send-to-agent-hostname` and `eif-send-to-agent-port` parameters. See [Figure 8 on page 46.](#)

The JAAS Configuration File

The JAAS Configuration File is updated. See [“The JAAS Configuration File” on page 51.](#)

Rename E2E_ADPT

The E2E_ADPT is renamed as follows in all locations it appears:

- E2E_ADPT to E2EADPT
- E2E_ADPT_X to E2EADPT_X
- E2E_SUPPORT to E2ESUPPORT
- E2E_SUPPORT APG to E2EADPT APG
- E2E_EAS to E2EEAS (not including AOF_E2E_EAS_PPI)
- E2E_AUTOOPS to E2EADPT_AUTOOPS

Example SSL configuration file

The example SSL configuration file is updated. See [“The SSL Configuration File” on page 49.](#)

The following topics are renamed

- "Policy item MESSAGES/USERDATA" is renamed to "Startup definitions and sequence". See [“Startup definitions and sequence” on page 82.](#)
- "Policy item STARTUP" is renamed to "Startup procedure INGXADPT". See [“Startup procedure INGXADPT” on page 85.](#)

- "MOVE group for the automation adapters (E2EADPT_X)" is updated and renamed to "Application group definitions for the automation adapters (E2EADPT and E2EADPT_X)". See ["Application group definitions for the automation adapters \(E2EADPT and E2EADPT_X\)"](#) on page 85.

Figure 19. Sysplex group relationships

This figure is updated by deleting the MA/WOD relationship and adding the new MU/WD and MA/WR relationships. See [Figure 19 on page 87](#).

Deleted Information

"Using the data directory for log files" is removed from Chapter 10 Customizing End-to-End Automation Adapter.

The following topics are removed from ["Defining automation policy"](#) on page 81:

- Policy item DESCRIPTION
- Policy item APPLICATION INFO
- Policy item RELATIONSHIPS
- Policy item STARTUP
- Policy item SHUTDOWN
- Policy item THRESHOLDS
- Defining commands
- Defining code processing

Chapter 1. Introducing End-to-End Automation, Agent, and Adapter

This information provides a brief introduction to concepts and architecture of SA z/OS end-to-end automation and how the end-to-end automation adapter (also called automation adapter in this information) relates to IBM Service Management Unite (SMU) and IBM Tivoli System Automation Application Manager.

If you use IBM Tivoli System Automation Application Manager to automate E2E solutions, see [Appendix A, “Using IBM Tivoli System Automation Application Manager,”](#) on page 79.

Service Management Unite (SMU) Server

IBM Service Management Unite (SMU) is a customizable service management user interface that provides dashboards to monitor and operate z system environments. It provides a single point of control to monitor and operate in your environment. Operators can quickly and confidently analyze, isolate and diagnose problems as all relevant data including important logs is provided in a single place. SMU also enables operators to interact directly with the system by issuing commands and viewing results without going to a different console.

An *automation domain* is a system or a cluster of systems that has a local automation technology of its own to automate local resources, for example System Automation for z/OS (SA z/OS) or the Universal Automation Adapter (UAA). Each automation domain is connected to the SMU server via the automation adapter.

Connecting Service Management Unite

[Figure 1 on page 2](#) shows how SA z/OS end-to-end automation adapter connects the SAplex to SMU. The *web-based dashboard* is a graphical user front-end to all connected automation domains. The graphical user front-end is part of the SMU component that is included in SA z/OS.

The Universal Automation Adapter (UAA) can also be connected to the SMU server. The UAA supports distributed platforms, such as Linux.

Note: [Advanced users] If you purely want to connect one or multiple SA z/OS sub-plexes to the SMU server and you have no need for E2E automation, then install only the E2E adapter. In this case, you don't need to install the E2E agent and define any DMN or REF resource in the System Automation policy.

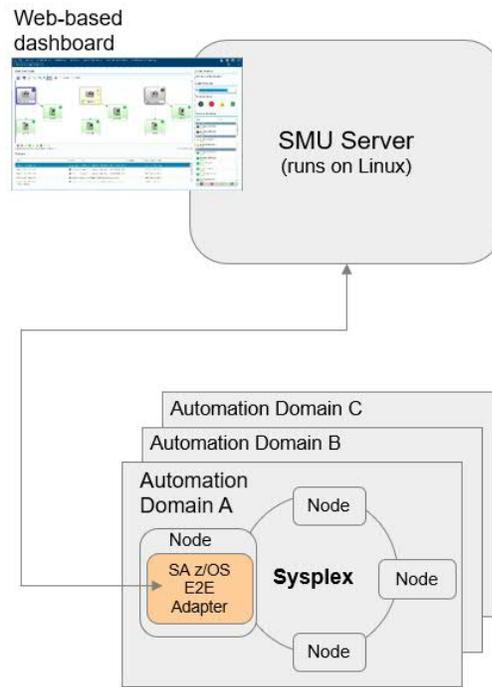


Figure 1. SA z/OS Automation Adapter Connected to SMU

End-to-End Automation on z/OS

With SA z/OS V4.1, the automation manager provides cross sysplex automation capabilities. With APAR OA55386, SA z/OS V4.1 extends its automation capabilities to cross platform resources.

Prior to V4.1, SA z/OS automation manager automated resources in the same sysplex (or SApplex). Now the SA z/OS automation manager is able to automate resources between multiple sysplexes (or SApplexes) and cross platform resources. In this case, the primary automation manager (PAM) is also called **E2E manager**. This cross sysplex/platform automation is also called **end-to-end automation**.

In conjunction with the E2E manager, there is a new address space, which represents the **E2E agent**. The E2E agent assists the E2E manager by executing start and stop orders or collecting status of resources running on remote automation domains. The E2E agent interfaces with the remote automation adapter.

To enable the PAM to become the E2E automation manager, you must have

1. Defined REF and DMN objects in the currently active policy.
2. Started a new address space, which represents the E2E agent.
3. Started an end-to-end automation adapter on each remote domain for which a DMN object exists.

Figure 2 on page 3 shows that E2E automation manager for z/OS connects multiple remote SA z/OS automation domains and Universal Automation Adapters to the E2E agent. The E2E agent enables the PAM to take over the role of the E2E automation manager.

This figure also shows that SA z/OS can be connected to an SMU server and the Universal Automation Adapters can be connected to the SMU server.

The web-based dashboard is used to view all automation domains and resources.

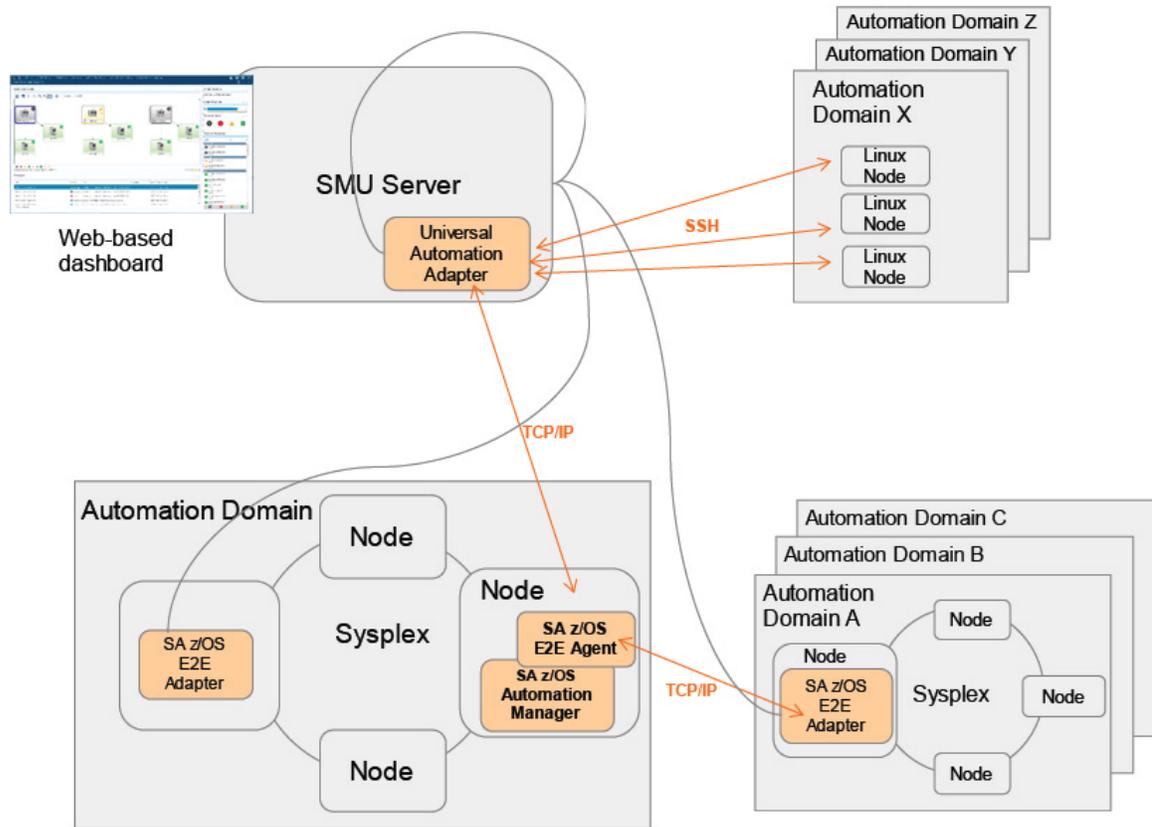


Figure 2. End-to-End Automation on z/OS

For a detailed description of the end-to-end automation concepts, see [Chapter 2, “Concepts of End-to-End Automation,”](#) on page 7.

SA z/OS End-to-End Automation Adapter

The automation adapter connects the automation domain with the SMU server or with the E2E agent.

Each SA z/OS automation domain represents an SAplex. It contains one or multiple z/OS systems with System Automation for z/OS and the end-to-end automation adapter. The automation adapter can run on only one system in the SAplex. The system where it runs is automatically selected as the primary agent. This is shown in command INGAMS by YES in column E2E.

The purpose of the automation adapter is to:

- Monitor status of resources within the automation domain.
- Send resource status change events.
- Start and stop resources within the automation domain.
- Provide information about resources in response to queries.

SA z/OS Node

For SA z/OS, a node is a z/OS system with NetView and System Automation for z/OS installed and running.

SA z/OS Automation Domain Name

The automation domain is the group of SA z/OS agents and managers that belong to the *same XCF Group ID* (same GRPID). SA z/OS calls this group SAplex.

The automation domain must have a unique name. You can change its default name to your needs by setting the parameters **plugin-domain-name** within the end-to-end automation adapter configuration file `ing.adapter.plugin.properties`.

If this parameter is not set, the default name of the automation domain for SA z/OS is the blank-delimited combination of physical sysplex name and INGXSgn where nn is the 2-character XCF group ID, for example, "KEYAPLEX INGXSG34".

You find the sysplex name and the XCF group name emphasized in INGAMS command output, as shown in Figure 3 on page 4. The YES under column E2E (SA version 4.1 and higher) or PA (SA version 3.5 or lower) shows that this agent is the primary agent and it runs the end-to-end automation adapter.

```

INGKYAM0          SA z/OS - Command Dialogs          Line 1    of 3
Domain Id . : IPUFJ ----- INGAMS -----          Date . . : 07/16/15
Operator Id : OPER          Sysplex = KEY1PLEX          Time . . : 14:10:19

Cmd:  A Manage      B Show Details  C Refresh Configuration  D Diagnostic

CMD System  Member  Role  Status  Sysplex  XCF Group  Release  Comm  E2E
-----
  KEYA     KEYA$$$2  PAM   READY   KEYAPLEX  INGXSG34  V3R5M0  XCF
  KEYA     KEYA     AGENT  READY   KEYAPLEX  INGXSG34  V3R5M0  XCF  YES
  KEYB     KEYB     AGENT  READY   KEYAPLEX  INGXSG34  V3R5M0  XCF
  
```

Figure 3. INGAMS Shows E2E Capability of PAM and AGENT on SA z/OS V4.1

Communication

The automation adapter sends events and receives synchronous requests for execution.

Figure 4 on page 5 shows that the automation adapter can send events to two different targets depending on customization. The event target is the SMU server and/or the E2E agent.

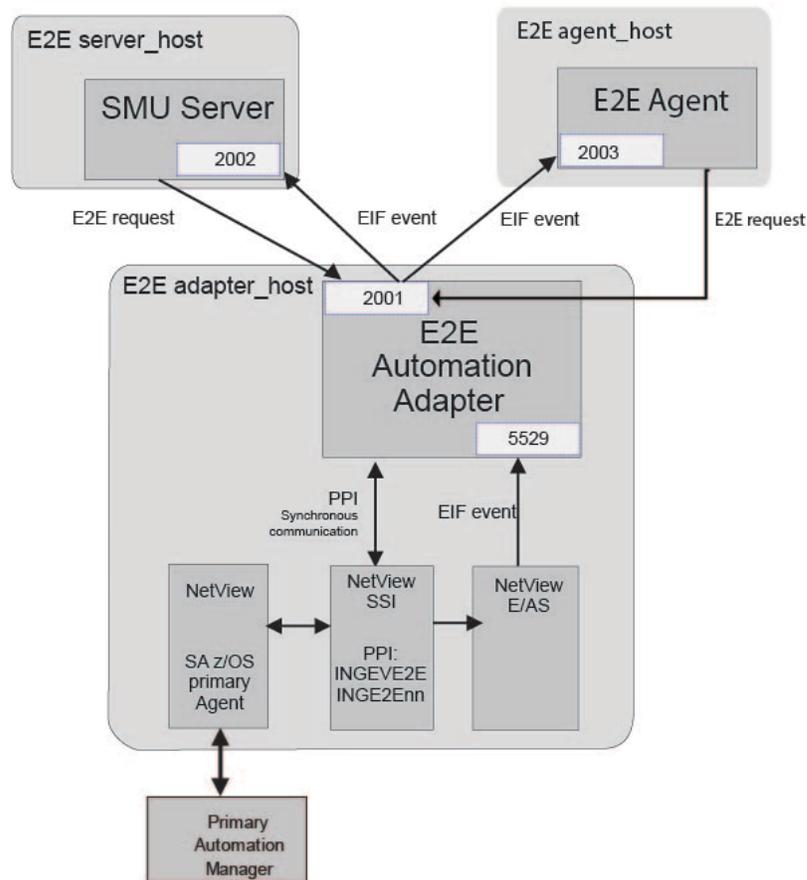


Figure 4. End-to-End Automation Adapter Communication

Synchronous Communication

As shown in Figure 4 on page 5, the automation adapter receives a request from the SMU server or from the E2E agent and schedules it via PPI communication to an SA z/OS task execution request processor that runs on the automated operator function E2EOPER or E2EOPRnn. The task execution request contains one or more end-to-end automation requests.

The program-to-program interface (PPI) enables application programs to send or receive data buffers from other application programs that are running on the same host. The PPI is an optional facility of the NetView subsystem address space and must be initialized with its PPIOPT start option.

Asynchronous Communication

The automation adapter provides an EIF event receiver and an EIF event emitter, both of which can be customized via the automation adapter master configuration file, `ing.adapter.properties`.

SA z/OS acts as an asynchronous data provider and sends SA z/OS specific events via E/AS to the EIF event receiver of the automation adapter. The automation adapter converts the events to a format understood by SMU server or by the E2E agent. The formatted events are sent to one or both targets through the EIF emitter component.

For more information about EIF, refer to https://www.ibm.com/support/knowledgecenter/SSHTQ_8.1.0/com.ibm.netcool_OMNIBus.doc_8.1.0/omnibus/wip/eifsdk/reference/omn_eif_eifreference.html.

The NetView Event/Automation Service (E/AS)

The message adapter service of the NetView event/automation service (E/AS) is used to convert and forward messages from NetView to a designated event server, such as the end-to-end automation adapter. The end-to-end automation adapter requires a separate E/AS address space for its own.

The Primary SA z/OS Automation Agent

The successful initialization of the automation adapter on a system makes the System Automation NetView/Agent to be the *primary automation agent*. INGAMS command shows this indication.

The primary automation adapter communicates with the primary SA z/OS automation manager (PAM). Commands and queries are sent and events are received from the primary automation agent.

Note: There is a small chance that if two automation adapters are started in the same SAPlex, each of the associated SA z/OS agents might become the primary agent. To avoid this, you should switch to the high availability solution for the automation adapter. For more information, see [Appendix B, “High Availability of the End-to-End Automation Adapter,”](#) on page 81.

Universal Automation Adapter (UAA)

The Universal Automation Adapter (UAA) enables Service Management Unite Automation (SMU) to monitor, operate, and automate resources running on non-z/OS systems. It connects to the remote non-z/OS systems using Secure Shell (SSH).

The Universal Automation Adapter is part of SMU. It's installed with Service Management Unite Automation and runs on the same Linux. It can connect to both SMU and E2E agent of SA z/OS. Automation domains managed by the Universal Automation Adapter are defined via the configuration utility **cfgsmu**.

The start, stop, and monitor commands for resources managed by the Universal Automation Adapter are defined in a policy. The Universal Automation Adapter policy can be edited and activated in SMU.

For more information about Universal Automation Adapter, see "Configuring access to the Universal Automation Adapter" in *Service Management Unite Automation Installation and Configuration Guide*.

Chapter 2. Concepts of End-to-End Automation

The SA z/OS automation manager version 4.1 or higher is able to automate resources between multiple SAPlexes. This capability is called end-to-end automation. Before using end-to-end automation, you must have activated it. If activated, the primary automation manager (PAM) also becomes the E2E manager and automates the resource references.

With end-to-end automation, you can

- Start and stop remote resources.
- Monitor remote resources.
- Manage cross domain dependencies between local and remote resources or between two remote resources.
- Compose business applications that have high availability across multiple automation domains.
- Manage a group of remote resources.

Overview of the Architecture

E2E Manager

The end-to-end automation manager manages dependencies between resources that are hosted on different automation domains. If end-to-end automation is activated, the PAM becomes the end-to-end automation manager. The end-to-end automation manager does not replace the local automation products on a remote domain, but it requests to start or stop resource references that are defined in the automation policy and point to remote resources on remote automation domains.

E2E Agent

The E2E agent assists the E2E manager by executing start/stop orders or collecting status of resources running on remote automation domains. The E2E agent interfaces with one or multiple remote automation adapters.

E2E Automation Adapter

The end-to-end automation adapter connects an automation domain with the E2E agent and thus with the end-to-end manager.

Remote Resource

The remote resource is a real resource on a remote automation domain. It's managed either by SA z/OS or by the Universal Automation Adapter.

Resource Reference

The resource reference is a resource of type REF that references to a real resource on a remote automation domain. The real resource is either on a remote SAPlex and managed by SA z/OS, or on a non-z/OS domain and managed by the Universal Automation Adapter.

Automation Domain

The automation domain is

- Either a System Automation SAPlex, which is connected to the E2E automation manager by the SA z/OS E2E automation adapter;
- Or a non-z/OS domain, which is connected to the E2E automation manager by the Universal Automation Adapter.

The automation domain is represented by a resource of type DMN.

DMN

DMN is the resource type of an object that represents a remote automation domain. Its entry type name can be up to eight characters long. It has the type DMN but no system qualifier, for example PLEX2/DMN.

The DMN object has attributes that refer to the actual name of the remote automation domain. This actual name can be up to 64 characters long and can be defined in the automation adapter configuration file. See also [“SA z/OS Automation Domain Name” on page 3.](#)

REF

REF is the resource type of a resource reference. Its entry type name can be up to eleven characters long. It has the type REF and the corresponding DMN-name as last qualifier instead of the system qualifier, for example CHILD2/REF/PLEX2.

The REF object has attributes that refer to the actual name of the remote resource on the remote automation domain.

Cross Domain Automation

Originally, you define a relationship between resources that all run in the same SAPlex, for example, hasParent relationship between a child and its parent. With SA z/OS 4.1, you can define a cross-sysplex relationship to resources outside of the local SAPlex. Along with APAR [OA55386](#), you can also define a cross platform relationship to resources on remote non-z/OS systems.

Note: Figure 5 on page 8 shows the cross domain relationship for SAPlex only, but can also be adapted to the domain managed by Universal Automation Adapter to model cross platform dependencies.

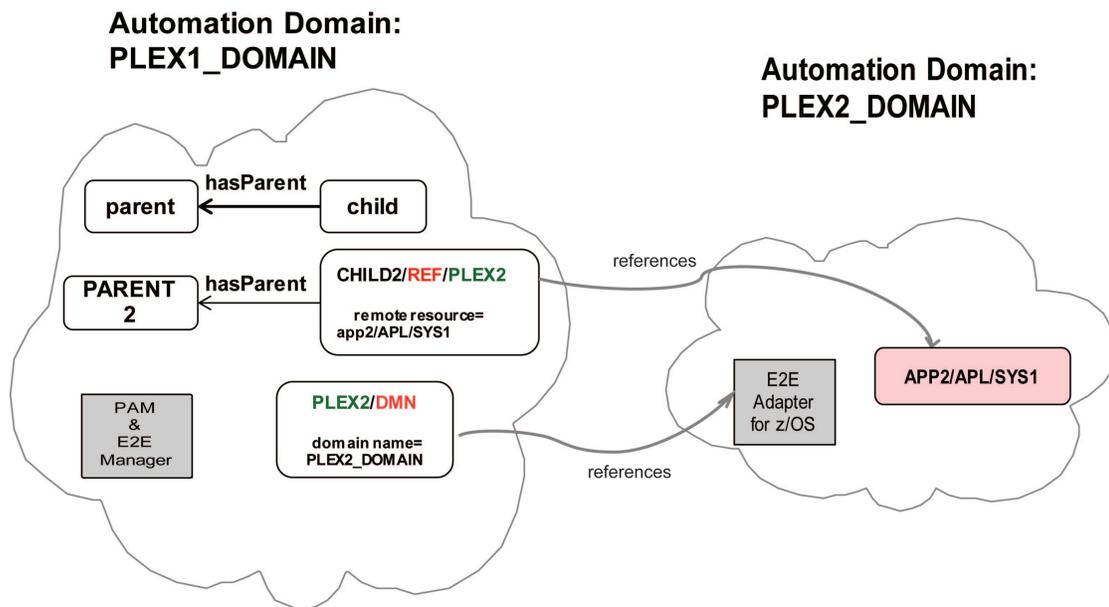


Figure 5. Cross Domain Relationship

Figure 5 on page 8 shows that each of the two automation domains is an SAPlex. The automation domain that has end-to-end automation activated is PLEX1_DOMAIN. The PAM that automates the SAPlex on this automation domain also automates the resource reference PARENT2, which points to the remote resource APP2/APL/SYS1 running on the automation domain PLEX2_DOMAIN.

An automation adapter runs on the automation domain PLEX2_DOMAIN. A TCP/IP connection is established between the E2E automation adapter and the E2E manager of automation domain PLEX1_DOMAIN.

There is an object with name PLEX2 and type DMN. It represents the automation domain PLEX2_DOMAIN. In the customization dialog, you have defined the object PLEX2/DMN and assigned the automation domain name to PLEX2_DOMAIN. This enables system automation to set the observed status of PLEX2/DMN to AVAILABLE after the remote automation adapter has successfully established a TCP/IP connection to PLEX1_DOMAIN.

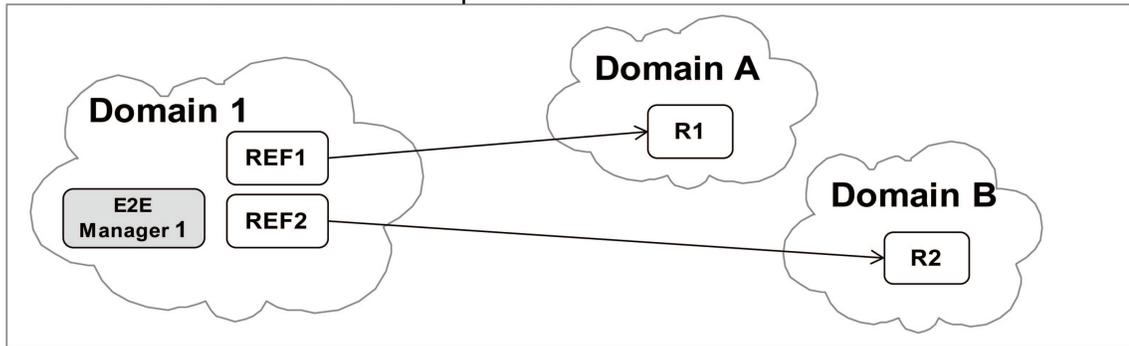
When PLEX2/DMN becomes AVAILABLE, system automation collects status of the remote resource APP2/APL/SYS1, converts this status to the status of the corresponding resource reference CHILD2/REF/PLEX2, and stores this information into the PAM. We assume APP2 is AVAILABLE and therefore the status of CHILD2/REF/PLEX2 is also AVAILABLE.

The resource PARENT2 is a local resource on automation domain PLEX1_DOMAIN. However, the E2E manager knows the hasParent relationship between the local resource PARENT2 and the resource reference CHILD2/REF/PLEX2. Therefore, if you stop the local resource PARENT2, the E2E manager would send a stop request for CHILD2 to the remote automation domain PLEX2_DOMAIN.

Multiple E2E Automation Domains

The set of automation domains connected to an E2E manager is called E2E automation domain. You can give it a unique name of your choice via parameter **INGAGT_E2EDOM** in E2E agent configuration file `ing2eagt.properties`. SMU is able to display this E2E automation view.

E2E automation domain: West coast production



E2E automation domain: East coast production

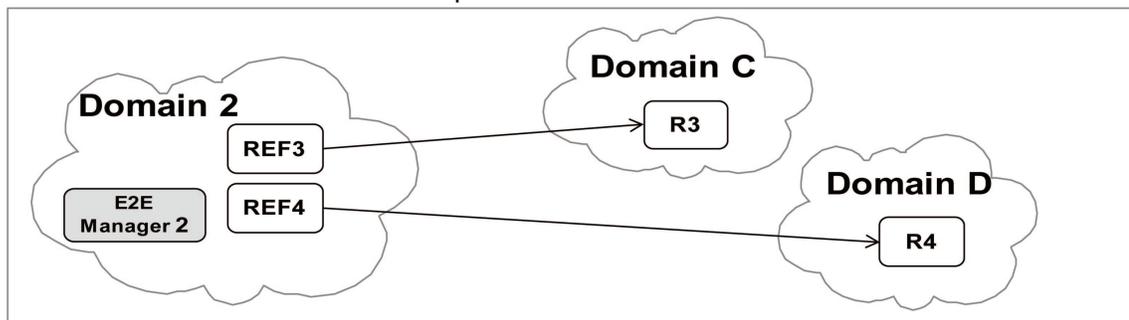


Figure 6. Multiple E2E Managers for Distinct Sets of Automation Domains

Figure 6 on page 9 shows that you might have multiple E2E managers, each controlling a unique set of automation domains. This is possible if the sets of automation domains are distinct. It means that if Domain 1 and Domain 2 are linked to the E2E Manager 1, they must not be linked to E2E Manager 2. The E2E Manager 2 has its own set of domains.

Using E2E Automation

The following topics describe how to use E2E automation.

Defining DMN and REF objects

Before starting the E2E agent, you should have defined the REF and DMN objects via the system automation customization dialog. For detailed description, see manual *IBM System Automation for z/OS Defining Automation Policy*.

Defining cross-domain relationships

You can define relationships between a resource reference and other local resources of type APL, APG, MTR, or REF. The supported relationships include MAKEAVAILABLE, MAKEUNAVAILABLE, HASPARENT, HASPASSIVEPARENT, or FORCEDOWN.

You can add a resource reference to an APG. For more information, see manual *IBM System Automation for z/OS Defining Automation Policy*.

Loading the E2E Automation configuration

After you define the DMN and REF objects and all relationships via the customization dialog, all data needed to perform end-to-end automation is stored into the normal system automation configuration file that you finally build via the customization dialog as usual.

Perform an INGAMS REFRESH to load the policy. Thus the primary automation manager knows the new resources. However, this action does not activate end-to-end automation, but prepares the activation.

Activating E2E Automation

To activate the end-to-end automation, you must start the E2E agent address space. The E2E agent may be defined as an APL and thereby activating end-to-end automation is simply starting the E2E agent APL. For details of how to set up the E2E agent, see [Chapter 4, “Customizing the End-to-End Agent,” on page 19](#).

During initialization, the E2E agent loads the REF and DMN objects from the automation policy and registers these resources to the primary automation manager and thereby activates end-to-end automation for the corresponding REF and DMN resources.

If you have not defined REF or DMN objects, end-to-end automation does not work and you will get message ING278I. Modify and refresh the automation policy again. This will automatically restart the E2E agent and activate the end-to-end automation.

Showing that E2E Automation is activated

You can find out that E2E automation is active by INGAMS command.

```
INGKYAM0                SA z/OS - Command Dialogs                Line 1      of 4
Domain Id . . : IPUFJ    ----- INGAMS -----          Date . . . : 01/22/18
Operator Id : OPER      Sysplex = AOC4PLEXBUMU          Time . . . : 14:10:19

Cmd:  A Manage          B Show Details    C Refresh Configuration  D Diagnostic

CMD System  Member      Role  Status      Sysplex  XCF Group  Release  Comm  E2E
-----
AOC4       AOC4$$$$1  PAM   READY      AOC4PLEX  INGXSGA4  V4R1M0  XCF   YES
AOC4       AOC4       AGENT READY      AOC4PLEX  INGXSGA4  V4R1M0  XCF
AOC5       AOC5$$$$1  SAM   READY      AOC4PLEX  INGXSGA4  V4R1M0  XCF
AOC5       AOC5       AGENT READY      AOC4PLEX  INGXSGA4  V4R1M0  XCF
```

Status of an automation domain

End-to-end automation can be connected or disconnected to an automation domain. You connect an automation domain to end-to-end automation when you start the corresponding E2E automation adapter

on the remote domain and the E2E agent is able to establish a TCP/IP connection to this E2E automation adapter.

There is a synchronous and asynchronous communication path from the E2E agent to the remote automation adapter. A check-health algorithm combined with heartbeat events determines periodically the observed status of the DMN. The observed status of the DMN object is AVAILABLE only if synchronous and asynchronous communication states are OK. Otherwise, the observed status is UNKNOWN or HARDDOWN if a problem was detected. It is only at the first time after a cold or warm start of the PAM that the observed status of a DMN is SYSGONE.

Status of a resource reference

The status of the resource reference represents the status of the remote resource. If a remote resource is AVAILABLE or SOFTDOWN, the resource reference should also be AVAILABLE or SOFTDOWN.

However, the statuses of the resource reference and remote resource are not always the same.

The status of a resource reference is calculated when a resource status change event for remote resource was received from the remote end-to-end automation adapter. Since the end-to-end automation adapter has its own common data model, backward calculation of resource status change events might result in different statuses of the resource reference in certain situations.

Another reason is that start/stop requests from E2EMGR and OPERATOR conflict due to different priorities. That leads to different statuses of resource reference and remote resource.

If a resource reference points to a remote resource which has the automation flag off or is suspended and the operator issues a STOP request twice on the resource reference, then the compound status of the resource reference remains INAUTO and the automation state remains in BUSY until the remote resource sends a status change event back to the E2E agent, for example because the automation flag of the remote resource is set to on again or it is resumed again.

Operational information

The operational information is not a status. It consists of a synonym and a self-explaining text. It is just an extra explanation for the current setting of observed and automation status. If the resource reference is not in the expected status, the operational information helps to understand the situation that leads to this status.

Commands supporting REF and DMN

The following system automation commands support REF and DMN objects.

Command	REF	DMN	Description
INGLIST	X	X	Display automation domains and resource references
INGINFO	X	X	Display detailed information about automation domains and resource references
INGREQ	X		Start and stop resource references
INGSET	X		Set status of a resource reference
INGSCHED	X		Creates a schedule overwrite for a resource reference

Display remote automation domains

You display all remote automation domains automated by the E2E manager via command `INGLIST */DMN`.

If the observed status of a DMN object is `AVAILABLE`, then the corresponding remote automation domain is connected and online.

You connect an automation domain to end-to-end automation when you start the corresponding E2E automation adapter and the E2E agent is able to establish a TCP/IP connection to this E2E automation adapter.

Operate remote resources

You can display the resource references via command `INGLIST */REF/*`.

Stopping a resource reference, for example via option C in `INGLIST`, may cause the E2E manager to send a stop request to the corresponding remote resource. If you have relationships defined, more start/stop requests may be sent to other local or remote resources as well eventually.

When the status of the remote resource changes on the remote domain, the status of the resource reference also changes.

Priority of start/stop request for the remote resources

You can start/stop a remote resource by issuing a start/stop request to the corresponding resource reference. End-to-end automation will send a start/stop request for the remote resource to the remote automation domain. However, for a remote SA z/OS resource, the priority of the E2E start/stop request might have a lower priority than an existing request on the remote resource. In this case, the remote PAM doesn't start/stop the remote resource until requests with higher priority are deleted.

The E2E manager always injects an `INGREQ` with source "E2EMGR", which has a lower ranking than "OPERATOR".

For more information, see [“Conflicting Goals of Different SA z/OS Automation Domains”](#) on page 13.

Display detailed information about a remote automation domain

You display detailed information of a remote automation domain via command `INGINFO`.

The following additional information is available:

```
1 E2E Agent...
  Runs on      : KEYA
Remote Domain...
2 Name        : KEYAPLEX INGXSG34
3 Operational Info : OK - The connection to the domain was successfully
  established.
4 Adapter OS   : z/OS
5 Adapter Hostname : KEYA.myhome.de.ibm.com
6 Adapter Port  : 2001
7 Adapter SSL   : false
8 Adapter Security : false
```

1 Shows the system within the SAplex where the E2E agent is running. It displays "none" if the E2E agent is not started.

2 The long name of the automation domain.

3 Operational info consists of a unique synonym such as OK and an English text.

4 The type of the operating system where the automation adapter runs.

5 The TPC/IP hostname of the operating system where the automation adapter runs.

6 The TPC/IP port of the operating system where the automation adapter runs.

7 Whether the TCP/IP connection is SSL encrypted or not.

8 Whether the automation adapter checks the user ID/password via SAF or not.

Display detailed information about a resource reference

You display detailed information of a remote automation domain via command INGINFO.

The following additional information is available:

```
1 E2E Agent...
  Runs on      : KEYA

Remote Domain...
2 Domain Name  : KEYAPLEX INGXSG34
3 Remote Resources : APPL/APL/KEYA
4 Operational Info : E2EVNT_ONLINE_OK - Resource is up and ready for
use.
```

1 Shows the system within the SAplex where the E2E agent is running. It displays "none" if the E2E agent is not started.

2 The long name of the corresponding automation domain.

3 Real name of the remote resource.

4 Operational info consists of a unique synonym such as OK and an English text.

Conflicting Goals of Different SA z/OS Automation Domains

This topic is specific to SA z/OS domains (SAplexes), but not to the domains managed by the Universal Automation Adapters.

This topic describes why the status of a resource reference can differ from the status of its remote resource. A resource reference is defined in the end-to-end automation policy which is interpreted by the end-to-end automation manager. It has a desired state which can differ from the desired state of the remote resource on the remote automation domain.

The end-to-end automation manager tries to ensure this desired state by sending requests to the PAM on the remote automation domain to start or stop the remote resource. The end-to-end automation manager does not replace the PAM on the remote domain. It is up to the PAM on the remote domain to react to those requests received from the end-to-end automation manager. Usually, the PAM on the remote domain will follow the request, but in some cases it will not fulfill the request from the end-to-end automation manager.

The following is one example of this behavior:

An operator places a stop request on "Resource A" hosted by remote automation domain. The resource reference for "Resource A" is named "ResourceReference A". This is defined in the currently active end-to-end automation policy. In this policy, the desired state has been set to available. Now there is a conflict to be solved by the PAM on the remote automation domain:

1. It has a request with source "E2EMGR" from the end-to-end automation manager to keep "Resource A" available.
2. It has another request from an operator (source "OPERATOR") to stop "Resource A".

The resulting situation is:

```
ResourceReference A
Desired: Available
Observed: Softdown

Operational Info : E2EVNR_WINNING_CONFLICT - There is a winning-request
                  conflict on the remote site. For example, stop votes
                  from source OPERATOR overrules a start vote from source
                  E2EMGR.

Compound: INAUTO
Requests: -none - Only desired state of end-to-end automation policy

Resource A
```

```
Desired: UnAvailable
Observed: Softdown

Compound: Satisfactory
Requests: 1. MakeUnAvalilable from Operator - is currently winning
          2. MakeAvailable_Only from Automation
```

When the operator removes the stop request he placed directly on "Resource A", the only remaining request on this resource is the start request from the end-to-end automation manager. The PAM of the remote automation domain will then start "Resource A" again.

User Credentials of the End-to-End Automation Manager

When the end-to-end automation manager issues requests against remote resources, it must authenticate itself to the first-level automation domains that host the remote resources. For authentication, the end-to-end automation manager uses the user credentials (user ID and password) that are specified in the credential file of the E2E agent.

If the referenced resource that is targeted by the request is hosted by an automation domain for which specific user credentials have been specified, the automation manager uses these credentials for authentication. If no specific user credentials for the domain are specified in the credential file, the automation manager uses the generic credentials that must be specified in the credential file.

If the user ID and password cannot be validated by the automation adapter on the remote domain, the resource reference will be set to HARDDOWN.

Chapter 3. Prerequisites and Environment Setup

Prerequisites

The following prerequisites are required for the automation adapter:

- SA z/OS 4.1 and IBM Tivoli® NetView® and z/OS as described in the program directory of SA 4.1
- Java™ Runtime Environment (JRE) 1.6 or higher installed on z/OS. The 32-bit Java version is required. It is recommended to use IBM Java
- The JRE Software Development Kit (SDK) if you choose to use the facility to create sample keys for test purposes
- The Event/Automation Service (E/AS) component of NetView. See the topic "Enabling Event/Automation Service" in the NetView manual *Installation: Configuring Additional Components* and [Chapter 7, "Enabling NetView Event/Automation Service \(E/AS\),"](#) on page 35.
- Full z/OS UNIX System Services (USS) with USS file system
- SSI address space with PPI function of NetView
- TCP/IP
- RACF if authentication is enabled.

Setup Java Runtime

The automation adapter is provided in Java Archive (JAR) file format and requires the 32-bit version of Java Runtime Environment (JRE) version 1.6 or higher to be installed on z/OS as the default java runtime.

End-to-end automation adapter

If the default java runtime cannot be changed, set the environment variable `E2E_JAVA_HOME` in the automation adapter configuration file `ingadapter.properties` to the required java runtime.

End-to-end agent

If the default java runtime cannot be changed, set the environment variable `E2EAGT_JAVA_HOME` in the automation agent configuration file `inge2eagt.properties` to the required java runtime.

Setup Environment Variables

The start script is located in the installation root directory and defaults to:

- Automation adapter: `/usr/lpp/ing/adapter/ingadapter.sh`
- End-to-end agent: `/usr/lpp/ing/adapter/inge2eagt.sh`

This script is assumed to be READONLY. No customization needs to be done in the script.

However, before starting this script, some USS environment variables must be set up. You find the default setting of these environment variables in the following configuration file:

- Automation adapter: `/usr/lpp/ing/adapter/ingadapter.properties`
- End-to-end agent: `/usr/lpp/ing/adapter/inge2eagt.properties`

This configuration file will be specified in the STDENV DD statement within the JCL procedure that starts the started task for:

- Automation adapter: `INGXADPT`
- End-to-end agent: `INGXEAGT`

Setup Install-Root and Custom-Root Directory

The automation adapter and/or the end-to-end agent require a directory structure within the USS file system.

There is a properties file that defines environment variables:

- Automation adapter: `ingadapter.properties`
- End-to-end agent: `inge2eagt.properties`

Within these properties files, you can define:

- install-root directory which is normally read-only
- custom-root directory which contains all customized configuration files and the log files.

You may run with the default USS file system.

However, if you have a different install-root directory and optionally you may want to distinguish between install-root and custom-root directory you must adapt the properties files to your needs.

Default USS File System

The SMP/E installation creates a *default* file structure for the various files that are associated with the automation adapter and/or the end-to-end agent, with the following subdirectories:

1. `/usr/lpp/ing/adapter`
2. `/usr/lpp/ing/adapter/config`
3. `/usr/lpp/ing/adapter/data`
4. `/usr/lpp/ing/adapter/lib`
5. `/usr/lpp/ing/adapter/ssl`

where the different paths are for:

1. Executable files, for example, the automation adapter start and stop scripts
2. Configuration files, for example, the master configuration file
3. Working files, for example, release information file or log files. This directory is initially empty.
4. JAR files and DLLs for the automation adapter
5. Security certificates. The directory is initially empty.

User-Defined USS File System for the Automation Adapter

This description assumes that you want to distinguish between install-root and custom-root directory.

Create the following sub-directories for the custom-root directory:

1. `/custom-root/adapter`
2. `/custom-root/adapter/config`
3. `/custom-root/adapter/data`
4. `/custom-root/adapter/ssl`

Copy the following files from the install-root directory into the custom-root directory.

```
cd /install-root/adapter/config
```

- `cp /ingadapter.properties /custom-root/adapter/config/ingadapter.properties`
- `cp /ing.adapter.properties /custom-root/adapter/config/ing.adapter.properties`
- `cp /ing.adapter.plugin.properties /custom-root/adapter/config/ing.adapter.plugin.properties`

- `cp /ing.adapter.ssl.properties /custom-root/adapter/config/ing.adapter.ssl.properties`

Modify the following environment variables within the configuration file `ingadapter.properties`.

```
E2E_INSTALL_ROOT=/install-root/adapter
E2E_CUSTOM_ROOT=/custom-root/adapter
E2E_JAVA_HOME=32-bit java-home-directory
```

Modify the following remaining files to your needs. For details about the parameters of the configuration files, see [Chapter 10, “Customizing End-to-End Automation Adapter,”](#) on page 45.

- `/custom-root/adapter/config/ing.adapter.properties`
- `/custom-root/adapter/config/ing.adapter.plugin.properties`
- `/custom-root/adapter/ssl/ing.adapter.ssl.properties`

The automation adapter writes output data into `custom-root/adapter/data`.

All other configuration files reside in the `install-root` directory because there is normally no need to modify them. For more details, refer to [Chapter 10, “Customizing End-to-End Automation Adapter,”](#) on page 45.

Shared USS File System

On a shared USS file system you have only one `install-root` and one `custom-root` directory for all systems in the SAPlex. Therefore you have only one set of configuration files for all systems.

Automation adapter

If you want to use a shared USS file system for all systems and you use a virtual IP (VIP) for the automation adapter remote contact IP address, you need to specify the VIP in the parameter `eez-remote-contact-hostname` in the configuration file `ing.adapter.properties`. If you do not want to use a VIP for the automation adapter remote contact IP address you let the automation adapter automatically determine the hostname.

By default, the automation adapter always writes messages and traces to the same log files regardless of which system it is currently running on. If you want separate output directories per system, you must start the automation adapter with the system name as suffix.

```
S INGXADPT ,F=START ,SX=SYS1
```

For more details, refer to [“USS Script”](#) on page 53.

In this case, the output directories will be system-dependent, for example:

1. `/custom-root/adapter`
2. `/custom-root/adapter/data/SYS1`
3. `/custom-root/adapter/data/SYS1/eez/log`

End-to-end agent

If you want separate output directories per system, you must start the end-to-end agent with the system name as suffix parameter, for example, if the system name is `SYS1`.

```
S INGXEAGT ,JOBNAME=E2EAGENT ,XID=00 ,RID=01 ,PORT=2003 ,SX=SYS1
```

In this case, the output directories will be system-dependent, for example:

1. `/custom-root/adapter`
2. `/custom-root/adapter/data/SYS1`
3. `/custom-root/adapter/data/SYS1/ing/log`

Setup TCP/IP Stack Name in a TCP/IP Multi-Stack Environment (OA58444)

You might want to run multiple TCP/IP stacks on the same system to provide network isolation for the System Automation E2E Adapter and E2E Agent and NetView E/AS.

Procedure

To use a separate TCP/IP stack name for E2E Automation and NetView E/AS, follow these settings:

1. Specify the TCP/IP stack name of your choice in both the corresponding E2E Adapter and E2E Agent configuration files. You can also specify system symbols here (see [“Support of MVS System Symbols in Configuration Files”](#) on page 18).

- E2E Adapter properties file `ingadapter.properties`

```
E2E_TCPNAME=stack_name
```

- E2E Agent properties file `inge2eagt.properties`

```
INGAGT_TCPNAME=stack_name
```

2. After you specify a specific TCP/IP stack name in the E2E Adapter configuration file, you must also run the associated NetView E/AS with the same TCP/IP stack. Add the following JCL statement in your started task procedure:

```
//STEP0 EXEC PGM=BPXTCAFF,PARM=stack_name
```

For more details, see [TCP/IP Considerations](#).

Support of MVS System Symbols in Configuration Files

The following configuration files support MVS system symbols:

- The automation adapter master configuration file, `ing.adapter.properties`
- The automation adapter plug-in configuration file, `ing.adapter.plugin.properties`
- The E2E adapter environment configuration file, `ingadapter.properties` (only `E2E_TCPNAME` parameter)
- The E2E agent environment configuration file, `inge2eagt.properties` (only `INGAGT_TCPNAME` parameter)

A system symbol starts with '&' and ends with dot '.', for example, `&SMUHOST.`. If it doesn't end with a dot, it cannot be recognized. The system symbol cannot be empty; otherwise, error message `EEZA0005I` is issued. You can specify one or more system symbols for a property. For example,

```
EIF-SEND-TO-HOSTNAME = &SMUHOST.  
EIF-SEND-TO-AGENT-HOSTNAME = &E2EAGT.
```

The E2E adapter translates the system symbols to uppercase, for example, `&HostName.` is translated to `&HOSTNAME.`. The E2E adapter also validates the existence of the system symbols. If at least one system symbol doesn't exist and therefore cannot be resolved, error message `EEZA0005I` is issued and E2E adapter initialization is stopped. There is no extra validation of the system symbols' naming convention. That means if a system symbol is too long or contains bad characters or is mixed case, it is treated as nonexistent.

A special meaning for **EIF-SEND-TO-AGENT-HOSTNAME**:

If system symbol `&E2EAGT.` contains the value `*IGNORE*`, the parameter **EIF-SEND-TO-AGENT-HOSTNAME** is treated as if the parameter was not specified at all.

Chapter 4. Customizing the End-to-End Agent

This information describes how to configure the end-to-end agent manually.

If you want to configure the end-to-end agent as part of the SA Configuration Assistant, then refer to "Base SA z/OS Configuration Using the Configuration Assistant" in *IBM System Automation for z/OS Planning and Installation*. If you plan to configure the end-to-end agent manually, then follow the instructions described afterward.

The System Automation end-to-end agent is installed by default through SMP/E into the same zFS directory as used by the automation adapter. The customization directory might also be the same as that of the automation adapter. See also Chapter 3, "Prerequisites and Environment Setup," on page 15.

It is assumed that there is a

- zFS installation path, read only: /usr/lpp/ing/adapter/zFS
- Customization path, read/write: /customization-path/

The following files reside in the customization path. Some of them need to be copied manually into the customization path and modified to your needs.

Path and File Name	To Be Copied Manually	Description
/customization-path/config/ • inge2eagt.properties	Yes	Configuration file used by the JCL procedure of the started task via DD STDENV. Must be copied manually from the installation path to the customization path and modified to your needs.
/customization-path/config/ • inge2eagt.credentials.properties • inge2eagt.ssl.properties	Yes	The credentials file is required. The SSL file is optional depending on parameter eez-remote-contact-over-ssl=false true . The files must be copied manually from the installation path to the customization path, renamed, and modified to your needs. See also the parameters INGAGT_CRED_CFG and INGAGT_SSL_CFG in inge2eagt.properties.
/customization-path/data/ • stderr.<sx><nn>.txt • stdout.<sx><nn>.txt		Standard error and output files defined and created by the JCL procedure of the started task.
/customization-path/data/<sx>/ing/logs/ • ing<nn>MsgFlat.log • ing<nn>MsgXml.log • ing<nn>TraceFlat.log • ing<nn>TraceXML.log		Message and log trace files within the output directory. Flat files and files in XML format. Created by the end-to-end agent.

Path and File Name	To Be Copied Manually	Description
/customization-path/ssl/ • ing.ssl.e2eagent.jks • ing.ssl.truststore.jks	Yes	Keystore file used for secure connections (SSL) to the remote E2E automation adapters. These files are not used if eez-remote-contact-over-ssl=false .

<sx> is the value specified in the JCL procedure via parameter **SX**. If specified, the output file may for example look like `stderr.SYS1.01.txt`. If not specified, it does not appear at all in the file name, for example `stderr.01.txt`.

<nn> is a two character value specified in the JCL procedure via the parameter `RID=nn`.

Copy and Customize the Start Procedure

You can use the JCL procedure `INGXEAGT` to start the end-to-end agent. This procedure is provided in the System Automation sample library. It starts the end-to-end agent via the USS shell script `/usr/lpp/adapter/inge2eagt.sh`. The shell script is read-only.

To customize the end-to-end agent, customize the parameters in the configuration file `inge2eagt.properties`. The `DD STDENV` in the JCL procedure points to this configuration file.

Copy the sample JCL procedure `INGXEAGT` from `ING.SINGSAMP` into your proclib. Rename it to a name of your choice and customize it. Follow the comments in the sample and specify the installation and customization paths in parameter **DIRI** and **DIRC**.

Below you find a sample z/OS command that starts the end-to-end agent as a started task:

```
S INXEAGT, JOBNAME=E2EAGENT, XID=00, RID=01, PORT=2003, OPT=DEBUG, SX=SYS1
```

For details about the parameter list or about the high availability of the end-to-end agent, see [Chapter 5, "Operating the End-to-End Agent,"](#) on page 25.

Copy and Customize the Environment Configuration File

The JCL procedure `INGXEAGT` has a `DD STDENV` that defines end-to-end specific environment variables. Copy the sample configuration file `/usr/lpp/ing/adapter/config/inge2eagt.properties` into your customization directory and customize it to your needs. Specify the configuration file in the start procedure `DD STDENV`.

The following parameters can be customized:

Name	Default Value	Description
<code>INGAGT_XID</code>	Empty string	Two-byte XCF group ID of the associated SA/NetView agent. This parameter might be overwritten by JCL procedure <code>INGXEAGT</code> .
<code>INGAGT_RID</code>	1	Two-byte suffix for the generated PPI receiver queue name used by the E2E agent. For example, suffix 01 will become receiver ID <code>INGJVM01</code> . This parameter might be overwritten by JCL procedure <code>INGXEAGT</code> .
<code>INGAGT_PORT</code>	2003	EIF receiver port number. This parameter might be overwritten by JCL procedure <code>INGXEAGT</code> .

Table 4. Parameters for *inge2eagt.properties* (continued)

Name	Default Value	Description
ING_HOST		IP address or hostname of the E2E agent. Default is automatically determined by USS command hostname. You have to specify this hostname if you use VIPA.
INGAGT_INSTALL_ROOT	/usr/lpp/ing/adapter	The root directory of the installation directory.
INGAGT_CUSTOM_ROOT	/usr/lpp/ing/adapter	The root directory of the customization directory. It contains logger output files, config files, and SSL related files. The end-to-end agent writes into this directory.
INGAGT_SSL_CFG	inge2eagt.ssl.properties	The name of the SSL configuration file residing in directory <code>INGAGT_CUSTOM_ROOT/config</code> . The SSL configuration file is required when the E2E automation adapter on at least one remote domain was started with <code>eez-remote-contact-over-ssl=true</code> .
INGAGT_SSL_VERSION (introduced in OA58444)	TLS	The version of the SSL protocol that is used for SSL communication, for example, TLSv1.2. If you change it here, you must change it for the E2E adapter (<code>E2E_SSL_VERSION</code>) and for the SMU server (<code>com.ibm.eez.adapter.ssl.version</code>) as well.
INGAGT_USRJP (introduced in OA58444)		Here you can add user specific Java system properties, for example, SSL debugging options. <code>-Djavax.net.debug=ssl:handshake:verbose:keymanager:trustmanager</code> This parameter is useful if you need to debug SSL handshake exceptions.
INGAGT_USRCP (introduced in OA58444)	/u/mydir/my.jar	Here you can add user specific classpath if needed.
INGAGT_CRED_CFG	inge2eagt.credentials.properties	The name of the credentials configuration file residing in directory <code>INGAGT_CUSTOM_ROOT/config</code> .
INGAGT_E2EDOM	E2E Domain	You can specify the name of your choice for the E2E domain.
INGAGT_MAX_CONS	9	Maximum number of E2E automation adapter requests that can be sent concurrently to remote domains.
INGAGT_PPI_TIMEOUT	30	Timeout in seconds when the E2E agent waits for a response from NetView via PPI.
INGAGT_CON_TIMEOUT	45	Timeout in seconds when the E2E agent waits for a response from a remote E2E automation adapter.
INGAGT_CKH_INTERVAL	300	Time interval in seconds. Used to send a check health request periodically to the remote E2E automation adapter in order to detect status of the remote domain.

Table 4. Parameters for *inge2eagt.properties* (continued)

Name	Default Value	Description
INGAGT_OPER	E2EAGT	System Automation auto operator function dedicated for the end-to-end agent event sink.
INGAGT_JAVA_HOME		You can specify which java installation to be used. It must contain the Java installation path. It is assumed that Java is installed into the following directories: <ul style="list-style-type: none"> • \$JAVA_HOME/lib • \$JAVA_HOME/bin It is required to use the 32-bit version of Java 1.6 or higher.
INGAGT_CODEPAGE	cp1047	Codepage used to translate ASCII to EBCDIC.
INGAGT_JAVA_XMX	256	Maximum heap size of the java process.
INGAGT_TCPNAME (introduced in OA58444)	Empty string	The name of the TCP/IP stack where you want to isolate E2E Automation and NetView E/AS. System symbols are supported here.

Create the Credentials Configuration File

The end-to-end agent credentials file is a USS file. The sample files resides in `/usr/lpp/ing/adapter/config/inge2eagt.credentials.properties`.

Copy the credentials file to `/customization-path/config` and adapt it to your needs.

The end-to-end agent credentials file contains for each automation domain three items: domain, user ID, password. The end-to-end agent needs these items to log on to a remote domain.

The user ID and password is a SAF defined user ID (for example, RACF). It will be passed over the connection between the end-to-end agent and the end-to-end automation adapters. User ID and password is used for the end-to-end automation adapter authentication checking and for third party System Automation authorization checking.

The parameters **user-generic** and **password-generic** define the default user ID and password used for all domains if not defined otherwise.

Each domain that is defined in the policy needs to have either a specific or a generic credential statement. For each specified user ID, you must also specify a password. The user ID must be a SAF user ID that is valid on the remote z/OS domain and it must have an OMVS segment defined.

The password might be AES-encrypted or clear text. You can use the script `ingencrypt.sh` to encrypt the clear test password to AES.

You can use the SMU Configuration Dialog to maintain the credentials file. In any other case, you modify the credentials file manually.

Here is an example:

```
#--
#-- Required pairs
#--
user-generic=<e2e-agent-status-operator>
password-generic=<password>
#--
#-- Optional pairs
#--
# user-<DOMAINNAME>=<SPEC_USERID>
# password-<DOMAINNAME>=<SPEC_PASSWORD>
```

```
user-SYSPLEX INGXSGNN=<userid>
password-SYSPLEX INGXSGNN=<password>
```

Set Up the SSL Configuration

When you plan to use SSL connections between the end-to-end automation adapter and end-to-end agent, you need to set up the SSL configuration file `/customization-path/config/inge2eagt.ssl.properties`.

You can set up the SSL configuration by one of the following techniques:

- Create certificates residing in RACF.
- Create keystore and truststore files residing in USS file system.

Customize the SSL configuration file using the RACF repository

For a detailed description of how to create certificates and a corresponding keyring in the RACF repository, see [Chapter 13, “Setting Up E2E Automation RACF Keyrings \(OA58444\),”](#) on page 61.

Customize the SSL configuration file residing in the USS file system

The file `/customization-path/config/inge2eagt.ssl.properties` should point to the keystore and truststore to be used by the end-to-end agent. The directory `/customization-path/ssl/` should contain

- keystore with the certificate for the end-to-end agent
- truststore with the public keys of all end-to-end automation adapters

You must complete the following extra steps:

1. Customize the SSL configuration file.
 - Copy and modify the SSL configuration file `inge2eagt.ssl.properties`.
 - Encrypt the passphrase.
2. Create the keystore file `ing.ssl.e2eagent.jks` and copy it into the `/customization-path/ssl/` directory.
3. Import the public key from all relevant keystores into the truststore `ing.ssl.truststore.jks` and copy it into the `/customization-path/ssl/` directory.

For detailed information about each step above, see following sections.

Customize the SSL Configuration File

Copy and modify the SSL configuration file

Copy the SSL configuration file `inge2eagt.ssl.properties` from the installation directory `/<installation-path>/config/` to the customization directory `/customization-path/config/inge2eagt.ssl.properties`.

Modify at least the **eez-ssl-keystore-password** parameter: `eez-ssl-keystore-password = passphrase`.

It is assumed that there is one password for the truststore and for all keystores. The default password is `passphrase`. You can modify it to your needs, whether to specify it in clear text or encrypt the password.

Encrypting the passphrase

You might want to hide the clear text password when you save it into the SSL configuration file `inge2eagt.ssl.properties`. For that purpose, you specify the encrypted password instead of the clear text password.

To generate an encrypted password, use the encryption tool: `/<installation-path>/ingencrypt.sh -encrypt your_password`.

Copy the encrypted password from the command output and paste it into the variable `eez-ssl-keystore-password`.

Create the Keystore File

Create a keystore file `ing.ssl.e2eagent.jks` with the certificate of your choice. The keystore must be a Java keystore file with the extension `.jks`.

Copy the keystore file into the following SSL directory: `/customization-path/ssl/ing.ssl.e2eagent.jks`.

Note: For test purposes, you can create a self-certified certificate. For how to create self-certified certificate, see [“How to Create Keystore and Truststore for Use by SSL”](#) on page 51. Follow the instructions there and rename the file names as needed.

Import the Public Key From the Keystore into the Truststore

The truststore should contain all public keys of all end-to-end automation adapters that report to the end-to-end agent and the public key of the end-to-end agent itself. Import the end-to-end agent certificate from the keystore `ing.ssl.e2eagent.jks` into truststore `ing.ssl.truststore.jks`.

Copy the truststore `ing.ssl.truststore.jks` in the following directory: `/customization-path/ssl/ing.ssl.truststore.jks`.

The truststore also needs to be copied to all systems hosting an end-to-end automation adapter that will connect to the end-to-end agent. For how to export a self-certified certificate and how to import it into the truststore, see *Service Management Unite Automation Installation and Configuration Guide*.

Configuring the End-to-End Automation Adapter for Use by the End-to-End Agent

The end-to-end automation adapter is able to connect to SMU and to the end-to-end agent simultaneously.

If you want an end-to-end automation adapter to connect to the end-to-end agent, you need to configure the end-to-end automation adapter. Add the following parameters to the configuration file `ing.adapter.properties` of the end-to-end automation adapter:

- `eif-send-to-agent-hostname = E2EAgent_host`
- `eif-send-to-agent-port = 2003`

The value specified in parameters **`eif-send-to-agent-hostname`** and **`eif-send-to-agent-port`** is the TCP/IP hostname and port of the end-to-end agent. The corresponding end-to-end agent parameters are configured by `INGAGT_HOST` and `INGAGT_PORT` in file `inge2eagt.properties`.

Chapter 5. Operating the End-to-End Agent

This chapter describes how to start and stop the end-to-end agent and how to make it highly available via a System Automation policy.

Starting the End-to-End Agent

The end-to-end agent is a USS process started via started task using BPXINIT. You use the sample JCL procedure ING.SINGSAMP(INGXEAGT) to start the end-to-end agent.

The following sample z/OS command starts the end-to-end agent as a started task:

```
S  INGXEAGT ,JOBNAME=E2EAGENT ,XID=00 ,RID=01 ,PORT=2003 ,OPT=DEBUG ,SX=SYSA
```

Description of the parameters:

XID

The two-character XCF group ID of the associated System Automation/NetView agent.

RID

The two-character suffix of the generated PPI receiver ID, for example, INGJVM01.

PORT

The EIF receiver port for the EIF events sent by the end-to-end automation adapters.

SX

May be used for shared zFS. It causes the end-to-end agent to write log files into a data directory that depends on the value of SX. It might be the system name or any other user specified differentiators.

For example, stderr.SYSA01.txt and the log files are written in /customization-path/data/SYSA/ing/logs/.

OPT

This parameter is for debugging only.

OPT=DEBUG or OPT=-d	Switch ON Java logger.
OPT=TRACE or OPT=-t	Switch ON shell trace.
OPT=TRCDBG or OPT=-dt	Switch ON both shell trace and Java logger.
OPT=OFF	Switch OFF all options.

Note: XID must be specified to connect to the System Automation/NetView agent. You can only omit XID if you use the default XCF group ID, which is blank.

RID and PORT have meaningful defaults (RID=01, PORT=2003) and may be omitted. SX and OPT are optional. OPT is used only for IBM Service.

Stopping the End-to-End Agent

You can stop the end-to-end agent from the NetView console via command INGE2E STOP_E2EAGT.

Self-Terminating the End-to-End Agent

After the PAM is stopped, System Automation will terminate the E2E agent's USS address space outside of automation. The E2E agent will be stopped outside of automation and therefore thresholds will be triggered for the E2E agent APL.

The ING377I message will be issued with return code 236 and reason text:

```
ING377I The E2E agent service INGRE2E failed with RC=236, REASON=E2E agent terminated itself since PAM was not available
```

After the PAM starts again, it detects that the E2E agent APL is unavailable and therefore starts the E2E agent APL again.

Cleaning Up the End-to-End Agent

If the end-to-end agent USS process has been terminated abnormally, it has no chance to perform cleaning up allocated resources. Use the following command to clean up the end-to-end agent: INGE2E CLEANUP_E2EAGT.

Recycling the End-to-End Agent

The E2E agent needs to be recycled when

- NetView recycles
- NetView SSI address space recycles
- PAM recycles
- PAM moves

While the E2E agent is being recycled, it will first be stopped. This sets the status of the DMN and its REF resources to UNKNOWN or WASAVAILBLE. While the E2E agent is being initialized, the status of the DMN and REF resources keep the same as they were previously stored in the PAM. After the E2E agent is able to establish connection to the DMN, the status of the DMN and its REF resources will be updated.

If you refresh the automation manager configuration file, for example, via INGAMS REFRESH, the E2E agent will be recycled automatically by System Automation when the REF or DMN configuration has changed.

However, recycling the E2E agent due to REFRESH will not cause to set the status of the DMN and its REF resources to UNKNOWN or WASAVAILBLE. They keep as they are and will be updated successively.

Checking If the End-to-End Agent Is Running

You can use INGAMS command to display all members of the SApdex.

If the PAM has indication YES in column E2E, then the end-to-end agent is running on this system and end-to-end automation is active. See also [“Showing that E2E Automation is activated”](#) on page 10.

Dependencies of the End-to-End Agent

To start the end-to-end agent, some dependencies must be fulfilled:

- OMVS and TCP/IP must be running.
- USS file system must be accessible where the end-to-end agent is installed and customized.
- PAM must be available and communication to the PAM works.

Making the End-to-End Agent Highly Available

The end-to-end agent runs as a started task and should be defined as APL within a System Automation policy. It must run once per SApdex. For that purpose, a new application category E2EAGT has been introduced.

A sample policy is available for the end-to-end agent.

The customer may import the following PDB entries within *E2E via component 'E2E Automation Agent':

Table 5. E2E Sample Policy - E2E Agent Component	
PDB Entry Name	Type
E2EAGENT	APL
E2EDVIPA	APL
C_DVIPA	APL, (only a placeholder here, the original values can be found in *BASE)
E2EAGENT	APG, system group to contain applications E2EAGENT and E2EDVIPA
E2EAGENT_X	APG, Sysplex move group – serial mode
E2EAGENT_AUTOOPS	AOP, Auto operator INGAGT maps on AUTAGT

Since E2EAGENT_X is a move group between systems of the SApdex, it is necessary for the end-to-end automation adapter to connect to the end-to-end agent via a virtual IP address (VIPA). A VIPA stays the same, even though the application (end-to-end agent) using it can be moved from one system to another.

The E2EDVIPA application makes use of class C_DVIPA, which is delivered with the *BASE add-on policy in the 'Dynamic Virtual IP Addressing (DVIPA)' component.

After you import the PDB entries, additional adaptations are needed:

- Change the XCF group ID. Change the startup parameter **XID=00** in the E2EAGENT application according to your needs.
- Change the startup parameter of E2EDVIPA application using the correlated IP stack and the DVIPA address.

Note: This DVIPA address has to be specified also in the **INGAGT_HOST** parameter of `inge2eagt.properties` file for the E2E agent, and in the **EIF-SEND-TO-AGENT-HOSTNAME** parameter of `ing.adapter.properties` file for each E2E automation adapter.

The automation policy has definitions for the broken E2E agent and its IEF450I abend message. It automatically performs cleanup recovery of the end-to-end agent.

The automation policy exploits the following messages to set the status of the corresponding end-to-end agent APL:

Message	SA Agent Status
INGX9632I System Automation E2E agent jobname terminates: <i>reason</i>	No status change and only message capturing is done.
INGX9633I System Automation E2E Agent <i>jobname</i> initialization complete	UP
INGX9634I System Automation E2E agent <i>jobname</i> user ID or password is not correctly specified or the credential file cannot be loaded from <i>path</i> .	HALTED
INGX9874I System Automation Agent Java Adapter <i>jobname</i> is active.	ACTIVE
INGX9876I System Automation Agent Java Adapter <i>jobname</i> has been stopped.	AUTODOWN
INGX9877I System Automation Agent Java Adapter <i>jobname</i> has been stopped due to failure, <i>rc=return code</i> .	BROKEN
ING377I The E2E agent service xxx failed with RC= <i>retcode</i> , REASON= <i>reason_text</i>	HALTED

Chapter 6. Quick Startup of End-to-End Automation Adapter

This information describes how to configure the end-to-end automation adapter manually.

If you want to configure the end-to-end automation adapter as part of the SA configuration assistant, then refer to "Base SA z/OS Configuration Using the Configuration Assistant" in *IBM System Automation for z/OS Planning and Installation*. If you plan to configure the end-to-end automation adapter manually, then follow the instructions described afterward.

Default parameters are used as much as possible to explain mandatory configurations but are not explained here.

A few abbreviations are used in this information:

- ING.SING* - SA z/OS target data sets as provided by SMP/E installation
- CNM.* - NetView target data sets as provided by SMP/E installation
- custom-root - User configuration directory of the end-to-end automation adapter
- install-root - Installation root directory of the end-to-end automation adapter (default installation directory is /usr/lpp/ing).

Overview of Quick Startup

The following configuration items need to be completed for a quick startup of the end-to-end automation adapter:

1. SA z/OS automation policy
2. SA z/OS initialization member INGXINIT
3. SA z/OS end-to-end automation adapter sample JCL procedure
 - ING.SINGSAMP(INGXADPT)
4. SA z/OS end-to-end automation adapter USS configuration files
 - /custom-root/adapter/config/ingadapter.properties
 - /custom-root/adapter/config/ing.adapter.properties
 - /custom-root/adapter/config/ing.adapter.plugin.properties
5. NetView subsystem (SSI) JCL procedure for PPI enablement
6. NetView Event Automation service (E/AS)
 - CNM.CNMSAMP(IHSAEVNT)
 - CNM.SCNMUXCL(IHSAINIT)
 - CNM.SCNMUXCL(IHSAMCFG)
7. Start and verify end-to-end automation adapter.
8. Follow-on activities.

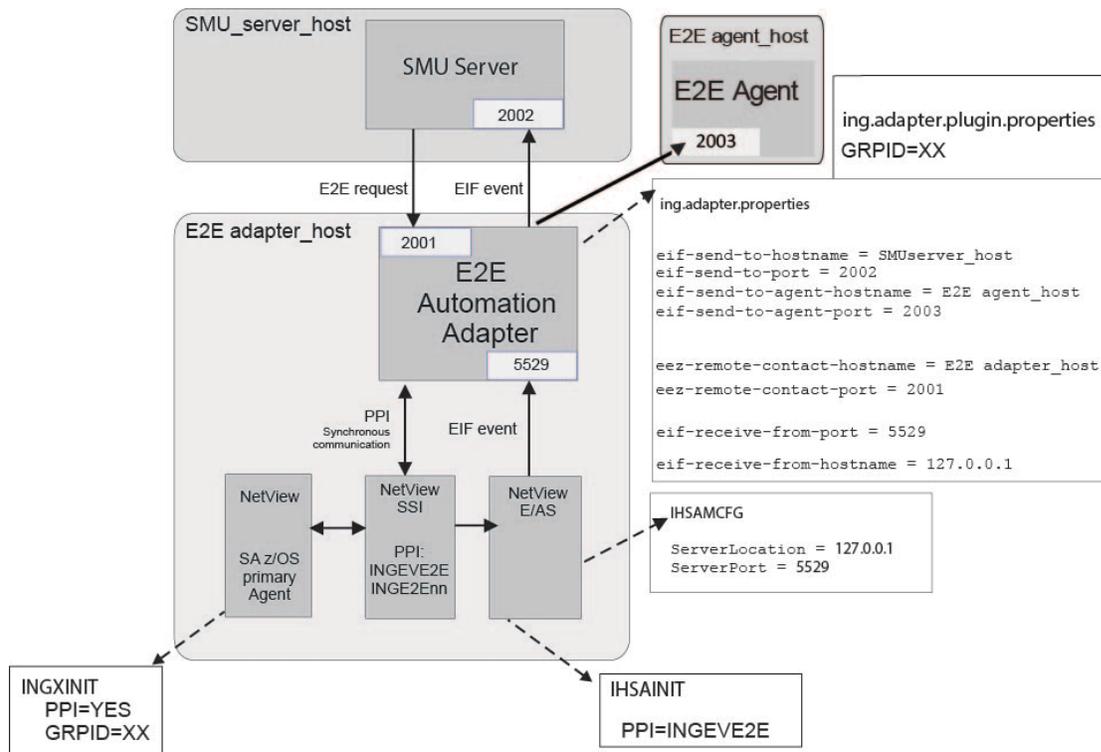


Figure 7. End-to-end automation adapter communication

SA z/OS Automation Policy

Import the Automation Operators (AOP) entry E2EADPT_AUTOOPS (Automated Functions for E2E) from the *E2E best practices policy, link it to the systems where you want to start the end-to-end automation adapter and build the automation configuration file.

For details, refer to “SA z/OS Automated Operator Functions” on page 37.

SA z/OS Initialization Member INGXINIT

Enable PPI communication within DSIPARM member INGXINIT by setting:

```
PPI=YES
```

For details, refer to “SA z/OS Initialization Member INGXINIT” on page 37.

SA z/OS Sample JCL Procedure INGADPT

The sample procedure INGADPT within SA z/OS samples data set allows you to start the automation adapter as a batch job or started task.

Copy ING.SINGSAMP (INGADPT) and follow the customization instructions within the sample job. Set JCL symbols DIRI and DIRC:

```
// DIRI='<install-root>/adapter'  
// DIRC='<custom-root>/adapter'
```

For more details on how to start the automation adapter, refer to [Chapter 11, “Operating the End-to-End Automation Adapter,”](#) on page 53.

SA z/OS USS Configuration Files (Automatically)

Logon to the NetView console on the system where you plan to run the end-to-end automation adapter.

Issue the command:

```
NETVASIS INGE2E PREPARE IPATH=/install-root/adapter
                        CPATH=/custom-root/adapter
                        SERVER=hostname
                        JAVA=/32bit_java_home
```

IPATH

Install root directory of the end-to-end automation adapter, for example, /usr/lpp/ing/adapter

CPATH

Custom root directory for read/write, for example, /local/ing/adapter.

Subdirectories will automatically be created.

SERVER

SMU server hostname

JAVA

32 bit java home directory, for example, /local/java/a60/J6.0

You may omit this parameter if the default java runtime is already a 32-bit java version 1.6.

For example:

```
NETVASIS INGE2E PREPARE IPATH=/usr/lpp/ing/adapter
                        CPATH=/u/ing/adapter
                        SERVER=SMULINUX
                        JAVA=/local/java/a60/J6.0
```

This command completes the following:

1. Creates the subdirectories within the custom-root directory
2. Copies all necessary configuration files
3. Updates the configuration files within the input parameters for quick startup.

You may want to browse and check the recently created configuration files via command:

```
NETVASIS INGE2E DISPCFG CPATH=/u/ing/adapter
```

Now you are done with the configuration of the USS configuration files.

For more details about the command INGE2E PREPARE, see [“INGE2E” on page 100](#).

SA z/OS USS Configuration Files (Manually)

If you do not wish to use the command INGE2E PREPARE you may perform manually as an alternative the customization steps of the USS files. If you have already successfully executed command INGE2E PREPARE, leave this section.

You can create the following subdirectories for the custom-root directory:

1. /custom-root/adapter
2. /custom-root/adapter/config
3. /custom-root/adapter/data
4. /custom-root/adapter/ssl

You copy the following files from the install-root directory into the custom-root directory.

```
cd /install-root/adapter/config
```

- cp /ingadapter.properties /custom-root/adapter/config/ingadapter.properties

- cp /ing.adapter.properties /custom-root/adapter/config/ing.adapter.properties
- cp /ing.adapter.plugin.properties /custom-root/adapter/configuring.adapter/plugin.properties
- cp /ing.adapter.ssl.properties /custom-root/adapter/config/ing.adapter.ssl.properties

You modify the following environment variables within the configuration file `ingadapter.properties`.

```
E2E_INSTALL_ROOT=/install-root/adapter
E2E_CUSTOM_ROOT=/custom-root/adapter
E2E_JAVA_HOME=32-bit java-home-directory
```

You may modify the files that you have copied to your needs.

For details about the parameters of the configuration files, refer to [Chapter 10, “Customizing End-to-End Automation Adapter,”](#) on page 45. The minimum changes are as follows:

- /custom-root/adapter/config/ing.adapter.properties
 - eif-send-to-hostname=SMUser_server_host
 - plugin-configfile-sa4os=/custom-root/adapter/config/ing.adapter.plugin.properties
- /custom-root/adapter/config/ing.adapter.plugin.properties
 - GRPID=xx

This parameter must match the GRPID of the DSIPARM member INGXINIT.

NetView SSI JCL Procedure

Enable PPI for the NetView subsystem interface address space (SSI) if not already done.

Access your NetView SSI startup procedure and enable the PPI infrastructure via the following parameter:

```
//      PPIOPT='PPI'
```

For more information, consult NetView sample job CNM.CNMSAMP(CNMSJ010).

NetView Event/Automation Service (E/AS)

You must enable a dedicated E/AS address space for the end-to-end automation adapter. The E/AS is the broker between the SA z/OS NetView agent and the end-to-end automation adapter.

Perform the following tasks:

- Allocation of the SCNMUXCL configuration data set
- Copy the SA z/OS specific message adapter format file ING.SINGSAMP(INGMFMTE)
- Copy and configure the E/AS initialization file CNM.SCNMUXCL(IHSAINIT)
- Copy and configure the E/AS message adapter configuration file CNM.SCNMUXCL(IHSAMCFG)
- Configure the E/AS started task JCL CNM.CNMSAMP(IHSAEVNT)

For detailed instructions, refer to [Chapter 7, “Enabling NetView Event/Automation Service \(E/AS\),”](#) on page 35.

End-to-End Automation Adapter Startup

You start the end-to-end automation adapter via the sample procedure INGADPT:

```
/S INGADPT,F=START
```

For more details on how to start the automation adapter, refer to [Chapter 11, “Operating the End-to-End Automation Adapter,”](#) on page 53.

Before you start end-to-end automation adapter, you must make sure that all previous customization steps are in effect:

1. If you have changed the NetView DSIPARM member INGXINIT then stop the SA z/OS communication task:

```
STOP TASK=INGPXDST
```

Optionally, you may shutdown NetView.

2. If you have adapted the NetView SSI startup JCL, stop the SSI address space.
3. Refresh the corresponding Primary Automation Manager with the modified SA z/OS configuration file (socntl_file). You may use one of the following commands:

INGAMS option C "Refresh Configuration"

```
F pam_jobname,REFRESH
F pam_jobname,REFRESH,socntl_file
```

4. If you have adapted the NetView SSI start procedure then recycle SSI.
5. Start the SA z/OS communication task:

```
START TASK=INGPXDST
```

Optionally recycle NetView and make sure that the NetView SSI is started as well.

6. Start the configured NetView event automation service address space (E/AS):

```
S IHSAEVNT
```

You may browse the Netlog and look for the following completion message:

```
ISH0124I Message Adapter task initialization complete.
```

7. Start the SA z/OS end-to-end automation adapter:

```
S INGXADPT,JOBNAME=E2EADPTR,F=START
```

Browse the Netlog and look for the following completion messages:

```
EEZA0102I The adapter is ready
EEZA0118I The connection to the server hostname: 2002 has been established
```

Verifying End-to-End Automation Adapter

Ensure that the E/AS address space is running that you have configured for the automation adapter. It is not necessary to have a running end-to-end automation adapter. The NetView operator that runs the VERIFY command should be allowed to access USS files.

On the NetView console, enter command:

```
NETVASIS INGE2E VERIFY JOBEAS=eas-jobname CPATH=/custom-root/adapter
```

For more details, see [“INGE2E” on page 100](#).

Following Up

You might want to perform the following list of customization tasks after successful "Quick Startup":

- SSL configuration
- End-to-End Adapter Security

- End-to-End Adapter High Availability using SA z/OS

Chapter 7. Enabling NetView Event/Automation Service (E/AS)

This information describes how to configure a dedicated end-to-end E/AS manually.

If you want to configure the E/AS as part of the SA Configuration Assistant, then refer to "Base SA z/OS Configuration Using the Configuration Assistant" in *IBM System Automation for z/OS Planning and Installation*. If you plan to configure the E/AS manually, then follow the instructions described afterward.

The end-to-end automation adapter uses the NetView message adapter service of E/AS as a gateway between the primary system automation agent and the end-to-end automation adapter.

The NetView message adapter service of E/AS converts incoming messages from NetView into EIF events and forwards them to the automation adapter.

There is additional information about how to enable the NetView message adapter service of E/AS for end-to-end automation. For more details, see the topic "Setting Up UNIX System Services for the NetView Program" in *Tivoli NetView for z/OS Installation: Configuring Additional Components*.

Starting the E/AS Address Space

The event/automation service may be started with a JCL procedure from an MVS system console.

A sample JCL procedure for starting E/AS is provided in member IHSAEVNT within NetView's sample library.

Customize the Sample Procedure

You have to customize the sample procedure IHSAEVNT with the following parameters:

INITFILE

Global initialization file (Default: IHSAINIT)

MSGCFG

Message adapter configuration file (Default: IHSAMCFG)

PPI

Must be INGEVE2E

If you want to use your own PPI receiver ID different to INGEVE2E, you have to define it to SA z/OS as an advanced automation option before starting SA z/OS. For more details, see "[SA z/OS Advanced Automation Options](#)" on page 38.

The concatenation of data sets for DD IHSSMP3 in the sample procedure must contain both E/AS initialization files mentioned here and the end-to-end specific message adapter format file which is provided in ING.SINGSAMP(INGMFMTE).

If you want to use your own data set for DD IHSSMP3 allocate it with the following data set characteristics:

```
RECFM=VB,RECL=516
```

As an initial size you might allocate the following number of tracks:

```
Primary Quantity . . 10
Secondary Quantity . 2
Directory Blocks . . 5
Block Size . . . . . 27998
```

Copy members IHSAINIT and IHSAMCFG from the NetView sample library into your data set.

Copy member INGMFMTE into your data set or add ING.SINGSAMP to the concatenation of DD IHSSMP3.

Configuring the Global Initialization File (Default: IHSAINIT)

Customize which E/AS tasks should be started. Only the message adapter is required.

```
# Tasks not started at initialization
NOSTART TASK=ALERTA
#NOSTART TASK=MESSAGEA
NOSTART TASK=EVENTRCV
NOSTART TASK=ALRTTRAP
NOSTART TASK=TRAPALRT
NOSTART TASK=MESSAGEC
NOSTART TASK=ALERTC
```

Configuring the NetView Message Adapter Service (Default: IHSAMCFG)

The automation adapter is listening to port 5529 and is located on the same z/OS system as the E/AS:

```
ServerLocation=127.0.0.1
ServerPort=5529
```

Specify the name of the NetView message adapter format file:

```
AdapterFmtFile=INGMFMTE
```

Establish a permanent IP connection between the automation adapter and the E/AS address space:

```
ConnectionMode=connection_oriented
```

Switch off the caching of events:

```
BufferEvents=no
```

Chapter 8. Customizing System Automation Agent

This information provides details of how to customize NetView and SA z/OS configuration members for the automation adapter.

SA z/OS Initialization Member INGXINIT

Set the following automation adapter-specific parameters in the SA z/OS initialization member `ING.DSIPARM(INGXINIT)`.

It is required to enable PPI communication by setting the following parameter:

```
PPI=YES
```

Ensure that the value of the parameter `GRPID` is the same as specified in the automation adapter plug-in configuration file `/custom-path/adapter/config/ing.adapter.plugin.properties`.

Optionally, you may set the PPI buffer queue limit (`PPIBQL`) higher if needed.

You may keep the default and if you receive one of the following error messages, increase the buffer queue limit.

```
PPIBQL=nnnn|1500
```

All input requests from the end-to-end automation adapter flow into this PPI queue. The buffer queue limit is exceeded (that is, the queue limit is too small), then:

- the automation adapter might not be able to send any further requests to the SA z/OS Agent, and the agent issues a JNI exception with the return code 1735:

```
INGX9820E JNI function ingippi failed with return code 1735
```

- the SA z/OS Agent might not be able to send any responses to the automation adapter, and an AOF350E message is issued.

SA z/OS Automated Operator Functions

End-to-end automation uses dedicated automated operator functions (AOP).

The corresponding automation operator tasks are defined in the NetView DSIPARM dataset. The sample member `AOFOPFSO` already contains default definitions for NetView tasks. Define more according to your needs.

If your environment is controlled by RACF, then consult `ING.SINGSAMP(INGESAF)` member to add these operators to the RACF environment.

End-to-end automation adapter

End-to-end automation uses dedicated automated operator functions (AOP) in the primary agent to execute requests and to forward events to and from the end-to-end automation adapter.

These AOPs must be added to your SA z/OS policy.

Import the Automation Operators (AOP) entry `E2EADPT_AUTOOPS` (Automated Functions for E2E) from the `*E2E` best practices policy. Link it to the systems where you want to start the end-to-end automation adapter and build the automation configuration file.

Automated operator function

E2EOPER
E2EOPR01 ... E2EOPRnn

EVTOPER

Automation Operator

AUTE2E
AUTE2E01 ... AUTE2Enn

AUTEVT1 (AUTEVT2 as a backup)

Note that the maximum value of *nn* can be 99 but should ideally be the same as the number of concurrent working threads, as defined by parameter `eez-max-connections` in the master configuration file (see “The Automation Adapter Master Configuration File” on page 46).

The automated operator functions (E2EOPER and E2EOPR01 through E2EOPRnn) are used to execute requests from the automation adapter.

The automated operator function EVTOPER is used to forward events to the automation adapter and you must define it. Otherwise the initialization of the automation adapter fails.

End-to-end agent

Import the Automation Operators (AOP) entry E2EAGENT_AUTOOPS (Automated Functions for E2E agent) from the *E2E best practices policy. Link it to the systems where you want to start the end-to-end agent and build the automation configuration file.

Automated operator function

E2EAGT

Automation Operator

AUTAGT

SA z/OS Advanced Automation Options

The following advanced automation options can be used to change the default settings of the primary automation agents.

Automation adapter advanced automation options shows the common global variables that are provided as advanced automation options. Adapt them to your needs.

<i>Table 6. Automation adapter advanced automation options</i>			
Variable	Value	Effect	Further details
AOF_E2E_EAS_PPI	User-defined	PPI receiver ID of the event/automation service to be used to forward events to the end-to-end automation adapter.	
AOF_E2E_EVT_RETRY	1 to <i>n</i>	Specifies the number of retries, at intervals of one second, that are used to transfer events via PPI TECROUTE to the message adapter of the event/automation service. The events are to be forwarded to the end-to-end automation adapter.	“Non-disruptive recovery from errors” on page 86
AOF_E2E_EXREQ_NETLOG	1	The output to requests received from the end-to-end automation adapter and issued by the primary automation agent, is logged to the NetView log.	
	0	The output to those requests is not logged to the NetView log. 0 is the default setting.	

Table 6. Automation adapter advanced automation options (continued)

Variable	Value	Effect	Further details
AOF_E2E_TKOVN_TIMEOUT	hh:mm:ss	If a hot restart of the automation manager takes longer than the value specified in this variable, the end-to-end automation manager is informed about the outage and has to resynchronize with the first-level automation.	“Non-disruptive recovery from errors” on page 86
INGREQ_PRI.E2EMGR	Priority	Specifies the priority that incoming requests from the end-to-end automation manager are executed at. The application manager may overwrite this priority with the incoming request. Default: LOW	“Request issued to resource objects” on page 96

Chapter 9. Customizing USS and TCP/IP

This information describes how to customize USS and TCP/IP for end-to-end automation adapter and end-to-end automation agent.

End-to-End Automation Adapter

Prerequisites for USS

1. The user who is executing the automation adapter start script must have RACF® access rights to perform the TCP/IP command `netstat -D`, see [“Setting access privileges for TCP/IP connections” on page 42](#).
2. The user who is executing the automation adapter start script must also have RACF access rights to perform the TCP/IP command `netstat -a`, see [“Setting access privileges for TCP/IP connections” on page 42](#).
3. The user who is executing the automation adapter start script must have RACF access rights to perform the UNIX command `ps -A`, see [“Setting UNIX access privileges” on page 41](#).
4. The user ID must have write access to the data directory and to the log directory (if it does not reside in the data directory). It must also have read access to all other directories.
5. The automation adapter can be started by different user IDs but these IDs must all belong to the same group.
6. If facility class BPX.DEAMON is defined in RACF, the dynamic load libraries that are used by the automation adapter must be RACF program-controlled. You may check the program-controlled flag by command `ls -Elisa *.so`.

This can be done for modules of the zFS by setting the extended attribute `p`. During SMP/E installation, the extended attribute `p` is set automatically for all libraries (`*.so`) and `ingwto` within the automation adapter `lib` directory.

Although not recommended, if you copy the executables to another directory, set the program-controlled flag using the `extattr` shell command with the `+p` option: `extattr +p *.so`.

You may also have to define C++ and LE-provided runtime libraries, such as `SCEERUN`, `SCEERUN2`, `SCLBDLL`, and `CSSLIB`, to be RACF program-controlled.

7. You should redirect the standard output and error streams to the files `stdout.txt` and `stderr.txt` otherwise these files will not automatically be copied to the IBM Support package that is created with the `ingxadapter.sh` IBMSupport command. See also the sample job `ING.SINGSAMP (INGXADPT)`.

Customizing USS

Setting UNIX access 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.

For details see *z/OS Security Server RACF Security Administrator's Guide*.

See *z/OS Unix System Services Planning* for a list of the resource names available in the UNIXPRIV class, the z/OS UNIX privilege associated with each resource, and the level of access required to grant the privilege.

To define the profiles so that you can use the `ps -A` USS shell command:

1. If the UNIXPRIV class is not active, enter:

```
SETROPTS CLASSACT(UNIXPRIV) RACLIST(UNIXPRIV)
```

2. Define the RACF profile for the privileges:

```
RDEFINE UNIXPRIV SUPERUSER.PROCESS.GETPSENT UACC(NONE)
```

3. To grant READ access to the user OPER1:

```
PERMIT SUPERUSER.PROCESS.GETPSENT CLASS(UNIXPRIV) ID(OPER1) ACC(READ)
```

4. Perform an RACF refresh:

```
SETROPTS RACLIST(UNIXPRIV) REFRESH
```

Setting access privileges for TCP/IP connections

You can use the `netstat -a` command to display TCP/IP connections, and the `netstat -D nnnn` command to drop the connection `nnnn`.

However, from the USS shell, the `netstat` command is protected through the RACF SERVAUTH class.

You can define new security product resource names in the SERVAUTH class to control user's access to the UNIX shell `onetstat` command options. For example, for the `netstat -a` command you can use the profile `EZB.NETSTAT.mvsname.tcprocname.ALLCONN`

For a table of resource names see "Monitoring the TCP/IP network" in *z/OS Communications Server: IP System Administrator's Commands*.

For the DROP option, you can use the profile `MVS.VARY.TCPIP.DROP` in the following definitions:

```
RDEFINE OPERCMDS (MVS.VARY.TCPIP.DROP) UACC(NONE)
PERMIT MVS.VARY.TCPIP.DROP ACCESS(CONTROL) CLASS(OPERCMDS) ID(uid)
SETROPTS RACLIST(OPERCMDS) REFRESH
```

It may also be adequate to define a single generic profile that covers all commands, as follows:

```
RDEFINE OPERCMDS (MVS.VARY.TCPIP.***) UACC(NONE)
PERMIT MVS.VARY.TCPIP.** ACCESS(CONTROL) CLASS(OPERCMDS) ID(uid)
```

For more details, see the section "z/OS UNIX security considerations for TCP/IP" in the Redbook "*Security Configuration in a TCP/IP Sysplex Environment*."

Customize the Link List

Add the following libraries to your link list:

- SYS1.SCLBDLL2
- SYS1.SCEERUN
- SYS1.SCEERUN2
- SYS1.SCLBDLL
- *hlq*.SINGMOD1
- *hlq*.SINGMOD2

Alternatively, edit the `ingadapter.sh` start script and add these libraries as follows:

```
export STEPLIB=SYS1.SCLBDLL2:SYS1.SCEERUN:SYS1.SCEERUN2:SYS1.SCLBDLL:h1q.SINGMOD1:$STEPLIB
```

You can check that these libraries have been linked using the following command:

```
D PROG,LNKLST
```

End-to-End Automation Agent

Prerequisites for USS

1. The user ID must have write access to the data directory and to the log directory (if it does not reside in the data directory). It must also have read access to all other directories.
2. The end-to-end agent can be started by different user IDs, but these IDs must all belong to the same group.
3. If facility class BPX.DEAMON is defined in RACF, the dynamic load libraries that are used by the end-to-end agent must be RACF program-controlled. You may check the program-controlled flag by command `ls -Elisa *.so`.

This can be done for modules of the zFS by setting the extended attribute `p`. During SMP/E installation, the extended attribute `p` is set automatically for all libraries (`*.so`) and `ingwto` within the end-to-end agent lib directory.

Although not recommended, if you copy the executables to another directory, set the program-controlled flag using the `extattr` shell command with the `+p` option: `extattr +p *.so`.

You may also have to define C++ and LE-provided runtime libraries, such as `SCEERUN`, `SCEERUN2`, `SCLBDLL`, and `CSSLIB`, to be RACF program-controlled.

4. Customize the link list as described for the automation adapter, [“Customize the Link List”](#) on page 42.

Chapter 10. Customizing End-to-End Automation Adapter

This information describes how to configure the end-to-end automation adapter manually.

If you want to configure the end-to-end automation adapter as part of the SA configuration assistant, then refer to "Base SA z/OS Configuration Using the Configuration Assistant" in *IBM System Automation for z/OS Planning and Installation*. If you plan to configure the end-to-end automation adapter manually, then follow the instructions described afterward.

Configuration Files

The following configuration files are delivered with the automation adapter:

- “The Automation Adapter Environment Configuration File” on page 45 `ingadapter.properties`
- “The Automation Adapter Master Configuration File” on page 46, `ing.adapter.properties`
- “The Automation Adapter Plug-in Configuration file” on page 48, `ing.adapter.plugin.properties`
- “The SSL Configuration File” on page 49, `ing.adapter.ssl.properties`
- “The JAAS Configuration File” on page 51, `ing.adapter.jaas.properties`

Note: In the extracts and samples that are given in this document, EEZ (or `eez`) refers to end-to-end automation.

Both the `ing.adapter.properties` and `ing.adapter.plugin.properties` configuration files support system symbols.

The Automation Adapter Environment Configuration File

To start the automation adapter, you need to setup the environment variables in the configuration file `ingadapter.properties`. Specify the configuration file in the start procedure DD `STDENV`.

See also Chapter 3, “Prerequisites and Environment Setup,” on page 15

Parameter	Default Value	Description
<code>E2E_INSTALL_ROOT</code>	<code>/usr/lpp/ing/adapter</code>	The automation adapter installation path.
<code>E2E_CUSTOM_ROOT</code>	<code>/usr/lpp/ing/adapter</code>	The automation adapter customization path for modified configuration files, logger and SSL related files.
<code>E2E_JAVA_XMX</code>	256	The maximum memory that the automation adapter is allowed to allocate. You can specify the maximum memory allocated by JVM in MB.
<code>E2E_SSL_PASSW</code>	password	The password for the truststore that is generated by the <code>generateSampleKeys</code> function. This is only needed if you plan to generate sample keys for test purposes.
<code>E2E_SSL_VERSION</code> (introduced in OA58444)	TLS	The version of the SSL protocol that is used for SSL communication, for example, TLSv1.2. If you change it here, you must change it for the E2E agent (<code>INGAGT_SSL_VERSION</code>) and for the SMU server (<code>com.ibm.eez.adapter.ssl.version</code>) as well.

Table 7. Environment Variables of the Automation Adapter (continued)

Parameter	Default Value	Description
E2E_USRJP (introduced in OA58444)		Here you can add user specific Java system properties, for example, SSL debugging options. <pre>-Djavax.net.debug=ssl:handshake:verbose:keymanager:trustmanager</pre> This parameter is useful if you need to debug SSL handshake exceptions.
E2E_USRCP (introduced in OA58444)	/u/mydir/my.jar	Here you can add user specific classpath if needed.
E2E_DEBUG	ON	The switch to set on the logger function. Needed only if problem analysis is required.
E2E_JAVA_HOME	/local/java/J6.0	The Java runtime and SDK paths to invoke Java and the Java keytool respectively if the default java is not appropriate.
E2E_DROPCON	NO or YES	Set NO if you do not want the start script to drop possible open or unused connections.
E2E_TCPNAME (introduced in OA58444)	Empty string	The name of the TCP/IP stack where you want to isolate E2E Automation and NetView E/AS. System symbols are supported here.

The Automation Adapter Master Configuration File

The automation adapter needs a master configuration file `./config/ing.adapter.properties`. Without a properly verified master configuration file, the automation adapter does not start.

Normally, you may omit parameter **eez-remote-contact-hostname**. If you use a shared USS file system, see “Shared USS File System” on page 17.

Figure 8 on page 46 shows an example of an automation adapter master configuration file.

```
# --- Adapter Configuration -----
1 eez-remote-contact-hostname = E2Eadapter_host
  eez-remote-contact-port    = 2001
  eez-remote-contact-over-ssl = false
2 eez-operator-authentication = true
3 eez-functional-authentication = true
  eez-initial-contact         = false
  eez-max-connections         = 3

# --- EIF Configuration -----
  eif-cache                    = true
  eif-cache-size               = 1000
  eif-retry-interval-seconds   = 30
3 eif-send-to-hostname         = SMUserver_host
  eif-send-to-port             = 2002
  eif-receive-from-hostname    = 127.0.0.1
4 eif-receive-from-port       = 5529
5 eif-send-to-agent-hostname   = E2EAgent_host
5 eif-send-to-agent-port      = 2003

# --- Plugin Configuration -----
6 plugin-configfile-sa4zos = /custom_root/adapter/config/ing.adapter.plugin.properties
```

Figure 8. Sample automation adapter master configuration file, `ing.adapter.properties`

The parameters that are highlighted must be configured to correspond to your setup, as follows:

1 eez-remote-contact-hostname

The host name or IP address that is used by the automation adapter to receive synchronous requests from the SMU server. If you omit this parameter, the automation adapter automatically determines the hostname to which it is listening.

2 eez-operator-authentication

This parameter determines the type of operator authentication that is used. If it is set to false, the automation adapter uses the SA z/OS JAAS default login module, or if it is set to true, it uses the RACF-provided routine. (See also “The JAAS Configuration File” on page 51.)

3 eez-functional-authentication (introduced in OA58444)

This parameter determines the type of authentication that is used for the SMU functional user ID. If it is set to true, the automation adapter performs RACF authentication for requests that SMU sends with the functional user ID. If it is set to false, the automation adapter does not perform RACF authentication for this kind of request. SMU uses the functional user ID for internal queries.

4 eif-send-to-hostname

The host name or IP address of the SMU server.

5 eif-receive-from-port

The port that is to be used by the automation adapter to receive EIF events. It must meet the naming convention of the original EIF ServerPort parameter.

6 eif-send-to-agent-hostname, eif-send-to-agent-port

eif-send-to-agent-hostname is the host name of the E2E agent. eif-send-to-agent-port is the port of the E2E agent.

For high availability reasons, the use of a dynamic virtual IP address (DVIPA) is recommended.

7 plugin-configfile-sa4zos

This is the location of the plug-in configuration file. Specify the custom-root directory of your choice.

Table 9 on page 89 gives details of all the parameters that can be specified in the master configuration file.

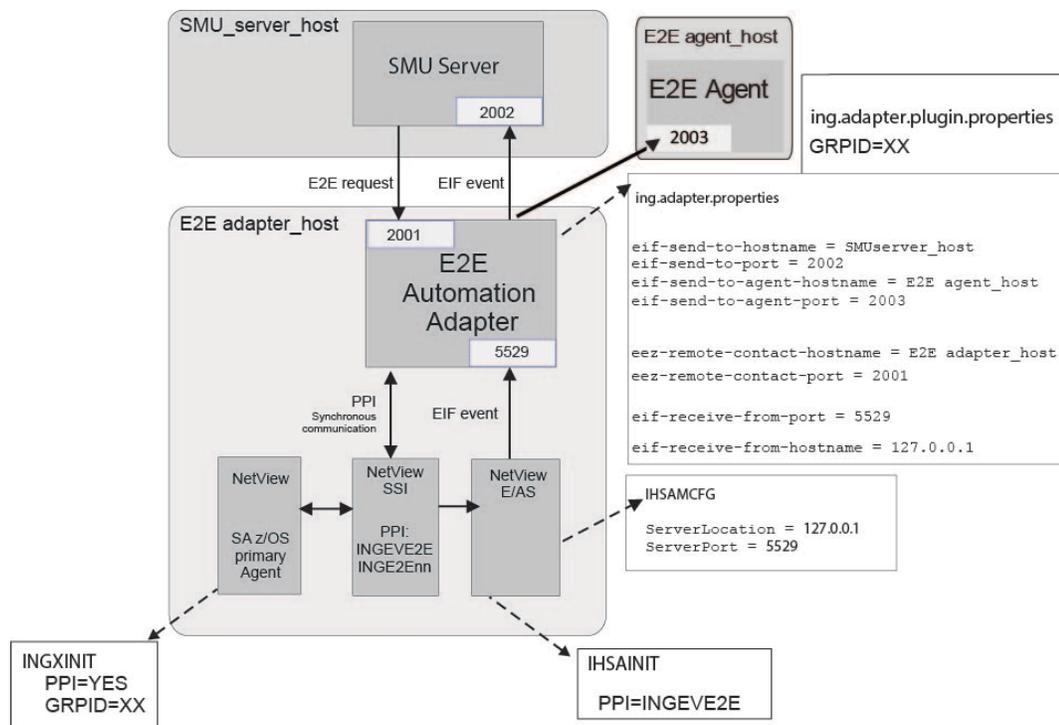


Figure 9. Communication Paths of the End-to-End Automation Components

The file IHSAMCFG is the NetView message adapter configuration member in data set SCNMUXCL, which is provided with NetView. Parameters `ServerLocation` and `ServerPort` in IHSAMCFG must match `eif-receive-from-hostname` and `eif-receive-from-port` in `ing.adapter.properties`.

Parameters `eif-send-to-port` in `ing.adapter.properties` must match the port and hostname of the SMU server.

Note: All property names and values are case sensitive. The Java Virtual Machine (JVM) of the automation adapter must be able to access the classes, paths, and so forth, that are specified in the configuration files.

The E2E adapter supports system symbols for `ing.adapter.properties`. For more details of how to specify the system symbols, see [“Support of MVS System Symbols in Configuration Files”](#) on page 18.

The Automation Adapter Plug-in Configuration file

In the adapter master configuration file, the property `plugin-configfile-sa4zos` points to the configuration file `ing.adapter.plugin.properties`. This plug-in properties file contains, for example, the customer specified domain name and the XCF group ID of the associated SA/NetView.

Figure 10 on page 48 shows an example of an automation adapter plug-in configuration file and highlights the entries that you may customize. GRPID must be customized to your needs. For other parameters, you can run with the default values.

```
# --- Specific settings for the SA/NetView communication ---
# --- Modify these parameters to your needs ---

1 GRPID    = XX
2 PPIBQL   = 1500
3 AUTOPFN  = E2EOPER
  TIMEOUT  = 600
4 TIMEOUT_SA_NOTAVAIL=180
  CODEPAGE = Cp1047

# --- Domain name may be modified or omitted ---

5 #plugin-domain-name = MyDomainName

6 AUDITMSG = true

# --- Do not modify these plugin settings ---
plugin-impl-class      = com.ibm.ing.sam.INGXPlugin0
plugin-impl-class-singleton = true
plugin-event-classes   = SystemAutomation_Resource_Status_Change SystemAutomation_...
plugin-auto-start      = true
```

Figure 10. Example automation adapter plug-in configuration file, `ing.adapter.plugin.properties`

1 GRPID

You must set the 2-byte XCF group ID of the SA z/OS subplex that you want to use in the plug-in configuration file. It must be the same as the group ID in `INGXINIT`.

2 PPIBQL

The number of elements in the PPI queue—this indicates how large the response to a request may be. It should be greater than the number of queue elements that you expect to be returned.

3 AUTOPFN

This a NetView operator task name or an SA z/OS automated function name to be used by SA z/OS to execute the incoming automation adapter request. See [“SA z/OS Automated Operator Functions”](#) on page 37.

4 TIMEOUT_SA_NOTAVAIL

You can set a timeout value in seconds with this parameter. A number greater than zero causes the plug-in to terminate if, after the specified time period, it detects that the SA z/OS NetView Agent is no longer available.

The automation adapter must receive two CHECK_HEALTH requests or other requests in order to calculate whether the timeout value has expired. If the SA z/OS NetView Agent cannot be contacted by both of the CHECK_HEALTH requests, the automation adapter issues message INGX9907I before it terminates itself.

The default value is 0 (zero).

5 plugin-domain-name

The plug-in domain name, which must be unique across *all* end-to-end automation domains (z/OS, Linux®, etc.). If you enter ? or omit the parameter, SA z/OS substitutes a name that is based on the XCF sysplex group name, however you can use your own naming strategy.

6 AUDITMSG=true

If parameter AUDITMSG was set to true, then requests sent by the SMU server and received by the automation adapter will be logged in the message file msgAdapter.log and msgFlatAdapter.log. If parameter AUDITMSG was set to false, these information messages are not logged. Default is true. Corresponding information messages INGX99nnI, nn=10,11,...,15, are issued independent of the success or the failure of the command. These messages will not be written to SYSLOG. The messages can be viewed on the end-to-end automation operator console. The following table maps the end-to-end automation requests to the information messages.

end-to-end automation Request Name	Information Message
RESET_FROM_NON_RECOVERABLE __ERROR	INGX9915I
EXCLUDE_NODE	INGX9911I
INCLUDE_NODE	INGX9911I
MOVE_AWAY	INGX9912I
MOVE_HOME	INGX9913I
MOVE_TO	INGX9914I
RESUME	INGX9910I
SUSPEND	INGX9910I
Any other command	INGX9910I

Table 10 on page 91 gives details about the parameters that you must include in your plug-in configuration file.

As same as the master configuration file `ing.adapter.properties`, the plug-in configuration file `ing.adapter.properties` also supports system symbols. For example:

```
GRPID=&E2EGRPID.
```

For more details of how to specify the system symbols, see [“Support of MVS System Symbols in Configuration Files”](#) on page 18.

The SSL Configuration File

The SSL configuration file is only required if `eez-remote-contact-over-ssl=true`, otherwise you can ignore this section.

You can set up the SSL configuration by one of the following techniques:

- Create certificates residing in RACF.
- Create keystore and truststore files residing in USS file system.

Customize the SSL configuration file using the RACF repository

For a detailed description of how to create certificates and a corresponding keyring in the RACF repository, see [Chapter 13, “Setting Up E2E Automation RACF Keyrings \(OA58444\),”](#) on page 61.

Customize the SSL configuration file residing in the USS file system

The automation adapter start-stop script, `ingadapter.sh`, reads the SSL configuration file, `ing.adapter.ssl.properties`, in order to set up the SSL parameters.

It is expected that the truststore file can be read without a password. The keystore and the certificates in the keystore must have the same password.

Figure 11 on page 50 shows an example of an SSL configuration file. It must contain at least the properties that are shown in Figure 11 on page 50. Table 11 on page 92 gives details about these parameters. You may specify an encrypted password instead of the clear text. See also extra section about usage of the encryption algorithm.

```
#
# --- Adapter SSL Configuration File -----
#
1 eez-ssl-truststore           = ./ssl/ing.ssl.truststore.jks
1 eez-ssl-keystore           = ./ssl/ing.ssl.adapter.jks
  eez-ssl-keystorepassword   = passphrase
  eez-ssl-keystorealias     = eeadapter
```

Figure 11. Example SSL configuration file, `ing.adapter.ssl.properties`

1

Note: The SSL directory is initially empty. The truststore and keystore files are *not* delivered with the automation adapter and must be created by the customer. Specify a full path in case the SSL is not a subdirectory of the adapter directory.

How to Encrypt the Keystore Password

In order to generate an encrypted password for parameter `eez-ssl-keystore-password`, you may use the SMU utility `cfgsmu`. This tool generates among other items the SSL configuration file for distributed platforms. Copy the line with the generated password from `eez fla.ssl.properties` and paste it into the SSL configuration file `ing.adapter.ssl.properties` on z/OS.

Notes:

1. The end-to-end automation must use the same version of the encryption algorithm as used by `cfgsmu`, which is normally AES.
2. In regard to the EBCDIC character conversion, the codepage IBM 1047 is used for all configuration files. Make sure that all characters are correctly converted. It may be necessary to convert characters manually using the ISPF editor, for example, {AES} must be 'C0C1C5E2D0' X. For example, if for parameter `eez-ssl-keystore-password` special characters are not correctly converted according to code page IBM 1047 then the password may not be recognized correctly during decryption.

Optionally, use the USS script `/custom-path/adapter/ingencrypt.sh` to encrypt the keystore password.

For details, refer to [“Encrypt SSL Password”](#) on page 54.

Example Invocation:

```
cd /usr/lpp/ing/adapter
./ingencrypt.sh -encrypt my password
```

Copy the line with the encrypted password from `stdout` and paste it into the SSL configuration file `/custom-path/adapter/config/ing.adapter.ssl.properties`, parameter **eez-ssl-keystore-password**.

How to Create Keystore and Truststore for Use by SSL

Below you find an example of how to generate end-to-end automation adapter keystore and the truststore via the Java keytool which resides in the Java SDK of your z/OS Java installation.

You will specify both files, keystore and truststore, in the end-to-end automation SSL configuration file `ing.adapter.ssl.properties`.

```
# SMU SSL config file directory
EEZ_CONFIG_DIR=/custom-root/ssl
# Keys will expire in 25 years
KEY_VALIDITY_DAYS=9125
# Passphrase at least 6 characters
PASSPHRASE=passphrase
```

1. Generate keystore for end-to-end automation z/OS

```
keytool -genkey -keyalg RSA -validity ${KEY_VALIDITY_DAYS} \
  -alias eezadapter -keypass ${PASSPHRASE} -storepass ${PASSPHRASE} \
  -dname "cn=E2E Adapter, ou=System Automation, o=IBM, c=US" \
  -keystore "${EEZ_CONFIG_DIR}/ing.ssl.adapter.jks"
```

2. Export Certificate file with public key for end-to-end automation:

```
${JAVA_KEYTOOL} -export -alias eezadapter \
  -file "${EEZ_CONFIG_DIR}/ingadapter.cer" -storepass ${PASSPHRASE} \
  -keystore "${EEZ_CONFIG_DIR}/ing.ssl.adapter.jks"
```

3. Create the truststore that contains the public keys for the end-to-end automation and the SMU

It is assumed that the certificate of the SMU server has the name *smuserver.cer* and the corresponding alias name is *smuserver*. Refer to the description of SMU configuration tool in "Security Tab" of *Service Management Unite Automation Installation and Configuration Guide*.

Import the end-to-end automation certificate to the truststore.

```
${JAVA_KEYTOOL} -import -noprompt -alias eezadapter \
  -file "${EEZ_CONFIG_DIR}/ingadapter.cer" -storepass ${PASSPHRASE} \
  -keystore "${EEZ_CONFIG_DIR}/ing.ssl.truststore.jks"
```

Import the SMU certificate to the truststore.

```
${JAVA_KEYTOOL} -import -noprompt -alias smuserver \
  -file "${EEZ_CONFIG_DIR}/smuserver.cer" -storepass ${PASSPHRASE} \
  -keystore "${EEZ_CONFIG_DIR}/ing.ssl.truststore.jks"
```

4. Copy the keystore from step 1 and the truststore from step 3 into the end-to-end automation SSL directory. Update `ing.adapter.ssl.properties` accordingly.

For more information, refer to [Generating Keystore and Truststore with SSL public and private keys in IBM Tivoli System Automation Application Manager Installation and Configuration Guide](#).

The JAAS Configuration File

The JAAS definitions for the automation adapter are defined in the JAAS configuration file, `/install-root/config/ing.adapter.jaas.properties`. The JAAS configuration file defines which login module to use for user authentication. User ID and password will be validated by the specified login module.

If you use RACF as user authentication, leave the JAAS configuration file as it is and you can skip this section.

If needed, the z/OS administrator may overwrite the JAAS login module with any third party security login module other than RACF. In this case, modify the entry `EEZAdapterLogin` in file `ing.adapter.jaas.properties`.

The automation adapter master configuration file, `ing.adapter.jaas.properties`, contains the parameter **eez-operator-authentication=true** to switch between the entries. Setting the parameter to false causes the automation adapter to use the JAAS entry `EEZAdapterDefaultLogin`. Otherwise, the entry `EEZAdapterLogin` is used.

Chapter 11. Operating the End-to-End Automation Adapter

The automation adapter is started as a batch job or a started task which runs the USS script `ingadapter.sh` via `BPXBATCH`.

How to Use the Automation Adapter

You can operate the automation adapter in the following ways:

- As a batch job or started task, see “[JCL Procedure Using BPXBATCH](#)” on page 55
- As a defined resource in SA z/OS, see “[SA z/OS Defined Resource](#)” on page 55

USS Script

Start/Stop end-to-end automation adapter

Syntax

The following USS shell script will be invoked by the JCL procedure `INGXADPT`:

```
ingadapter.sh function suffix option
```

This script has the following parameters:

function

This can be one of the following:

start|START

This starts the automation adapter. The default is start.

stop|STOP (deprecated)

This stops the automation adapter.

This stop option is deprecated and remains for compatibility purpose only. Please use `MVS stop` command instead: **P jobname,A=asid**.

generateSampleKeys

This creates and stores sample keystore and truststore files in the automation adapter SSL directory. Both `java` and `keytool` must be found in the `UNIX PATH` environment variable if you want to use this function.

Note: Use the sample keys for *test purposes only*.

IBMSupport

This produces a zip file in the data directory of all the necessary log and trace files that IBM Support might need.

suffix

The suffix is optional. It might be used for a shared zFS. The meaning of *suffix* depends on a hyphen "-" as 1st character.

1. Suffix that starts WITHOUT a hyphen "-".

The suffix now supports shared zFS environments in a better way.

2. Suffix that starts WITH a hyphen "-".

In order to support the meaning of the suffix of older releases add a hyphen "-" in front of the suffix.

Suffix starts WITHOUT a hyphen "-".

We refer to output such as end-to-end automation adapter log files, PID file, stdout, stdin or the adapter release info file. If you want output to be written into a subdirectory depending on the system name or depending on any other system differentiator then use a suffix without a hyphen. This might be useful for shared zFS. It is treated as the user specified name of the subdirectory inserted into the data directory path used for output data. Where suffix can be any differentiator for the current system.

For example,

```
ingadapter START SYSA
```

causes the end-to-end automation adapter to write info into the following output directory

```
/custom-root/adapter/data/SYSA/eez/log
```

if we assume `E2E_CUSTOM_ROOT = /custom-root/adapter/` was set within configuration file `ingadapter.properties`.

Note, *suffix* is defined in the JCL parameter and the JCL defines the location of `stderr` and `stdout`. See example `ING.SINGSAMP(INGXADPT)`.

Suffix starts WITH a hyphen "-".

In this case the suffix (whereby hyphen is removed) is used as middle qualifier for the master configuration file, for example,

```
ing.adapter.suffix.properties
```

Normally this is no longer needed and should not be used any more. This function is only supported for compatibility reasons. Example of the corresponding master configuration files name with suffix SYSA `ing.adapter.SYSA.properties`.

option

The parameter option may be used for debugging purposes in order to trace the shell script or debug the end-to-end automation adapter.

-T, -t

switches on the shell script trace.

-D, -d

switches on the end-to-end automation adapter logger.

Encrypt SSL Password

The USS script `ingencrypt.sh` enables you to encrypt and decrypt any string such as the SSL password.

In order to invoke the command, ensure that the current working directory is the automation adapter install-root directory. The java runtime must be installed.

Output is written to `stdout`.

Example invocation:

```
cd /usr/lpp/ing/adapter
./ingencrypt.sh -encrypt abcdef
```

Command syntax

```
ingencrypt.sh function string
```

function

The function to be executed. This can be one of :

-encrypt|-ENCRYPT

-decrypt|-DECRYPT

string

The string to be encrypted or decrypted.

JCL Procedure Using BPXBATCH

There is a sample procedure, INGADPT, that is in the SA z/OS samples data set that allows you to start the automation adapter as a batch job or started task.

You must customize the paths and job statements according to your needs.

For example, to start the automation adapter, you may use the **MVS START** command

```
S INGADPT,F=START
```

Or if you want to run the automation adapter with a debug option

```
S INGADPT,F=START,OPT=-D
```

To stop the automation adapter, use the same procedure with a stop parameter.

For example, use the **MVS START** command

```
S INGADPT,F=STOP
```

It is recommended to define the automation adapter as an application in the automation policy to operate it with SA z/OS. The goal of these definitions is:

- Ease of operations support
- Availability support reducing downtimes

See [Appendix B, “High Availability of the End-to-End Automation Adapter,”](#) on page 81 for further details.

SA z/OS Defined Resource

Define the automation adapter as an application in the automation policy to operate it with SA z/OS. You can use the sample of the automation adapter E2ADPT in the available *E2E policies.

The goal of these definitions is:

- Ease of operations support
- Availability support reducing downtimes

See [Appendix B, “High Availability of the End-to-End Automation Adapter,”](#) on page 81 for further details.

Command INGE2E

The INGE2E command provides the operational API for the end-to-end automation adapter and the end-to-end agent. It can be issued on the NetView console. For details, see [“INGE2E”](#) on page 100.

Chapter 12. Security Consideration for the End-to-End Automation Adapter

This information gives details of the security (authentication and authorization) that is provided for the automation adapter.

Authentication

When the SMU server or end-to-end agent issues a request to the automation adapter, there is always a user ID and password associated with it.

You must define the user ID and password with RACF and the user ID must have an OMVS segment. These define who is allowed to send an end-to-end automation request.

Note: The E2E automation adapter does not check the corresponding NetView application ID for class APPL, as it would be done by a normal NetView logon. For execution of an E2E automation adapter request, a successful checking of the user ID and password is sufficient.

The automation adapter performs authentication checking using RACF via the JAAS architecture. You have three alternatives:

- Set the **eez-operator-authentication** parameter to true in the master configuration file, `ing.adapter.properties`.
- Set the **eez-operator-authentication** parameter to false in the master configuration file, `ing.adapter.properties`.
- Define the JAAS logon module that is provided by any third party product.

Authority checking by the automation adapter

Authorization checking is used to determine whether the automation adapter grants or rejects the execution of an operation for the user ID that is associated with the request.

The automation adapter supports two alternative techniques to protect from unauthorized operations:

1. The use of the automation adapter authorization exit as described in previous releases
2. The exploitation of the System Automation built-in third party security.

The second technique is used in any case. The first technique is still supported for compatibility reasons.

Authorization checking by user exit

The automation adapter performs the authorization user exit for each command that is to be executed, except for query requests.

The authorization user exit is an external REXX program that you must provide. The name of the authorization user exit must be AOFEXE2E. A sample of this program can be found in SINGSAMP.

If no authorization user exit exists, a user is considered to be authorized for each request.

If the authorization user exit is provided, it decides whether the request is accepted or rejected with message ING238E.

The following arguments are passed to the authorization user exit via the NetView PIPE default safe:

- User ID
- Request name
- SA z/OS command, resulting from the request

where the end-to-end automation request names are mapped to SA z/OS commands as shown in the following table:

End-to-end automation request name	Command abbreviation for RACF profile
Online	INGREQ
Offline	INGREQ
CancelRequest	INGSET
IncludeNode	INGSET
ExcludeNode	INGREQ
ResetFromNRE	SETSTATE
SetTarget The SetTarget request sets or resets the availability target of an APG resource.	INGGROUP
MoveTo, MoveAway, MoveHome	INGMOVE
NativeCommand	Any user command provided by the WebUI Command Widget

For further details about the command parameters, see [“Requests issued by the SMU and the end-to-end automation manager” on page 97.](#)

Note: The authorization user exit must return a zero return code if the user is allowed to execute the command. Otherwise a non zero return code has to be returned.

A non-zero return code results in the message:

```
ING238E TASK EXECUTION REQUEST EXECUTE_UNSOLICITED_REQUEST FAILED:
        ERROR=5SERVICE='AOFEXE2E' RC=rc
        REASON='Authorization failed for user user id
        for command command'
```

Sample authorization user exit

If you do not want to provide an authorization user exit of your own, you can use the sample that is provided by SA z/OS as member AOFEXE2E in the sample data set SINGSAMP.

Copy the sample into a clist concatenation -- it will then be active and will expect the appropriate RACF profiles. If you have not defined these profiles no requests will be authorized.

The sample exit includes a RACF checking routine that uses the RACF class FACILITY and the following profile pattern to check whether the user has the correct RACF access rights:

```
hlq.command.resourceName.resourceType.resourceSystem
```

where *hlq* is a unique prefix, for example, INGE2E, and *command* is one of the SA z/OS commands given in the table above (that is, INGREQ, INGSET, or SETSTATE).

Appropriate RACF profiles

If you use the SA z/OS sample authorization exit, you must define RACF profiles for all users that should pass the authorization check.

Using the SA z/OS sample authorization exit there is no means to disable authorization checking of an end-to-end automation request. You must define at least the profile INGE2E* for the user that is associated with the request.

By adding an RACF user ID to the access list of such an RACF profile and giving the UPDATE authority, the user is allowed to execute the command for the resource *resourceName.resourceType.resourceSystem*.

Examples of profiles for command *INGREQ*

INGE2E.INGREQ.TSO.APL.KEY1

The user is allowed to start or stop application TSO on system KEY1.

INGE2E.INGREQ.TSO.**

The user is allowed to start or stop TSO resources of type application or group on all systems.

INGE2E.INGREQ.**

The user is allowed to start or stop all resources on all systems.

Authorization checking by System Automation security

The automation adapter performs the old authorization user exit plus additional security checking:

1. Basic third party security check for command INGRXTX0
2. Extended third party security check for architected end-to-end automation adapter requests
3. If the old authorization exit AOFEXE2E exists, it is processed
4. System Automation resource security check for architected end-to-end automation adapter requests.

Basic third party security checking

The purpose of the basic authorization is to check whether an already authenticated end-to-end automation user has the permission to access the end-to-end automation backend INGRXTX0.

The basic authorization is sufficient to perform any end-to-end automation adapter function except architected end-to-end automation adapter requests for which extended security is required.

A user ID that sends a request to the end-to-end automation adapter must fulfill the following third party SAF check:

```
class      NETCMDS
permission READ
profile:   netid.netvdom.INGRXTX0
```

Extended third party security checking

The purpose of extended authorization is to check whether an already authenticated end-to-end automation user has the permission to perform architected end-to-end automation adapter requests.

For a mapping between end-to-end automation adapter requests and System Automation commands, see [Appendix D, “SA z/OS specific implementation of the end-to-end automation resource data model,” on page 93.](#)

A user ID that sends an architected end-to-end automation adapter requests to the end-to-end automation adapter must fulfill the following third party check:

```
class      NETCMDS
permission READ
profile:   netid.netvdom.xxxxxxxx
```

where xxxxxxxx is one of the following System Automation commands:

- INGRYRU0 for synonym INGREQ
- INGRYSE0 for synonym INGSET
- INGRYMX for synonym INGMOVE
- AOFRASTA for synonym SETSTATE
- INGRYGRA for synonym INGGROUP

System Automation resource security checking

The purpose of resource security is to check whether an already authenticated end-to-end automation user has the permission to perform an action on the related System Automation resource.

For details about System Automation resource security, see *IBM System Automation for z/OS Planning and Installation*.

A user ID that sends an architected end-to-end automation adapter request to the End-to-end automation adapter must fulfill the following third party SAF check:

class		SYSAUTO
permission		UPDATE or CONTROL
profile:		AGT.sysplex.xcgrp.RES.name.type.system
INGREQ	permission	UPDATE
INGSET	permission	CONTROL
SETSTATE	permission	UPDATE
INGMOVE	permission	UPDATE
INGGROUP	permission	UPDATE

Native command security checking

A native command provided by the WebUI command Widget is executed under the NetView operator task and logon user ID of the WebUI.

For example if BOB did logon to the WebUI and issued an arbitrary NetView command, then the command is executed within the security context of NetView operator BOB. The standard NetView command authorization applies.

Chapter 13. Setting Up E2E Automation RACF Keyrings (OA58444)

This chapter describes how to set up RACF keyrings for SSL communication between the E2E adapter and E2E agent and SMU.

A RACF keyring is created for the E2E adapter and another one for the E2E agent. A self-signed certificate is created for each keyring and the public keys are exchanged to each keyring.

The public key from SMU keystore is exported and imported to the E2E adapter keyring. The public key from E2E adapter server is exported and imported to the SMU truststore.

It is assumed that the E2E adapter and E2E agent started task runs as user E2EUSER. It is also assumed that the ID of the keyring is E2EUSER. If not, see [“Additional RACF Profile Access for E2EUSER”](#) on page 69.

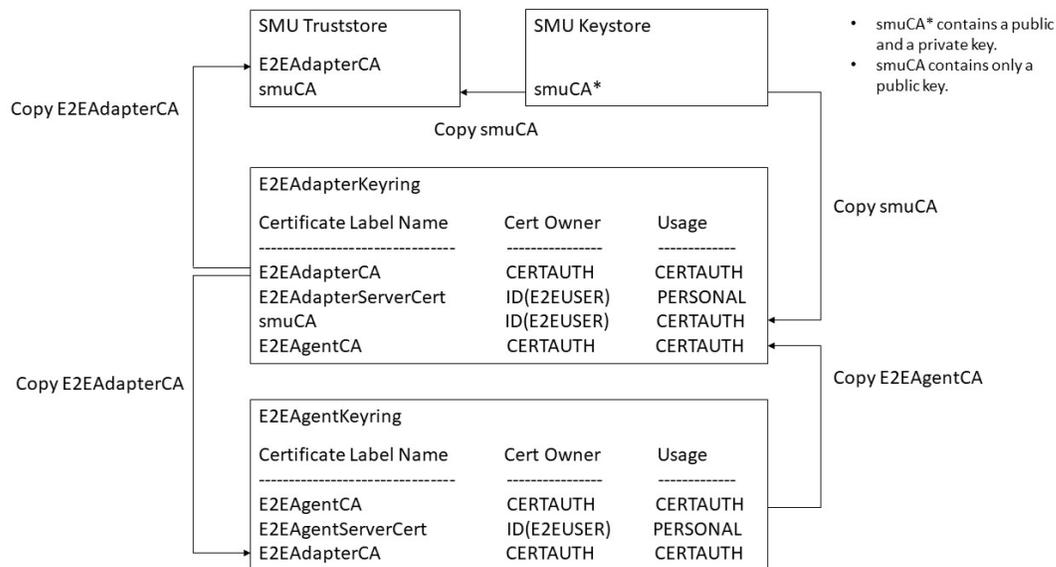


Figure 12. E2E Automation RACF Keyring settings

E2E Adapter RACF Keyring

This section describes the RACF definitions that are needed to set up the keyring for the E2E adapter, how to export E2E adapter's public key, and how to import SMU public key to E2E adapter keyring.

Define E2EAdapterCA

Procedure

1. Create a RACF keyring for the E2E adapter to use it as keystore.

```
RACDCERT ID(E2EUSER) ADDRING(E2EAdapterKeyring)
```

2. Create a self-signed RSA key pair to act as a Certificate Authority (CA) certificate. A key pair consists of a public and a private key. "E2EAdapterCA" is the label or alias of the certificate.

```
RACDCERT GENCERT CERTAUTH SUBJECTSDN(CN('CA for E2EAdapter')
O('IBM') OU('E2EAdapter') C('US')) SIZE(2048) WITHLABEL('E2EAdapterCA')
NOTAFTER(DATE(2029-12-31))
```

3. Connect the CA certificate to the keyring.

```
RACDCERT ID(E2EUSER) CONNECT(RING(E2EAdapterKeyring)
LABEL('E2EAdapterCA') CERTAUTH)
```

4. Create a personal certificate, signed by the CA certificate "E2EAdapterCA". Create an RSA key pair for the E2E adapter server signed by the CA certificate. The common name (CN) value is typically the host name of the z/OS LPAR, which hosts the E2E adapter. "E2EAdapterServerCert" is the alias of the personal certificate of the E2E adapter.

```
RACDCERT ID(E2EUSER) GENCERT SUBJECTSDN(CN('myServer.host.com')
O('IBM') OU('E2EAdapter') C('US')) SIZE(2048) SIGNWITH(CERTAUTH
LABEL('E2EAdapterCA')) WITHLABEL('E2EAdapterServerCert')
NOTAFTER(DATE(2029-12-31))
```

5. Connect the personal certificate to the keyring.

```
RACDCERT ID(E2EUSER) CONNECT(RING(E2EAdapterKeyring)
LABEL('E2EAdapterServerCert'))
```

6. Verify that the keyring and certificates were created correctly.

```
RACDCERT ID(E2EUSER) LISTRING(E2EAdapterKeyring)
RACDCERT CERTAUTH LIST(LABEL('E2EAdapterCA'))
RACDCERT ID(E2EUSER) LIST(LABEL('E2EAdapterServerCert'))
```

The following screen capture shows the LISTRING response:

```
Ring:
  >E2EAdapterKeyring<
Certificate Label Name      Cert Owner      Usage      Default
-----
E2EAdapterCA              CERTAUTH       CERTAUTH   NO
E2EAdapterServerCert      ID(E2EUSER)    PERSONAL   NO
```

7. Activate the RACF definition.

```
SETROPTS RACLIST(DIGTCERT) REFRESH
```

Export E2EAdapterCA

Export the E2E adapter server's CA certificate for use by the E2E agent and SMU server.

Procedure

Export the E2E adapter server's CA certificate, containing the public key, to a z/OS sequential file. Format CERTDER specifies a DER-encoded X.509 certificate.

```
RACDCERT CERTAUTH EXPORT(LABEL('E2EAdapterCA'))
DSN('HLQ.CERTS.E2EADAPT') FORMAT(CERTDER)
```

It is expected by RACDCERT that the data set is variable-byte (VB). Here is a sample data set definition.

```
Organization . . . : PS
Record format . . . : VB
Record length . . . : 84
Block size . . . . : 27998
```

What to do next

Now you can transfer the E2E adapter's public key to the SMU server. Import it into the truststore of the SMU server. See [“Import E2EAdapter Public Server CA into SMU”](#) on page 64.

You can also connect the E2E adapter's public key to E2E agent keyring. See [“Connect E2EAdapterCA public key to E2E Agent Keyring”](#) on page 67.

Import SMUCA

Import the public key from SMU (E2E adapter client) into the E2E adapter keyring.

Before you begin

Assume that you have created and exported the public key of the client CA from SMU for use by the E2E adapter. If you have not done these prerequisite steps yet, see [“Create and Export SMU Public Client CA into E2E Adapter”](#) on page 63.

Procedure

1. Import the client's CA certificate (public key) into RACF as a trusted certificate.

```
RACDCERT ID(E2EUSER) ADD('HLQ.CERTS.SMUCA') WITHLABEL(smUCA') TRUST
```

2. Connect the client's CA certificate (public key) as a trusted (CERTAUTH) certificate to the RACF keyring that is used by the E2E adapter server as its truststore.

```
RACDCERT ID(E2EUSER) CONNECT(RING(E2EAdapterKeyring)  
LABEL('smUCA') USAGE(CERTAUTH))
```

3. Verify that the E2E adapter's keyring contains the correct certificates.

```
RACDCERT ID(E2EUSER) LISTRING(E2EAdapterKeyring)  
RACDCERT ID(E2EUSER) CERTAUTH LIST(LABEL('smUCA'))
```

The following screen capture shows the LISTRING response:

```
Ring:  
>E2EAdapterKeyring<  
Certificate Label Name          Cert Owner          USAGE          DEFAULT  
-----  
E2EAdapterCA                   CERTAUTH           CERTAUTH      NO  
E2EAdapterServerCert          ID(E2EUSER)       PERSONAL      NO  
smUCA                           ID(E2EUSER)       CERTAUTH      NO
```

4. Activate the RACF definition.

```
SETROPTS RACLIST(DIGTCERT) REFRESH
```

Keytool – SMU

This section describes how to create a self-signed certificate for SMU and export it for use by the E2E adapter, and how to import the E2E adapter's public key into SMU truststore file.

Create and Export SMU Public Client CA into E2E Adapter

Create a self-signed certificate for SMU and export it for use by the E2E adapter.

Procedure

1. Create a self-signed RSA key pair for the client and add them to a JKS file, which acts as the client's keystore.

```
keytool -genkeypair
  -alias smuca
  -keystore smu.keystore.jks
  -dname "CN=SMU.host.com, OU=E2EAdapterClient, O=IBM, C=US"
  -keyalg RSA
  -keysize 2048
  -keypass passphrase
  -storepass passphrase
  -validity 365
```

2. Export the SMU CA certificate containing the public key to a certificate file.

```
keytool -exportcert
  -alias smuca
  -file smuCA.cer
  -keystore smu.keystore.jks
  -storepass passphrase
```

3. Transfer the exported SMU CA certificate "smuCA.cer" in binary format to the z/OS LPAR where the E2E adapter is hosted. Use FTP to put the CA certificate into the USS path `u/e2e/certs/smuCA.cer`. It is expected by RACDCERT that the dataset is variable-byte (VB).
4. Use the following command to copy the SMU CA certificate into a z/OS sequential file.

```
cp /u/e2e/cert/smuCA.cer "'HLQ.CERTS.SMUCA'"
```

What to do next

Now you can import the SMU CA certificate into E2E adapter's keyring. See [“Import SMUCA”](#) on page 63.

Import E2EAdapter Public Server CA into SMU

The SMU server and the E2E agent need to import the E2E adapter server CA (public key only) into its truststore jks file.

Procedure

1. Copy the z/OS sequential file `HLQ.CERTS.E2EADAPT` in binary format to a USS file. Use the following command to save the binary file:

```
cp "'HLQ.CERTS.E2EADAPT'" ./E2EAdapterCA.cer
```

2. Transfer the binary file to the system where SMU resides. Import the certificate into SMU's truststore `smu.truststore.jks`.

```
keytool -importcert
  -noprompt
  -alias E2EAdapterCA
  -file E2EAdapterCA.cer
  -keystore smu.truststore.jks
  -storepass passphrase
```

3. Import the public key from the SMU certificate `smuCA.cer` into SMU's truststore.

```
keytool -importcert
  -noprompt
  -alias smuca
  -file smuCA.cer
  -keystore smu.truststore.jks
  -storepass passphrase
```

4. Verify the contents of SMU truststore by using the following command. Check if there are two entries of `trustedCertEntry` type for the `smuca` and the `E2EAdapterCA`.

```
keytool -list -keystore SMU.truststore.jks -storepass passphrase -v
```

```

Keystore type: JKS
Keystore provider: SUN

Your keystore contains two entries.

Alias name: e2eadapterca
Creation date: Aug 13, 2019
Entry type: trustedCertEntry

Owner: CN=CA for E2EAdapter, OU=E2EAdapter, O=IBM, C=US
Issuer: CN=CA for E2EAdapter, OU=E2EAdapter, O=IBM, C=US
Serial number: 0

Valid from: Mon Aug 12 00:00:00 CEST 2019 until: Mon Dec 31 22:59:59 CET 2029
Certificate fingerprints:
  MD5:          31:F0:C0:9E:96:8E:92:FA:F3:BC:AE:22:3F:35:89:4F
  SHA1:         FC:C2:56:0E:B0:8C:D9:6F:EB:33:B9:6C:07:FA:8E:83:1A:73:93:1A
  Signature algorithm name: SHA1withRSA
  Version: 3

Extensions:

#1: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [
  Key_CertSign
  Crl_Sign
]

#2: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
  CA:true
  PathLen:2147483647
]

#3: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
  KeyIdentifier [
0000: 86 AF FA 63 AE E6 8F D0 97 3C B8 F7 BC 11 54 E0 ...c.....<....T.
0010: B2 8B FC 56          ...V
  ]
]

#4: ObjectId: 2.16.840.1.113730.1.13 Criticality=false

*****
*****
Alias name: smuca
Creation date: Aug 8, 2019
Entry type: trustedCertEntry

Owner: CN=SMU.host.com, OU=E2EAdapterClient, O=IBM, C=US
Issuer: CN=SMU.host.com, OU=E2EAdapterClient, O=IBM, C=US
Serial number: 5d4c1535
Valid from: Thu Aug 08 14:27:33 CEST 2019 until: Tue May 03 14:27:33 CEST 2022
Certificate fingerprints:
  MD5:          B7:BE:A4:75:6F:68:01:D3:D1:B9:78:88:A4:01:0C:1E
  SHA1:         C1:A1:7F:72:9F:0A:16:09:BC:B0:F6:C6:3E:0B:26:06:29:10:05:9C
  Signature algorithm name: SHA1withRSA
  Version: 3

*****
*****

```

E2E Agent RACF Keyring

This section describes the RACF definitions that are needed to set up the keyring for the E2E agent and how to connect the public part of the certificates from the E2E agent and E2E adapter to each keyring.

Define E2EAgentCA

Procedure

1. Create a RACF keyring for the E2E agent to use it as keystore.

```
RACDCERT ID(E2EUSER) ADDRING(E2EAgentKeyring)
```

2. Create a self-signed RSA key pair containing a public and a private key to act as a CA. "E2EAgentCA" is the label or alias of the certificate.

```
RACDCERT GENCERT CERTAUTH SUBJECTSDN(CN('CA for E2EAgent')  
O('IBM') OU('E2EAgent') C('US')) SIZE(2048) WITHLABEL('E2EAgentCA')  
NOTAFTER(DATE(2029-12-31))
```

3. Connect the CA certificate to the keyring.

```
RACDCERT ID(E2EUSER) CONNECT(RING(E2EAgentKeyring)  
LABEL('E2EAgentCA') CERTAUTH)
```

4. Create a personal certificate, signed by the CA certificate "E2EAgentCA". Create an RSA key pair for the E2E agent server signed by the CA certificate. The common name (CN) value is typically the host name of the z/OS LPAR, which hosts the E2E agent. "E2EAdapterServerCert" is the alias for the personal certificate of the E2E agent.

```
RACDCERT ID(E2EUSER) GENCERT SUBJECTSDN(CN('myServer.host.com')  
O('IBM') OU('E2EAgent') C('US')) SIZE(2048) SIGNWITH(CERTAUTH  
LABEL('E2EAgentCA')) WITHLABEL('E2EAgentServerCert')  
NOTAFTER(DATE(2029-12-31))
```

5. Connect the personal certificate to the keyring.

```
RACDCERT ID(E2EUSER) CONNECT(RING(E2EAgentKeyring)  
LABEL('E2EAgentServerCert'))
```

6. Verify that the keyring and certificates were created correctly.

```
RACDCERT ID(E2EUSER) LISTRING(E2EAgentKeyring)  
RACDCERT CERTAUTH LIST(LABEL('E2EAgentCA'))  
RACDCERT ID(E2EUSER) LIST(LABEL('E2EAgentServerCert'))
```

The following screen capture shows the LISTRING response:

```
Ring:  
>E2EAgentKeyring<  
Certificate Label Name          Cert Owner      USAGE          DEFAULT  
-----  
E2EAgentCA                     CERTAUTH       CERTAUTH       NO  
E2EAgentServerCert             ID(E2EUSER)    PERSONAL       NO
```

7. Activate the RACF definition.

```
SETROPTS RACLIST(DIGTCERT) REFRESH
```

Connect E2EAgentCA public key to E2E Adapter Keyring

Procedure

1. Connect the E2E agent CA certificate as a trusted (CERTAUTH) certificate to the RACF keyring that is used by the E2E adapter server as its truststore.

```
RACDCERT ID(E2EUSER) CONNECT(CERTAUTH RING(E2EAdapterKeyring)  
LABEL('E2EAgentCA') USAGE(CERTAUTH))
```

2. Verify that the E2E adapter's keyring contains the correct certificates.

```
RACDCERT ID(E2EUSER) LISTRING(E2EAdapterKeyring)
```

The following screen capture shows the LISTRING response:

```

Ring:
>E2EAdapterKeyring<
Certificate Label Name      Cert Owner      USAGE      DEFAULT
-----
E2EAdapterCA              CERTAUTH        CERTAUTH    NO
E2EAdapterServerCert      ID(E2EUSER)     PERSONAL    NO
smuCA                     ID(E2EUSER)     CERTAUTH    NO
E2EAgentCA                CERTAUTH        CERTAUTH    NO

```

Connect E2EAdapterCA public key to E2E Agent Keyring

Procedure

1. Connect the E2E adapter CA certificate as a trusted (CERTAUTH) certificate to the RACF keyring that is used by the E2E agent as its truststore.

```
RACDCERT ID(E2EUSER) CONNECT(CERTAUTH RING(E2EAgentKeyring)
LABEL('E2EAdapterCA') USAGE(CERTAUTH))
```

2. Verify that the E2E agent's keyring contains the correct certificates.

```
RACDCERT ID(E2EUSER) LISTRING(E2EAgentKeyring)
```

The following screen capture shows the LISTRING response:

```

Ring:
>E2EAgentKeyring<
Certificate Label Name      Cert Owner      USAGE      DEFAULT
-----
E2EAgentCA                CERTAUTH        CERTAUTH    NO
E2EAgentServerCert        ID(E2EUSER)     PERSONAL    NO
E2EAdapter                 CACERTAUT      CERTAUTH    NO

```

3. Activate the RACF definition.

```
SETROPTS RACLIST(DIGTCERT) REFRESH
```

Configure the E2E Automation SSL Properties File

There are the following E2E adapter client interfaces with different SSL property files:

- SMU server (runs on Linux)
- E2E agent (runs on z/OS)
- In addition, the E2E adapter itself needs to be configured with the certificates

They all have in common an SSL properties file with the following key and value pairs:

```

eez-ssl-truststore = path_client_truststore
eez-ssl-keystore = path_client_keystore
eez-ssl-keystore-password = passphrase
eez-ssl-keystore-alias = alias_of_client_ca

```

The real names for the placeholders *path_client_truststore*, *path_client_keystore*, and *alias_of_client_ca* depend on the type and the platform of the client.

SMU Server (Linux)

The truststore and keystore of the SMU server resides in

- /etc/opt/IBM/smsz/ing/cfg/ssl/SMU.truststore.jks
- /etc/opt/IBM/smsz/ing/cfg/ssl/SMU.keystore.jks

Modify the keytool commands used in “Keytool — SMU” on page 63 accordingly to create and copy the certificates, keystore and truststore.

Modify the parameters of the SMU SSL properties file using the SMU Automation configuration tool `cfgsmu`.

```
eez-ssl-truststore = /etc/opt/IBM/smsz/ing/cfg/ssl/SMU.truststore.jks
eez-ssl-keystore = /etc/opt/IBM/smsz/ing/cfg/ssl/SMU.keystore.jks
eez-ssl-keystore-password = passphrase
eez-ssl-keystore-alias = smuca
```

Note: The case of the `eez-ssl-keystore-alias` parameter should be consistent with the alias of the SMU CA certificate. If SSL handshake doesn't work with the alias name, list the keystore or truststore to check how the alias name of the certificate is stored. It might be that the alias name is not correctly specified in the SMU SSL properties file.

E2E Agent (z/OS)

The E2E agent runs on z/OS only. Therefore, the keystore and truststore and the definitions of the certificates can be done via RACF.

Modify the E2E agent SSL properties file:

<custom-root>/adapter/config/inge2eagt.ssl.properties

```
eez-ssl-truststore = safkeyring://E2EUSER/E2EAgentKeyring
eez-ssl-keystore = safkeyring://E2EUSER/E2EAgentKeyring
eez-ssl-keystore-password = passphrase
eez-ssl-keystore-alias = E2EAgentServerCert
```

If you omit the user ID in the `safkeyring` specification string, then the user ID of the currently running E2E agent job is used. For example, you can specify:

```
eez-ssl-truststore = safkeyring:///E2EAgentKeyring.
eez-ssl-keystore = safkeyring:///E2EAgentKeyring
```

E2E Adapter (z/OS)

The E2E adapter runs on z/OS. Therefore, the keystore and truststore and the definitions of the certificates can be done via RACF.

Modify the E2E adapter SSL properties file

<custom-root>/adapter/config/ing.adapter.ssl.properties

```
eez-ssl-truststore = safkeyring://E2EUSER/E2EAdapterKeyring
eez-ssl-keystore = safkeyring://E2EUSER/E2EAdapterKeyring
eez-ssl-keystore-password = passphrase
eez-ssl-keystore-alias = E2EAdapterServerCert
```

In addition, set the following property to true.

```
eez-remote-contact-over-ssl = true
```

This property is located at <custom-root>/adapter/config/ing.adapter.properties.

If you omit the user ID in the `safkeyring` specification string, then the user ID of the currently running E2E adapter job is used. For example, you can specify:

```
eez-ssl-truststore = safkeyring:///E2EAdapterKeyring
eez-ssl-keystore = safkeyring:///E2EAdapterKeyring
```

Additional RACF Profile Access for E2EUSER

If the E2E adapter or agent started task runs with E2EUSER but you define the keyring with another ID different from E2EUSER, you must give E2EUSER additional RACF profile access.

Procedure

Define the following RACF profile access for ID E2EUSER.

```
RDEFINE FACILITY IRR.DIGTCERT.LIST UACC(NONE)
RDEFINE FACILITY IRR.DIGTCERT.LISTRINGUACC(NONE)
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ID(E2EUSER) ACC(READ)
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(E2EUSER) ACC(READ)
```

Chapter 14. Problem Determination of the End-to-End Automation Adapter

This information describes methods for determining and solving various problems that you might encounter with end-to-end automation adapter. It includes descriptions of methods to analyze:

- [“The Java log file” on page 71](#)
- [“Problems with selecting the primary agent” on page 72](#)
- [“Problems with the program-to-program interface” on page 73](#)
- [“Problems with NetView event/automation service” on page 73](#)
- [“Problems with consumer registration” on page 74](#)
- [“Link list errors” on page 74](#)
- [“TCP/IP bind socket problem” on page 75](#)
- [“Timeout problems” on page 75](#)
- [“USS problems” on page 76](#)
- [“CHECK_HEALTH function failure” on page 76](#)
- [“Automation domain remaining unavailable after error recovery” on page 76](#)

The Java log file

The automation adapter writes Java log files into the following directories:

- /custom-root/data/eez/log
- /custom-root/data

The trace and event log files are written only if debugging is switched on.

The message log file can be accessed via the automation adapter Web-based graphical user interface (WebUI).

The trace log files are to be used by IBM Support for debugging purposes. Use the IBMSupport parameter of the ingadapter.sh USS script to generate all of these files as a zip file in the data directory.

All log and trace files are in UTF-8 format. If you use an FTP client and an editor that can handle UTF-8, you should not have any problems reading the logs.

In a telnet session to z/OS USS you can use the following command (although you might lose a few characters):

```
view -a filename
```

In z/OS you must convert the files from UTF to EBCDIC using the command:

```
iconv -f utf-8 -t ibm-500 traceFlatAdapter.log > new.log
```

Diagnosis of the SA z/OS tasks for the adapter

Use command DISPAOPS if initialization of the domain fails (and message AOF233E is issued) to check the existence and status of the automated operator function EVTOPER.

Output is similar to that shown in [Figure 13 on page 72](#).

```

AOFK2SO          SA z/OS - Command Dialogs          Line 1 of 33
Domain ID = IPSNP ----- DISPAOPS ----- Date = 11/25/04
Operator ID = OPER1                               Time = 11:08:30

  System      Automated
  Function    Primary   Status   Secondary  Status
-----
KEY4         EVTOPER   AUTEVT1  ACTIV     AUTEVT2    ACTIV
KEY4         E2EOPER   AUTE2E   ACTIV
KEY4         E2E0PR01 AUTE2E01  ACTIV
KEY4         E2E0PR02 AUTE2E02  ACTIV
KEY4         E2E0PR03 AUTE2E03  ACTIV

Command ==>
PF1=Help      PF2=End      PF3=Return   PF6=Roll
               PF8=Forward  PF9=Refresh  PF12=Retrieve

```

Figure 13. DISPAOPS command response

End-to-end automation adapter diagnosis commands

If necessary, you can switch on or off automation adapter trace using the following commands:

```

INGE2E SETDBGVLV NORMAL
INGE2E SETDBGVLV NORMAL,EXP,SEC,API
INGE2E DELDBGVLV ALL

```

If necessary, you can collect debug information using the following command:

```

INGE2E DEBUG SIZE=LONG EAS=eas_jobname TCPIP=tcp_jobname

```

The debug information that is collected is written to DSILIST(INGE2E00).

If you do not specify the EAS parameter, E/AS information is not retrieved.

If you do not specify the TCPIP parameter, it is taken from CNMSTYLE.TCPNAME

Problems with selecting the primary agent

To check whether the NetView agent has been selected as the primary agent to work with the automation adapter, enter the INGAMS command to display the primary agent flag, as shown in Figure 14 on page 72. The value YES in the last column indicates that the primary agent flag has been set for the agent.

```

INGKYAM0          SA z/OS - Command Dialogs          Line 1 of 2
Domain ID = IPSNP ----- INGAMS ----- Date = 02/20/17
Operator ID = OPER1                               Sysplex = KEY1PLEX Time = 17:01:49

Cmd:  A Manage      B Show Details  C Refresh Configuration  D Diagnostic

Cmd  System  Member  Role  Status  Sysplex  XCF-Group  Release  Comm  E2E
----
_   KEY4    KEY4$$$$3  PAM  READY  KEY1PLEX  INGXSG34  V4R1M0  XCF
_   KEY4    KEY4      AGENT  READY  KEY1PLEX  INGXSG34  V4R1M0  XCF  YES

Command ==>
PF1=Help      PF2=End      PF3=Return   PF6=Roll
               PF9=Refresh  PF12=Retrieve

```

Figure 14. INGAMS command dialog panel

Problems with the program-to-program interface

To display status and buffer queue information for a PPI receiver, use the NetView command:

```
DISPPI ALL|RCVRID=receiver_id
```

or, from an MVS console:

```
MODIFY ssiname,DISPPI ALL|RCVRID=receiver_id
```

where *ssiname* specifies the MVS subsystem interface name and *receiver_id* the PPI receiver ID.

Problems with NetView event/automation service

To display operational characteristics of the event/automation service, use the MVS command:

```
MODIFY jobname,DISPLAY,STATUS|QSTATS
```

The STATUS option displays information about:

- The operational status of each service
- The connectivity of the service outside of the event/automation service address space via PPI and TCP/IP

The QSTATS option displays information about the number of data buffers sent and received by all of the subtasks that make up the event/automation service, and the total number of data buffers that are currently queued to a subtask and that have not been processed.

To display the configuration parameter of the NetView message adapter, use the MVS command:

```
MVS F NETEVE2E,SETTINGS,TASK=MESSAGEA
```

For more details concerning the information that is displayed see *IBM Tivoli NetView for z/OS Command Reference Vol. 1*.

Examples

1. At the NetView console entering:

```
MVS F NETEVE2E,DISPLAY,STATUS
```

displays information as shown in [Figure 15 on page 73](#).

```
IHS0141I SERVICE   STATUS   ADDITIONAL INFO
IHS0142I -----   -----   -----
IHS0143I ALERTA    DOWN    connection idle
IHS0143I MESSAGEA  UP      *
IHS0143I EVENTRCV  DOWN    *
IHS0143I TRAPALRT  DOWN    *
IHS0143I ALRTTRAP  DOWN    *
IHS0143I PPI       UP      INGEVE2E
IHS0143I TCP/IP     UP      9.152.87.64
```

Figure 15. Event/automation service STATUS information

2. At the NetView console entering:

```
MVS F NETEVE2E,DISPLAY,QSTATS
```

displays information as shown in [Figure 16 on page 74](#). You can use this to find out where the event flow has broken down.

```

IHS0145I TASK      QCOUNT  TOTAL SENT  TOTAL RCVD
IHS0146I -----
IHS0147I CONTROL  0          23          0
IHS0147I ALERTA   0          23          0
IHS0147I MESSAGEA 0          0          23
IHS0147I EVENTRCV 0          0          0
IHS0147I TRAPALRT 0          0          0
IHS0147I ALRTRAP  0          0          0

```

Figure 16. Event/automation service QSTATS information

Problems with consumer registration

In order to be receive events, end-to-end automation has to register as an event consumer.

To check whether the end-to-end automation was correctly registered at initialization, use the following command:

```
INGEVOBS LIST
```

The output should be as shown in [Figure 17 on page 74](#).

```

Consumer      Events  Exit      AUTOOPER  User data
-----
E2E           1,2    INGRTECS          INGEVE2E
*** End of Display ***

```

Figure 17. INGEVOBS LIST command response

It should list E2E as an event consumer and display the clist INGRTECS as the exit. The User data column shows the PPI receiver ID that has been defined for the event/automation service. The default value is INGEVE2E and you can change it with the advanced automation option AOF_E2E_EAS_PPI before starting the automation adapter. For details see [“SA z/OS Advanced Automation Options” on page 38](#).

Link list errors

Before running the automation adapter make sure that the link list contains the necessary libraries.

Use the command:

```
D PROG,LNKLST
```

and check the existence of the following libraries:

```

SYS1.SCLBDLL2
SYS1.SCEERUN
SYS1.SCEERUN2
SYS1.SCLBDLL
hlq.SINGMOD1

```

Otherwise you might get an error similar to the following:

```

CEE3501S The module INGIOC was not found.
          From entry point __dllstaticinit at compile unit offset +000005EE
          at entry offset +000005EE at address 2C936316

```

Alternatively, edit the ingadapter.sh start script and add these libraries as follows:

```
export STEPLIB=SYS1.SCLBDLL2:SYS1.SCEERUN:SYS1.SCEERUN2:SYS1.SCLBDLL:hlq.SINGMOD1:$STEPLIB
```

TCP/IP bind socket problem

After stopping the automation adapter, if you then restart it immediately, you might receive a message similar to the following (note that the port number is customizable):

```
EEZA0061E Unable to bind a socket to address BOEKEY4 at port 2001.  
Reason: EDC8115I Address already in use.
```

Although the connection has automatically been freed by TCP/IP, this occurs because the SMU server or end-to-end agent has not yet closed the socket to the terminated automation adapter.

To drop the connection:

1. Enter the TSO command NETSTAT (PO *nnnn*) to list connections for port *nnnn*, for example:

```
EZZ2350I MVS TCP/IP NETSTAT CS V1R5          TCPIP Name: TCPIP          16:29:23  
EZZ2585I User Id  Conn      Local Socket          Foreign Socket      State  
EZZ2586I -----  ----  -----  
EZZ2587I CU11ADAP 0000D0B2 9.152.87.64..2001    9.152.172.117..1365  FinWait2
```

2. Enter the TSO command:

```
TSO NETSTAT,DROP CONNID
```

where *CONNID* is the connection ID in the column Conn. For example, the command TSO NETSTAT ,DROP 0000D0B2 should produce the following output:

```
EZZ2622I Connection successfully dropped
```

Timeout problems

Timeout problems in an E2E automation environment may occur if you access the data of a remote SA z/OS domain or remote SA z/OS resources.

The data that is displayed has to be requested from the automation adapter. These end-to-end automation requests are mapped to SA z/OS commands that query the requested data from the automation manager. Before sending the query commands to the automation manager, the primary automation agent checks the expiration time for the end-to-end automation requests. If the expiration time that remains is too short, the requested command is rejected, resulting in error message ING249E that indicates that a task execution request timed out.

The expiration time of an end-to-end automation request is determined by where it is issued, as follows:

Request issued from:	Expiration time determined by:
SMU Server	The timeout interval that is defined in the WebSphere® Admin Console for the environment variable com.ibm.eez.aab.invocation-timeout-seconds
The automation adapter (such as INIT_DOMAIN)	The timeout interval that is defined in the automation adapter ing.adapter.plugin.properties configuration file
E2E agent	The timeout interval INGAGT_CON_TIMEOUT that is defined in the automation agent inge2eagt.properties configuration file

It is possible that an end-to-end automation request that has been issued might be queued in the automation adapter or the NetView operator task for a certain period of time. You should therefore make the timeout interval large enough to take this delay into account.

For details about the relationship between the expiration time and the WAIT parameter of SA z/OS commands, see [The WAIT Parameter](#).

USS problems

To avoid problems with loading DLLs in shared memory, it is recommended that you always have the Java Just-In-Time (JIT) compiler enabled. That is, you should never set (or always leave unset) the JAVA_COMPILER environment variable.

USS Shared File System

If the USS File System is shared between the systems where the automation adapter might be moved from and to, then the old and the new automation adapters write into the same files stdout.txt and stderr.txt.

This might result in truncated output lines.

You can avoid this by using the parameter suffix in the automation adapter start command, for example,

```
ingadapter.sh start SYS1
```

Refer to [Chapter 11, “Operating the End-to-End Automation Adapter,”](#) on page 53.

CHECK_HEALTH function failure

When you stop NetView or the SSI Subsystem you should recycle the automation adapter.

See the description of REFRESHSTART in [“Startup procedure INGADPT”](#) on page 85.

If you do not recycle the automation adapter, any CheckHealth command that is periodically received from the SMU server will result in error messages ING9905E and ING9820E. This may cause many error messages to be written to the SYSLOG.

Automation domain remaining unavailable after error recovery

When restarting the automation manager or the INGPXDST communication task, during which the asynchronous communication path from the primary agent to the automation adapter is still available, the end-to-end automation manager is notified about the unavailable automation domain.

If there are problems notifying the end-to-end automation manager about the recovered error situation, the domain may remain unavailable rather than become available.

When creating the ready event for the end-to-end automation manager, requests sent to the automation manager may time out. In this case the ready event is not sent and the adapter online event is also not sent to the end-to-end automation manager.

By default, the primary agent waits 90 seconds for the response. A longer timeout interval can be provided in the common global variable INGREQ_WAIT.

The following error messages relating to these timeouts are written to the NETLOG:

```
ING008I INGPYAMC SERVICE FAILED, RC=2 - REASON=TIMEOUT MOD=INGRYEVT - GET_QUAL_RESOURCE_RELS  
(INGEVOBS)  
ING235W No information from automation manager. READY event not created for E2E.  
ING276E Execution of a service timed out.
```

The command INGE2E RESYNC can also be used to notify the end-to-end automation manager about the recovered error situation.

Problem Determination of the End-to-End Automation Agent

This information describes methods for determining and solving various problems that you might encounter with end-to-end automation agent. It includes descriptions of methods to analyze:

- [“The Java Log File” on page 77](#)
- [“End-to-End Agent Diagnosis Commands” on page 77](#)
- [“End-to-End Agent Loses Connection Information” on page 77](#)
- [“Mixed SA Environment with End-to-End Agent” on page 77](#)

The Java Log File

The end-to-end agent writes Java log files into the following directories:

- /custom-root/data/ing/log
- /custom-root/data

The trace files are written only if debugging is switched on.

End-to-End Agent Diagnosis Commands

If necessary, you can switch on the end-to-end agent trace using the following commands:

INGE2E DEBUG_E2EAGT ON

This command switches on the Java logger and the REXX debug statements at the same time.

INGE2E DEBUG_E2EAGT OFF

This command switches off debugging the end-to-end agent.

End-to-End Agent Loses Connection Information

If you perform a COLD start of the primary automation manager and you restart the E2E agent, then the E2E agent loses the TCP/IP connection information of all remote domains. The observed status of the DMN objects will be set to SYSGONE. The reason is that the E2E agent saves the TCP/IP connection information in the takeover file of the PAM. After a COLD start, all information in the takeover file is lost.

If you load a new configuration file without e.g. DMN00001 and later on you load again the original configuration file that contains the DMN00001, the connection information to DMN00001 is lost as well.

The resolution of above two cases is to recycle all automation adapters on the remote automation domains.

Mixed SA Environment with End-to-End Agent

If you have an SAplex with an E2E agent running on SA z/OS version 4.1 and you want to start an end-to-end automation adapter, you must start the end-to-end automation adapter on SA z/OS version 4.1 as well.

Appendix A. Using IBM Tivoli System Automation Application Manager

IBM Tivoli System Automation Application Manager can be used to:

- Automate the operation of resources within heterogeneous environments.
- Manually operate resources within heterogeneous environments.

The SA z/OS end-to-end automation adapter relates both to Service Management Unite (SMU) and IBM Tivoli System Automation Application Manager. If you use IBM Tivoli System Automation Application Manager to automation E2E solutions, you can read information about SMU in this whole book, and any mentioning of SMU server or Service Management Unite server can be exchanged with IBM Tivoli System Automation Application.

The following information is specific to IBM Tivoli System Automation Application Manager.

The IBM Tivoli System Automation Application Manager library

The following table shows the information units in the [IBM Tivoli System Automation Application Manager library](#):

Title	Order Number
<i>IBM Tivoli System Automation Application Manager Administrator's and User's Guide</i>	SC34-2701
<i>IBM Tivoli System Automation Application Manager Installation and Configuration Guide</i>	SC34-2702
<i>IBM Tivoli System Automation Application Manager Reference and Problem Determination Guide</i>	SC34-2703

Connecting IBM Tivoli System Automation Application Manager

Figure 18 on page 80 shows that end-to-end automation adapter for z/OS connects SA z/OS to the IBM Tivoli System Automation Application Manager. IBM Tivoli System Automation Application Manager provides automation capabilities for resources running on distributed automation domains.

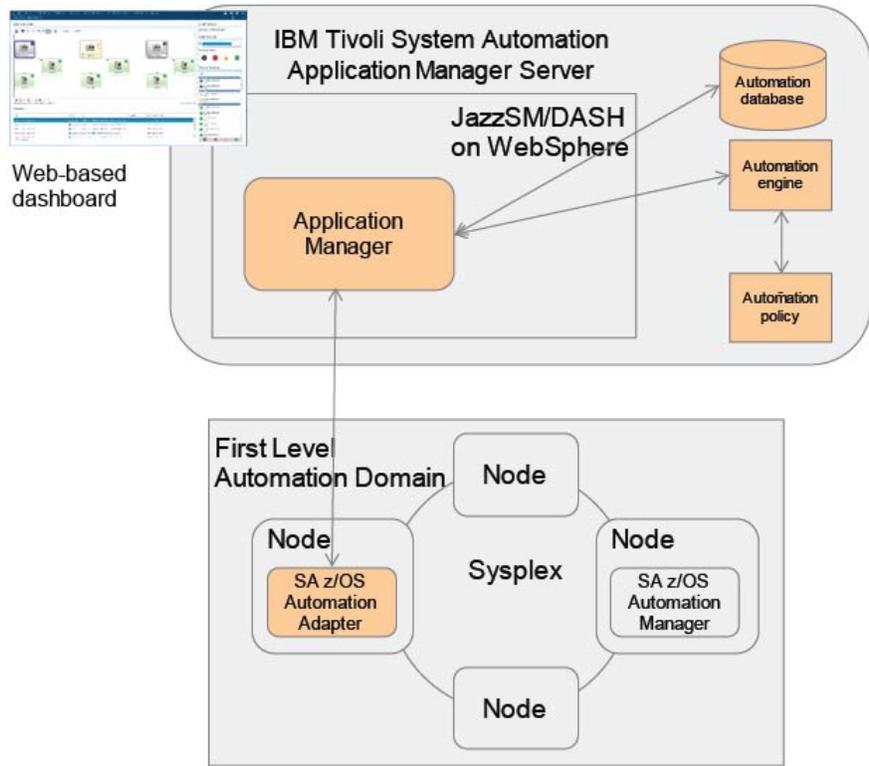


Figure 18. Automation Adapter Connected to IBM Tivoli System Automation Application Manager

Appendix B. High Availability of the End-to-End Automation Adapter

This information describes how to define the automation adapter as an application to be automated by SA z/OS for simplified operating and high availability.

Limitations to the automation

1. If the E/AS message adapter is stopped separately with the command `F E2EEAS,STOP TASK=MESSAGEA`, events might be lost unnoticed by the automation because E/AS still receives incoming events and discards them.

To prevent losing events in this way, always stop the message adapter by stopping E2EEAS. The automation then notices the outage and tries to make the E2EEAS application available again by restarting it in the same system or by moving it together with the automation adapter to a backup system.

2. No actions are taken when NetView is stopped on the primary agent, because it is assumed that NetView is usually down only for a short time. Thus the automation adapter does not have to be moved to another system.

If a longer outage of NetView is planned, the automation adapter has to be moved manually to another system in the sysplex before NetView is stopped.

Defining automation policy

In the following description of the automation policy definitions, fields of the customization dialog where no values have to be specified are not mentioned.

After having customized the automation policy, the BUILD process has to be executed to create the control files, and the configuration has to be refreshed with the INGAMS command.

■ A sample policy is available for the end-to-end automation adapter.

■ The customer may import the following PDB entries within *E2E via component 'E2E Automation Adapter'.

Application definitions for the NetView event/automation service (E2EEAS)

Defines an application for the NetView event/automation service for each system in the sysplex.

It has defined HASPARENT relationships to TCP/IP and the NetView SSI to enforce the right start and stop order, because EAS depends on TCP/IP and the PPI function of the NetView SSI. It also defines a MAKEUNAVAILABLE relationship to the E2E automation adapter (E2EADPT) to ensure that E2EEAS is stopped after E2EADPT is down.

It is a member of the application group E2EADPT (type SYSTEM and nature BASIC) together with the E2E automation adapter to ensure that both are up and running.

Application definitions for the end-to-end automation adapter (E2EADPT)

Defines an application for the automation adapter for each system in the sysplex.

It is a member of the application group E2EADPT (type SYSTEM and nature BASIC) together with the E2EEAS to ensure that both are up and running.

It defines the following relationships:

- A MAKEAVAILABLE/WhenRunning relationship to E2EEAS to ensure that the E2EADPT started after the E2EEAS is already running, because the automation adapter depends on the E/AS message adapter service.
- A FORCEDOWN relationship to E2EEAS to enforce a shutdown of the automation adapter if the EAS resource fails and cannot be restarted on the same system.
- A FORCEDOWN relationship to the NetView SSI to enforce a shutdown of the automation adapter when the NetView SSI stops, because the communication between the automation adapter and the primary agent depends on the PPI of the NetView SSI.

Startup definitions and sequence

The automation adapter manages a plug-in for z/OS that is started and stopped by the automation adapter. It runs in the same address space as the automation adapter and issues its own messages. During a successful startup of the automation adapter, the following messages are issued, indicating the progress of the startup.

```

1 S INGXADPT,JOBNAME=E2EADPT,F=START
IEF695I START INGXADPT WITH JOBNAME E2EADPT IS ASSIGNED TO USER STCUSER, GROUP
STCGROUP
$HASP373 E2EADPT  STARTED
2 IEF403I E2EADPT - STARTED
3 INGX9704I Preparing the environment to start the automation
adapter using custom root directory /var/ing/adapter
$HASP373 BPXAS  STARTED
IEF403I BPXAS - STARTED
BPXP024I BPXAS INITIATOR STARTED ON BEHALF OF JOB E2EADPT RUNNING IN ASID 00A0
4 EEZC0003I Base output directory for serviceability related files (for
example,
message log files and trace files) has been set to /var/ing/adapter/data/eez/*.
5 EEZA0100I The adapter has been started
6 EEZA0101I The adapter is active
EEZA0111I The plug-in is starting: class com.ibm.ing.sam.INGXPlugin
INGX9802I INGXLogger has successfully been initialized using configuration
file ing.adapter.jlog2.properties from path /var/ing/adapter/config.
INGX9902I INGXPluginLogger has successfully been initialized using configuration
file ing.adapter.jlog2.properties from path /var/ing/adapter/config.
IHS0029I Message Adapter: The current ServerLocation is location 1,
address 127.0.0.1, port 5530.
7 EEZA0112I The plug-in has been started: class com.ibm.ing.sam.INGXPlugin
8 EEZA0102I The adapter is ready
9 EEZA0116I The status of the event sender changed:
Address=smu.company.com/9.152.135.22, Port=2002, Status=1 *
EEZA0118I The connection to the management server
smu.company.com : 2002 has been established.*
EEZA0116I The status of the event sender changed:
Address=/9.152.86.237, Port=2003, Status=2 *
EEZA0118I The connection to the management server
9.152.86.237 : 2003 has been established.*

```

The explanations of these messages are as follows:

- 1** The start command for the end-to-end automation adapter.
- 2** Message IEF403I is used to change the agent status of the automation adapter to ACTIVE.
- 3** Message INGX9704I indicates the custom root directory that is used by the automation adapter.
- 4** Message EEZC0003I indicates the location of the trace and log files of the automation adapter.
- 5** The first message that is issued by the automation adapter is also used to change the agent status of the automation adapter to ACTIVE.
- 6** The automation adapter has successfully established the connection to the management server and now continues initialization by starting the plug-in.
- 7** Message EEZA0112I indicates that the plug-in has successfully initialized the domain. This message is used to change the agent status of the automation adapter to UP.

8 Message EEZA0102I indicates that the automation adapter startup sequence is complete.

9 The defined targets – SMU server or end-to-end agent, or both – are listed when the connection is established.

If the startup of the automation adapter or the plug-in fails, the automation adapter stops after it issues an error message. SA z/OS has to be told whether the error message indicates a recoverable abnormal end of the application so that the automation will try to restart the application. The status BROKEN indicates that an unrecoverable abend and automation will not try to restart the automation adapter.

During a failed startup of the automation adapter, the following messages might be issued, for example:

```
1 S INGXADPT,JOBNAME=E2EADPT,F=START
$HASP100 E2EADPT ON STCINRDR
IEF695I START INGXADPT WITH JOBNAME E2EADPT IS ASSIGNED TO USER STCUSER, GROUP
STCGROUP
$HASP373 E2EADPT STARTED
2 IEF403I E2EADPT - STARTED
3 INGX9704I Preparing the environment to start the automation adapter using
custom root directory /var/ing/adapter
4 EEZC0003I Base output directory for serviceability related files (for example,
message log files and trace files) has been set to /var/ing/adapter/data/eez/.
5 EEZA0100I The adapter has been started
EEZA0101I The adapter is active
EEZA0111I The plug-in is starting: class com.ibm.ing.sam.INGXPlugin
INGX9802I INGXLogger has successfully been initialized using configuration file
ing.adapter.jlog2.properties from path /local/SA/410/srv/usr/lpp/ing/adapter/
config.*
INGX9902I INGXPluginLogger has successfully been initialized using
configuration file
ing.adapter.jlog2.properties from path /local/SA/410/srv/usr/lpp/ing/adapter/
config.*
6 INGX9905E Adapter function INIT_DOMAIN failed: INGX9820E JNI function
ingjppi failed with return code 1726. ->com.ibm.ing.util.INGXBadJNIExc
eption: INGX9820E JNI function ingjppi failed with return code 1726.
7 EEZA0115I The plug-in startup failed: class com.ibm.ing.sam.INGXPlugin
INGX9904E Adapter plug-in initialization was
unsuccessful.

EEZA0037E The adapter stops running because no plug-in has been successfully
initialized
EEZA0103I The adapter is
stopping*

EEZA0113I The plug-in is stopping: class
com.ibm.ing.sam.INGXPlugin

EEZA0114I The plug-in has been stopped: class
com.ibm.ing.sam.INGXPlugin

EEZA0060I The termination of the adapter is delayed for 5 seconds
8 EEZA0032E Initialization of the adapter failed
9 EEZA0105I The adapter has been stopped due to a failure,
rc=20

INGX9703E Script ingadapter.sh terminated unsuccessfully.
10 IEF404I E2EADPT -
ENDED

$HASP395 E2EADPT ENDED - RC=0000
```

The explanations of these messages are as follows:

1 The start command for the automation adapter.

2 Message IEF403I is used to change the agent status of the automation adapter to ACTIVE.

3 Message INGX9704I indicates the custom root directory that is used by the automation adapter.

4 Message EEZC0003I indicates the location of the trace and log files of the automation adapter. Have a look in these files to investigate the start failure of the automation adapter.

- 5 The first message that is issued by the automation adapter is also used to change the agent status of the automation adapter to ACTIVE.
- 6 Message INGX9905E points to the original error situation that message INGX9820E with return code 1726 has a mismatch in the XCF group ID definition. This message changes the status of the automation adapter to BREAKING.
- 7 Message EEZA0115I indicates the error situation that the plug-in definition is wrong. This message changes the status of the automation adapter to BREAKING.
- 8 Message EEZA0032E changes the status of the automation adapter to BROKEN.
- 9 Message EEZA0105I changes the status of the automation adapter to ABENDED.
- 10 Message IEF404I is used as the final termination message.

Defining status messages

The automation table provided by SA z/OS already includes statements for status updates that are based on the following messages that are issued by the automation adapter during startup or shutdown processing and in error situations.

Message ID	Status
EEZA0032E	BROKEN
EEZA0100I	ACTIVE
EEZA0103I	TERMINATING
EEZA0104I	TERMINATED
EEZA0105I	ABENDED
EEZA0112I	UP
EEZA0115I	BREAKING

Defining messages to be captured

Messages that are issued by an application can be captured for subsequent display with the DISPINFO command. Capturing the messages that are issued by the automation adapter helps in analyzing problems with the automation adapter.

The following messages are defined and captured (with an appropriate severity):

- EEZA0037E
- EEZA0038E
- EEZA0055E
- EEZA0056I
- EEZA0057E
- EEZA0061E
- EEZA0101I
- EEZA0102I
- EEZA0111I
- EEZA0113I
- EEZA0114I
- EEZA9991E
- EEZA9992E

- INGX9705E
- INGX9707I
- INGX9709E
- INGX9711E

Startup procedure INGXADPT

The sample procedure INGXADPT that is provided to start the automation adapter uses the value of parameter F to decide whether to start or stop the automation adapter with the same procedure. You must therefore specify parameter F as an additional parameter for the default startup command. If you have customized the sample procedure to accept further parameters, append them to this list.

Special considerations

In case of:

- A recycle of the primary agent
- A configuration refresh

a recycle of the application group E2EADPT_X which results in a recycle of automation adapter on the same system is needed. This is ensured by the command INGE2E RECYCLE E2EADPT_X/APG in the 'refreshstart' section of policy STARTUP.

Application group definitions for the automation adapters (E2EADPT and E2EADPT_X)

E2EADPT defines an application group of type SYSTEM and nature BASIC. It is linked to each system and ensures that the E2EEAS and E2EADPT are running on the same system.

E2EADPT_X defines an application group of type SYSPLEX and nature MOVE to ensure that only one E2E automation adapter together with its EAS is running. The member of this APG is the E2EADPT application group of each system.

E2EADPT must be linked to the entries of type SYS which represent the systems in the SAplex.

E2EADPT_X must be linked to the entry of type GRP (with a group type of SYSPLEX) which represents the SAplex.

Automation purpose and rules

The automation adapter provides the interface between the end-to-end automation manager and SA z/OS as the first level automation.

It communicates with the local automation agent on the same system, selected as the primary agent. For synchronous communication it uses the program-to-program interface (PPI), for the asynchronous communication path the NetView event/automation service is used.

To optimize the availability of the end-to-end automation infrastructure, automation of the end-to-end automation adapter with SA z/OS can be focused on:

- Automatically restarting failed components that the automation adapter depends on
- Automatically restarting the automation adapter
- Failover of a non-recoverable automation adapter to a backup system
- Cleaning up the primary agent after automation adapter has been broken
- Enforcing resynchronization with the end-to-end automation manager if events have possibly been lost via the asynchronous communication path during outages

- Enforcing the recycling of the automation adapter if the primary agent has been recycled because of an error situation

Error situations and their resulting automation actions have the following impacts to the automation adapter:

- [“Non-disruptive recovery from errors” on page 86](#)
- [“Resynchronization of the automation adapter” on page 86](#)
- [“Recycling the automation adapter on the same system” on page 86](#)
- [“Restarting the automation adapter or failover to a backup system” on page 87](#)

Non-disruptive recovery from errors

Some error situations can be recovered by SA z/OS without interruption of the automation adapter.

These error situations are:

- A hot restart of the automation manager, when the takeover does not take longer than two minutes or the time interval provided by the advanced automation option AOF_E2E_TKOVN_TIMEOUT.
- A short outage of the NetView event/automation service during which no events failed to be transferred from the primary agent via PPI TECROUTE to the message adapter of the event/automation service.

The number of send retries can be customized with the advanced automation option AOF_E2E_EVT_RETRY. The default value is 60 retries at intervals of one second.

Resynchronization of the automation adapter

If events to be forwarded to the automation adapter are lost during error situations, information about first-level automation resources that is collected by the end-to-end automation manager cannot be kept up to date.

As long as the asynchronous communication path from the primary agent to the automation adapter is available despite the error situation, the end-to-end automation manager is notified that the automation domain is offline. As soon as the end-to-end automation manager is informed that the domain is online again, the end-to-end automation manager has to request initial status information for all resources known by it in the affected automation domain. Such a resynchronization of the end-to-end automation manager with the first-level automation might cause increased traffic rate.

Error situations that the automation adapter has to be resynchronized after are:

- A stop or abend of the NetView INGPXDST communication task
- A recycle of the automation manager, when the start type is WARM or COLD. In case of a HOT restart, a resynchronization is only necessary if the outage took longer than two minutes or the time interval that is defined with the advanced automation option AOF_E2E_TKOVN_TIMEOUT.

Recycling the automation adapter on the same system

If the asynchronous communication path between the automation adapter and the primary agent is interrupted, the end-to-end automation manager cannot be notified about the error situation via this communication path.

Therefore the automation adapter will be stopped and restarted by SA z/OS after it has recovered from the error situation.

Furthermore a recycle of the automation adapter is initiated by SA z/OS if the automation adapter needs to re-initialize the automation domain, for example, after a recycle of the primary agent.

Error situations that the automation adapter has to be recycled after are:

- A restart of the primary agent
- A configuration refresh
- ACF COLD

- An outage of NetView event/automation service, if events have been lost
- When the PPI rejects the events that are to be forwarded to the automation adapter

Restarting the automation adapter or failover to a backup system

When the automation adapter stops itself or the appropriate address space fails, resulting in a recoverable status, the automation adapter will be restarted by the automation as long as the specified critical threshold for the automation adapter has not been exceeded.

If restart retries are exhausted or the automation adapter terminates indicating a non-recoverable error, the automation initiates a failover to a backup system, if one is available in the same subplex.

More error situations when the automation adapter has to be moved to a backup system are:

- If restart retries of the NetView event/automation service are exhausted
- If the NetView PPI is down

Automated applications

Concerning the end-to-end automation adapter, the infrastructure considers the following two components to be defined as applications to SA z/OS:

- The NetView event/automation service
- The automation adapter

Grouping considerations

To guarantee that only one automation adapter is running at any given time within a subplex, a MOVE group is defined that contains the BASIC group E2EADPT (containing the E2EADPT and the E2EEAS application) on the different systems in the subplex. For more information, see [Figure 19 on page 87](#).

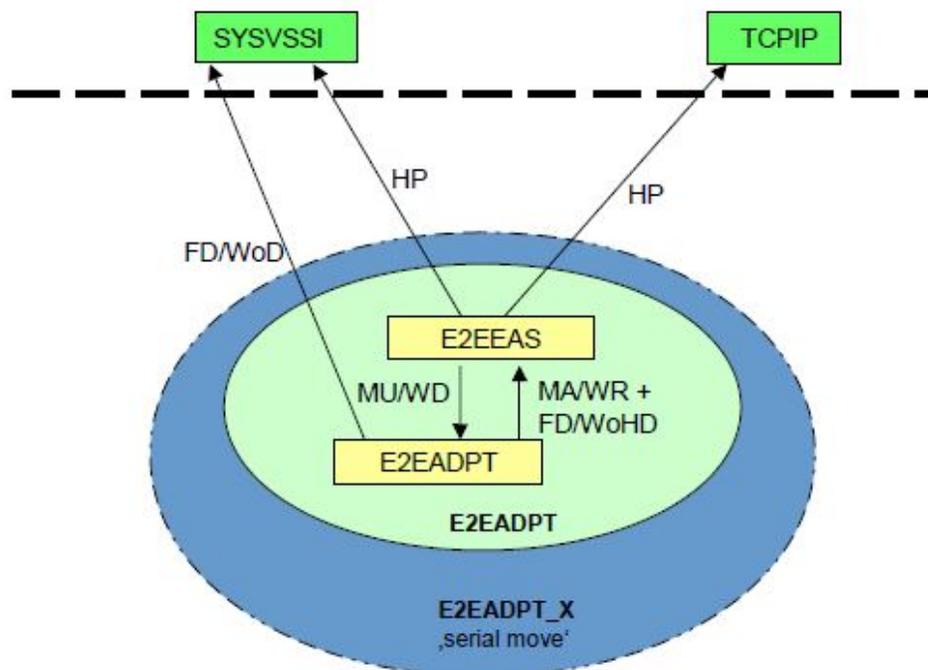


Figure 19. E2E automation adapter policy relationships

The following abbreviations for relationship types are used:

HP

HASPARENT

FD/WOHD

FORCEDOWN/WhenObservedHardDown

FD/WOD

FORCEDOWN/WhenObservedDown

MU/WD

MAKEUNAVAILABLE/WhenDown

MA/WR

MAKEAVAILABLE/WhenRunning

Appendix C. Configuration file parameters

The following tables provide details about all of the configuration file parameters.

Parameter Name	Default	Description
eez-initial-contact ¹	false	Value: true or false. For SMU server and the end-to-end agent, set eez-initial-contact to false.
eez-initial-contact-retry-interval-minutes ¹	0 min	Value range: 0, 1, ..., 1440 minutes. Zero (0) means never ending. Otherwise the automation adapter stops after the specified number of minutes. The automation adapter attempts to send an AdapterJoinEvent to the server every minute. If the SMU server responds with an acknowledgement request, the automation adapter continues with normal initialization.
eez-max-connections	3	Value range: 1 through 10. The number of concurrent working threads. Ideally, this should be the same as the number of NetView end-to-end automation tasks.
eez-operator-authentication	true	Value: true or false. If this is set to false, the automation adapter uses the SA z/OS-provided JAAS default login module for operator ID and password validation. That is, the following entry in the adapter JAAS configuration file, ing.adapter.jaas.properties, is used: <pre>EEZAdapterDefaultLogin { com.ibm.eez.adapter.EEZAdapterDefaultLoginModule required ; };</pre> However, if this parameter is set to true, the automation adapter uses the RACF-provided user ID checking routine. That is, the following entry in the adapter JAAS configuration file is used: <pre>EEZAdapterLogin { com.ibm.security.auth.module.OS390LoginModule required; };</pre>
eez-plugin-classpath ¹	none	An optional security feature. It specifies the directories that are used to load the backend and the plug-in Java classes. If specified, the plug-in is not loaded from the standard CLASSPATH. If not specified, the classes are loaded from the standard CLASSPATH. All other classes are loaded from the standard CLASSPATH.

Table 9. Master configuration file (*ing.adapter.properties*) parameters (continued)

Parameter Name	Default	Description
eez-remote-contact-activity-interval-seconds ¹	400 sec	Value range: 0, 1, ..., <i>max</i> Setting this parameter to 0 means it never expires. <i>max</i> indicates the maximum interval that can be defined in Java. The threshold that the automation adapter is stopped at if a synchronous request (including CHECK_HEALTH) is not received. Verification is triggered by an asynchronous received event.
eez-remote-contact-hostname	automatically	The host name or IP address that is used by the automation adapter to receive synchronous requests from the SMU server or end-to-end agent. Specify hostname for a VIPA environment or if the hostname cannot automatically be detected. Default is the hostname which is detected automatically by the command hostname.
eez-remote-contact-over-ssl	false	Value: true or false. The automation adapter uses SSL (true) or plain (false) socket connection for remote contact to the SMU server or end-to-end agent.
eez-remote-contact-port	none	Value range: 1 through 65535 The port that belongs to the host name that is defined in eez-remote-contact-hostname.
eez-stop-delay ¹	5 sec	Value range: 3 through 60 seconds. Setting this parameter delays the stopping of the automation adapter for the specified number of seconds. This gives the adapter time to deliver the leave event properly.
eif-cache	false	Value: true or false. Switch EIF caching on (true) or off (false).
eif-cache-size	500	Value: 1, 2, ... The maximum number of events that are cached if the event cannot be delivered. This parameter is only used if eif-cache=true.
eif-receive-from-hostname	none	The host name to be used by the automation adapter to receive EIF events. It must fit the naming convention of the original EIF ServerLocation parameter.
eif-receive-from-port	5529	The port that is to be used by the automation adapter to receive EIF events. It must fit the naming convention of the original EIF ServerPort parameter.
eif-retry-interval-seconds	30 sec	Value: 1, 2, ... The number of seconds until a disconnected EIF socket will attempt to reconnect.

Table 9. Master configuration file (ing.adapter.properties) parameters (continued)

Parameter Name	Default	Description
eif-send-to-hostname	SMUserver_host	The host name that is to be used by the automation adapter to send EIF events. It must fit the naming convention of the original EIF ServerLocation parameter.
eif-send-to-port	none	The port that is to be used by the automation adapter to send EIF events. It must fit the naming convention of the EIF ServerPort parameter.
eif-send-to-agent-hostname	E2Eagent_host	The host name of the E2E agent.
eif-send-to-agent-port	2003	The port of the E2E agent.
plugin-configfile-sa4zos	none	The name of the plug-in configuration file.

1

These parameters are not specified in the sample master configuration file that is supplied.

Note: Table 10 on page 91 lists only those parameters that you can update.

Table 10. Plug-in configuration file (ing.adapter.plugin.properties) parameters

Parameter Name	Default	Description
GRPID	None	This is a 2-character group ID. It must be equal to the group ID of the corresponding SA z/OS XCF group. See DSIPARM(INGXINIT).
PPIBQL	3000	Value: positive number The PPI buffer queue limit for the PPI queue that is used by each connection. The number of elements in the PPI queue—this indicates how large the response to a request may be. It should be greater than the number of queue elements that you expect to be returned.
AUTOPFN	None	Value: 8-character string This a NetView operator task name or an SA z/OS automated function name to be used by SA z/OS to execute the incoming automation adapter request.
TIMEOUT	600	Value: positive number Timeout in seconds that is used when initializing or stopping the domain and for any incoming request that has no timeout value of its own.
TIMEOUT_SA_NOTAVAIL	0	Value: positive number or zero Timeout value in seconds. A number greater than zero causes the plug-in to terminate if, after the time period that is specified, it detects that the SA z/OS NetView Agent is no longer available. The timeout is initially set to 180 seconds.
CODEPAGE	Cp1047	Value range: See Java code page definition. The code page that is used to convert Java strings to EBCDIC. This applies to SA z/OS resource name, owner field, comment field, and so on. The name of the code page must match the code page names supported by java.

Table 10. Plug-in configuration file (*ing.adapter.plugin.properties*) parameters (continued)

Parameter Name	Default	Description
AUDITMSG	true	If parameter AUDITMSG is set to true, then requests sent by SMU server or end-to-end agent and received by the automation adapter will be logged within the message file <code>msgAdapter.log</code> and <code>msgFlatAdapter.log</code> . If parameter AUDITMSG was set to false these information messages are not logged.
plugin-domain-name	?Domain	<p>Value: string up to 64 characters</p> <p>This is the name of the SA z/OS domain as it appears in the end-to-end automation operations console. In end-to-end automation, a SA z/OS domain is defined as a group of SA z/OS agents and managers that belong to the same XCF group ID (GRPID).</p> <p>All domains known by end-to-end automation must be unique. This includes domains that are defined within a sysplex as well as domains defined within UNIX clusters using System Automation Application Manager.</p> <p>If the parameter is omitted or the name begins with a ?, the plug-in generates a name that consists of the sysplex name and the <code>INGXSGnn</code> where <code>nn</code> is the XCF group ID GRPID.</p> <p>You can however pursue your own naming strategy.</p>

Table 11. SSL configuration file (*ing.adapter.ssl.properties*) parameters

Parameter Name	Description
eez-ssl-truststore	<p>The name of the truststore file that is used for SSL (optional).</p> <p>If omitted the truststore is used from the Java property <code>-Djavax.net.ssl.trustStore</code></p>
eez-ssl-keystore	<p>The name of the keystore file that is used for SSL (optional).</p> <p>If omitted the keystore is used from the Java property <code>-Djavax.net.ssl.keyStore</code></p>
eez-ssl-keystorepassword	<p>The password of the keystore file. It is required if the keystore file was specified.</p> <p>If omitted the keystore is used from the Java property <code>-Djavax.net.ssl.keyStorePassword</code></p>
eez-ssl-keystorealias	<p>The alias name of the certificate to be used by the server. If this is not specified the keystore file must contain only one entry, which is the one to be used. See Chapter 12, "Security Consideration for the End-to-End Automation Adapter," on page 57.</p>

Appendix D. SA z/OS specific implementation of the end-to-end automation resource data model

This information describes the SA z/OS specific implementation of the end-to-end automation resource data model.

Resource data displayed by the WebUI

This section describes the information that is displayed by the WebUI.

Automation Domains

It shows the name of the automation domains and its systems. For an SA z/OS domain, the systems are those that have been defined in the automation policy and linked to the sysplex group. The sysplex group is represented by the XCF group name `INGXSGnn` where `nn` is the corresponding group ID.

The following elements of the operations console are described:

- [“Domain object” on page 93](#)
- [“Node object” on page 94](#)
- [“Resource object” on page 95](#)

Domain object

A domain object shows general information for an automation domain.

Domain state

Available

SA z/OS is ready for automation.

Unavailable

SA z/OS is inhibited from automating resources or events are lost. The following are examples of error situations:

- The NetView task `INGPXDST` has stopped
- The automation manager has stopped within either at least 2 minutes or the time interval that is provided by the advanced automation option `AOE_E2E_TKOVN_TIMEOUT`.

Unknown

The automation adapter is not running.

Policy database name

The name of the automation policy database.

Policy file name

The name of the automation policy database file (ACF file name) that is currently loaded by the automation manager. An automation manager cold or warm start, or a configuration refresh might change this domain attribute.

Activation date

The date of the last refresh of the automation configuration as provided by the automation manager. If the configuration has not yet been refreshed, the start date of the automation manager is shown.

Activation time

The time of the last refresh of the automation configuration. If the configuration has not yet been refreshed, the start time of the automation manager is shown.

The domain object has additional properties such as:

Product name

The product name of System Automation for z/OS.

Version

The version of System Automation for z/OS.

Adapter name

The name of the end-to-end automation adapter.

Version

The version of the end-to-end automation adapter.

Node

The identifier for the system where the end-to-end automation adapter is running. This is the host name as defined with the eez-remote-contact-hostname parameter in the automation adapter configuration file, ing.adapter.properties.

Primary Manager StartType

The start type of the automation manager. Possible values are COLD, WARM, and HOT.

Primary Manager Name

The name of the primary automation manager, as shown by the primary agent via the INGAMS command.

XCF Group Name

The name of the XCF group that is associated with the automation manager and primary agent.

Sysplex Name

The name of the sysplex that the primary agent is located in.

Primary Agent Version

The SA z/OS version of the primary agent.

Primary Agent Name

The name of the primary agent, as shown by the primary agent via the INGAMS command.

Primary Agent NetView Version

The NetView version of the primary agent.

Adapter Operating System

The operating system that the automation adapter is running in.

Adapter Java Runtime Version

The Java runtime version that the automation adapter is running in.

Adapter Startup Time

The timestamp when the end-to-end automation adapter was started.

Adapter Plugin Startup Time

The timestamp when the end-to-end automation adapter plug-in was initialized.

Adapter Plugin Build Version

A unique identifier that defines the version of the end-to-end automation adapter plug-in and when it was built.

Primary Manager Startup Time

The start time of the primary automation manager.

Primary Manager System Name

The z/OS system name that the automation manager is running on.

Node object

A node object is a z/OS system or a node where the resource to be automated runs on.

The node object has the following properties:

Name

The name of the system or node.

Class

The resource type of the system or mode.

Description

The description as specified as the short description in the DESCRIPTION policy item of the automation policy.

Observed state

Unavailable if the observed state of this resource as detected by SA z/OS is SYSGONE; otherwise available.

Is included/excluded in automation

Excluded if there is a stop request to the resource system/SYG/system; otherwise included.

Include/Exclude node

The automation adapter converts these functions to the SA z/OS specific command. For details of these requests see [“Requests issued by the SMU and the end-to-end automation manager” on page 97.](#)

Resource object

The resource object has properties such as:

Name

The resource name consisting of name, type and optionally system or node.

Class

The resource type.

Automation domain

The name of the automation domain. If it is not specified in the plugin-domain-name parameter of the automation adapter plug-in configuration file, it is automatically generated as the concatenation of the sysplex name and the XCF group name.

Node

The z/OS system name.

Owner

The owner as specified in the APPLICATION INFO policy item of the automation policy.

Link

The link as specified in the APPLICATION INFO policy item of the automation policy.

Description

The description as specified as the short description in the DESCRIPTION policy item of the automation policy. Due to translation between UTF-8 and the EBCDIC code page, some characters might be lost. See also the CODEPAGE parameter in [“The Automation Adapter Plug-in Configuration file” on page 48.](#)

Observed state

The end-to-end automation-specific resource status information that is derived from the automation manager status of SA z/OS. For mapping details see [“Status mapping” on page 104.](#)

Desired state

The end-to-end automation-specific resource status information that is derived from the automation manager status of SA z/OS. For mapping details see [“Status mapping” on page 104.](#)

Request stop/start

The automation adapter converts these functions to SA z/OS specific commands. For details concerning these requests see [“Requests issued by the SMU and the end-to-end automation manager” on page 97.](#)

Cancel request

The automation adapter converts these functions to SA z/OS specific commands. For details concerning these requests see [“Requests issued by the SMU and the end-to-end automation manager”](#) on page 97.

View requests

This lists requests and votes in a similar way to INGVOTE.

Suspend/Resume automation

This allows you to enable and disable the automation flag of a resource that is maintained by the automation manager.

There are **additional properties** for a resource object:

There is resource status information about the automation manager for SA z/OS:

- The SA z/OS compound status
- The SA z/OS observed status
- The SA z/OS desired status
- The SA z/OS health status

It shows the nature if the resource is a group and it shows the SMFID of the corresponding system.

The Last Compound Status Change such as 2015-08-10 07:41:59 +0000 is the GMT date and time when the compound status of the resource has been changed.

Top level resource

Top level resources consist of:

- All z/OS systems in the sysplex as defined in the automation policy
- All members of type APL or APG that are attached to resources *system/SYG/system*, but not included as members in any other group, where *system* is any z/OS system of the sysplex as defined in the automation policy
- All Monitor resources (*MTR/*)

Relations to resource objects

When requesting relations the following relationship types are listed:

- MakeAvailable
- MakeUnavailable
- HasParent
- HasPassiveParent
- Externally
- HasMonitor
- ForceDown
- PrepAvailable
- PrepUnavailable

Request issued to resource objects

For a resource requests are shown.

Requested action

Start for MakeAvailable requests, stop for MakeUnAvailable requests.

Source

Source of the request, indicating who made the request, for example, OPERATOR or E2EMGR.

User ID

ID of the operator or automation manager that made the request.

Priority

Value Low, High or Force, depending on the priority assigned to the request in SA z/OS:

Low

Priority <2,000,000

High

2,000,000 ≤ Priority <3,000,000

Force

3,000,000 ≥ Priority <4,000,000

Created

The date and time when the request was issued.

Type

Type of the request, indicating whether it is a request or a vote.

Requests issued by the SMU and the end-to-end automation manager

The SMU server or end-to-end agent can send requests to the automation adapter.

These incoming end-to-end automation requests are mapped to SA z/OS commands as listed in [Table 12](#) on page 97.

<i>Table 12. Mapping of end-to-end automation requests to SA z/OS commands</i>	
End-to-end automation request name	SA z/OS command¹
Online	INGREQ <i>resource</i> REQ=START SCOPE=ONLY SOURCE= <i>source</i> USERID= <i>requestor</i> WAIT= <i>time</i> CMT= <i>comment</i>
Offline	INGREQ <i>resource</i> REQ=STOP SCOPE=ALL SOURCE= <i>source</i> USERID= <i>requestor</i> WAIT= <i>time</i> CMT= <i>comment</i>
CancelRequest	INGSET KILL <i>resource</i> REQUEST=* SOURCE= <i>source</i> WAIT= <i>time</i>
IncludeNode	INGSET CANCEL <i>system</i> /SYG/ <i>system</i> REQUEST=MAKEUNAVAILABLE SOURCE= <i>source</i> WAIT= <i>time</i>
ExcludeNode	INGREQ <i>system</i> /SYG/ <i>system</i> REQ=STOP SOURCE= <i>source</i> USERID= <i>requestor</i> WAIT= <i>time</i> CMT= <i>comment</i>
SetTarget	INGGROUP <i>resource</i> OUTMODE=LINE WAIT= <i>time</i> ACTION=ADJUST AVTGT= <i>value1</i> SATTGT= <i>value2</i> or INGGROUP <i>resource</i> OUTMODE=LINE WAIT= <i>time</i> ACTION=RESET
ResetFromNRE	SETSTATE <i>resource</i> AUTODOWN TARGET= <i>system</i>

Table 12. Mapping of end-to-end automation requests to SA z/OS commands (continued)

End-to-end automation request name	SA z/OS command ¹
SuspendResource	For SA z/OS V4.1: INGSUSPD <i>resource</i> REQ=SUSPEND USERID= <i>user</i> wait= <i>time</i> For SA z/OS V3.5: INGSET SET <i>resource</i> AUTOFLAG=NO WAIT= <i>time</i>
ResumeResource	For SA z/OS V4.1: INGSUSPD <i>resource</i> REQ=RESUME USERID= <i>user</i> wait= <i>time</i> For SA z/OS V3.5: INGSET SET <i>resource</i> AUTOFLAG=YES WAIT= <i>time</i>
MoveAway MoveHome MoveTo	INGMOVE <i>resource</i> OUTMODE=LINE WAIT= <i>time</i> TO=*ANY *HOME <i>system</i>
1. These commands also take the parameters OUTMODE=LINE and VERIFY=NO.	

MoveAway/MoveHome/MoveTo

These requests move a resource to:

- Any other system
- The resources's home systems
- A specific target system

Online/Offline/CancelRequest

The SOURCE can be either OPERATOR or E2EMGR.

IncludeNode/ExcludeNode

To exclude a node, a stop request is issued to the system group system/SYG/system with the effect that all resources known to the automation are stopped. This might also include the automation manager and the automation agent, if they are defined as usual to the automation.

A subsequent IncludeNode request is only possible if the automation is still running after having excluded the node. This comprises NetView and all dependant resources, the automation adapter and the automation manager.

ResetFromNRE

The ResetFromNRE request is to be used for resources that are in a non-recoverable error status. These resources are applications and monitor resources in the SA z/OS observed state HardDown, as described in “[Status mapping](#)” on page 104. Issuing the ResetFromNRE request might not be successful if the problem with the resource that produces the HardDown status has not been resolved.

The ResetFromNRE request is rejected for resources with agent status CTLDOWN.

SuspendResource/ResumeResource

These requests enable or disable the automation flag for a resource that is maintained by the automation manager.

WAIT parameter

The number of seconds for the WAIT parameter is calculated as the difference between the time when the SA z/OS command was issued within the NetView environment and the expiration time given to the end-to-end automation adapter request.

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

1. The time when the corresponding end-to-end automation request was issued from the SMU server or end-to-end agent.
2. The timeout in seconds defined in the SMU server's WebSphere Admin Console for the environment variable `com.ibm.eez.aab.invocation-timeout-seconds`

An end-to-end automation request might be queued in the automation adapter or the NetView operator task for a period of time. You should therefore make the environment variable `com.ibm.eez.aab.invocation-timeout-seconds` that is defined in the WebSphere Admin Console large enough to take this delay into account.

The calculated time difference is taken for the WAIT parameter. This value is shortened to 999 seconds if it exceeds this maximum value given by SA z/OS. If at the other side the calculated value for the WAIT parameter is less than a given minimum value, the requested command will be rejected.

This minimum value is given by an advanced automation option (AAO) for different requests, as follows:

Request	AAO
Online, Offline and ExcludeNode	INGREQ_WAIT
CancelRequest	INGSET_WAIT
ResetFromNRE	INGLIST_WAIT

If the appropriate common global variable is not set, 30 seconds is assumed as minimum value.

The calculated time difference is evaluated directly before the requested command is to be issued.

If, for example, the original timeout for the end-to-end automation request is 2500 seconds, and the remaining time directly before issuing the SA z/OS command from the primary agent is still 2000 seconds, it will be shortened to 999 as the maximum value. If the value of 999 seconds is too short for command execution, it can happen that the command times out, even though the end-to-end automation timeout value is very large.

For example, if the calculated time difference directly before issuing an Online request is 35 seconds, and thus less than the minimum value of 40 seconds that is provided in the common global variable `INGREQ_WAIT`, the requested command will be rejected, and an error message issued to say that the task execution request timed out.

Overwriting commands

The SA z/OS commands that result from [Table 12 on page 97](#) can be overwritten in the authorization user exit, as described in sample exit `AOFEXE2E`. But keep in mind that the overwritten command is no longer validated by SA z/OS. Thus you must ensure that the overwritten command can be executed without errors.

Request priority

By default, incoming requests from the end-to-end automation manager are executed with a low priority and they are sorted behind requests from OPERATOR and AUTOOPS in the hierarchy of all possible requests.

However the default value of the priority parameter for `INGREQ` commands from the end-to-end automation manager can be changed via advanced automation options by changing the value of the global variable `INGREQ_PRI.E2EMGR`.

For further details about the hierarchy of requests, see the chapter "Concepts" in *IBM System Automation for z/OS User's Guide*.

Disregarding the hold flag

Because requests from the end-to-end automation manager are handled like those from SA z/OS operators, they cause the *hold* flag for a resource to be overruled.

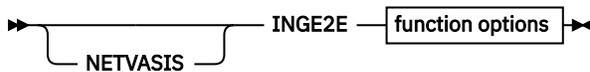
This means that if a resource reference has a desired state of available, the end-to-end automation manager issues a start request for it after the remote SA z/OS resource is started (after an automation agent restart), even though the NOSTART option is specified for the resource.

INGE2E

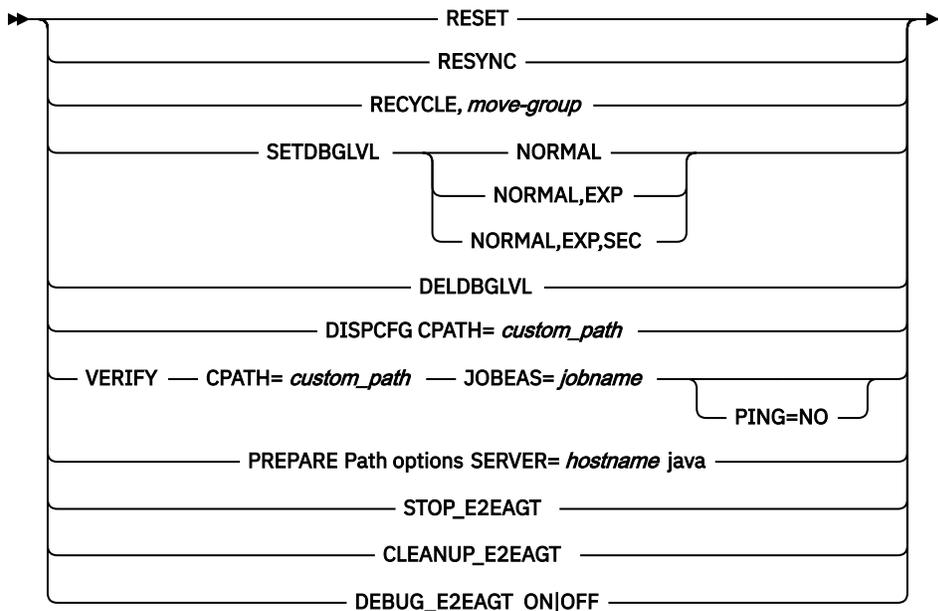
Purpose

The INGE2E command provides the operational API for the end-to-end automation adapter and the end-to-end agent.

Syntax



function options



Path options



java



Parameters

RESET

Specifies cleaning up end-to-end automation adapter-specific settings in the primary automation agent.

This command is used in SA automation policy in case of normal (IEF404I) or abnormal (IEF450I) termination of E2E automation adapter address space. It's normally not issued manually by operators.

RESYNC

Specifies resynchronizing event notification of the E2E agent or SMU server with the first-level automation. For this purpose, SA first notifies them that the automation domain is offline. Afterwards, an online notification is sent. This causes the E2E agent or SMU server to request initial status information for all resources known by it in that automation domain.

This command is used by SA, for example if PAM hot-start takes longer than the value specified in the AOF_E2E_TKOVN_TIMEOUT variable (default is 2 minutes). It's normally not issued manually by operators.

RECYCLE

Specifies recycling a MOVE group. The MOVE group is part of the end-to-end automation high availability policy as described in Appendix B, “High Availability of the End-to-End Automation Adapter,” on page 81. The MOVE group includes the end-to-end automation adapters on the different systems within a subplex. The INGE2E command is specified in REFRESHSTART of the entry type E2EADPT in the STARTUP policy. It recycles the MOVE group either by recycling the primary automation agent, or by a refresh or an ACF COLD of the configuration file.

move-group

The name of the MOVE group to be recycled.

Example to recycle the MOVE group E2EADPT_X/APG, specify:

```
INGE2E RECYCLE E2EADPT_X/APG
```

SETDBGVLV

Use this function only if IBM Support asks you to do so. Set ON debug level for the end-to-end automation adapter REXX backend.

NORMAL

The standard debug level.

EXP

Specifies to debug expiration checking.

SEC

Specifies to debug security user exit.

DELDBGVLV

Use this function only if IBM support asks you to do so. Deletes all debug levels for the end-to-end automation adapter REXX backend.

VERIFY

Verifies the installation of the end-to-end automation adapter. Requires that you have copied and customized the end-to-end automation adapter configuration files.

Requires that the E/AS is already running.

Requires that the operator that calls function VERIFY has rights to access OMVS and to access files and subdirectories within CPATH.

JOBEAS

Specifies the jobname of the E/AS address space.

CPATH

Specifies the customization path of the end-to-end automation adapter. This is the directory that contains the /config subdirectory. For example, local/ing/adapter.

PING

Specifies whether a ping should be performed for the hostnames specified in the adapter configuration file.

Specify NO if you do not want to send a ping to the hostnames.

If you omit this parameter or you set it to anything else than the NO function, VERIFY checks if the ping to the hostnames works correctly.

This function verifies:

- GRPID in INGXINIT must match GRPID in USS file `ing.adapter.plugin.properties`.
- `ServerLocation` in E/AS message adapter configuration file must match `elf-recv-from-hostname` in USS file `ing.adapter.properties`.
- `ServerPort` in E/AS configuration file must match `elf-recv-from-hostname` in USS file `ing.adapter.properties`.
- PPI parameter in E/AS start procedure must match INGEVE2E or CGlobal AOF_E2E_EAS_PPI.
- PPI=YES must be set in INGXINIT.
- E2E_CUSTOM_ROOT in USS file `ingadapter.properties` must match value of CPATH.
- E/AS is not running.
- Performs PING to `elf-send-to-hostname` in USS file `ing.adapter.properties`.
- Performs PING to `elf-send-to-agent-hostname` in USS file `ing.adapter.properties`.

If verification was not successful it writes information text to the console that describes the parameter mismatch in more detail.

Example:

```
NETVAVIS INGE2E VERIFY JOBEAS=EVE2E CPATH=/local/ing/adapter
```

DISPCFG

Reads end-to-end automation adapter configuration files from the custom-root directory CPATH and displays them all one after another on the NetView WINDOW.

Example:

```
NETVAVIS INGE2E DISPCFG CPATH=/local/ing/adapter
```

PREPARE

This function prepares the USS configuration files for the end-to-end automation adapter and completes the following:

1. creates necessary subdirectories within the custom-root directory,
2. copies all necessary configuration files from the install-root to the custom-root,
3. updates the configuration files with the input parameters.

Note: If the directory structure already exists then the command INGE2E PREPARE stops with an error message in order to prevent overwrite an existing configuration.

IPATH

Specifies the install path of the end-to-end automation adapter. The default install path is `/usr/lpp/ing/adapter`.

CPATH

Specifies the customization path of the end-to-end automation adapter. The function PREPARE or the end-to-end automation adapter writes into this directory.

All subdirectories are automatically created.

This is the subdirectory that for example contains the subdirectory `/config` or `/data`.

For example, `/local/ing/adapter`.

SERVER

Specify the SMU server hostname.

JAVA

Optionally specify the 32-bit java home directory, for example, `/local/java/a60/j6.0`

You may omit this parameter if the default java runtime is already 32-bit java version 1.6.

Example:

```
NETVASIS INGE2E PREPARE IPATH=/user/lpp/ing/adapter
CPATH=/local/ing/adapter
SERVER=lnxe2e4x JAVA=/local/java/a60/J6.0
```

STOP_E2EAGT

Stops the E2E agent in a normal manner. This command is used in the SA automation policy. It's normally not issued manually by operators.

CLEANUP_E2EAGT

Cleans up the E2E agent when the E2E agent is not terminated normally, for example, after the E2E agent is canceled. It deregisters the E2E agent from the automation manager and cleans up internal data such as CGlobals. This command is used in the SA automation policy. It's normally not issued manually by operators.

DEBUG_E2EAGT

Specifies whether debugging the E2E agent is on or off.

ON

Switches on debugging the E2E agent for the REXX trace and for the Java logger.

OFF

Switches off debugging the E2E agent.

Example

To recycle the MOVE group E2EADPT_X/APG, specify:

```
INGE2E RECYCLE,E2EADPT_X/APG
```

Example verification output

When you issue on the NetView console command

```
NETVASIS INGE2E VERIFY JOBEAS=neteve2z CPATH=/local/ing/adapter
```

the following output may be viewed:

```
01 SA agent GRPID.....: 34
02 SA agent PPI.....: YES
03 E/AS message adapter jobname.....: NETEVE2Z active
04 E/AS message adapter PORT.....: 5529
05 E/AS message adapter IP.....: 127.0.0.1
06 E/AS message adapter PPI.....: INGEVE2E
07 E2E adapter PPI.....: INGEVE2E
08 E2E adapter EIF PORT.....: 5529
09 E2E adapter EIF IP.....: 127.0.0.1
10 E2E adapter GRPID.....: 34
11 E2E adapter automation domain.....: KEYAPLEX INGXSG34
12 E2E adapter automation operators.....: all active
13 E2E adapter custom root directory.....: /local/ing/adapter
14 E2E adapter install root directory.....: /usr/lpp/ing.adapter
15 E2E adapter java home directory.....: /usr/lpp/java/a80/J8.0
16 E2E SMU server location.....: lnxe2e4x : 2002
17 E2E SMU server location ping.....: successful
18 E2E agent location.....: boekeya : 2003
19 E2E agent location ping.....: successful

20 Verification was successful
```

Explanation of the output

```
01 Parameter GRPID extracted from currently used member INGXINIT.
   See LISTA DSIPARM,INGXINIT
02 PARAMETER PPI extracted from currently used member INGXINIT.
   See LISTA DSIPARM,INGXINIT
03 Jobname of the E/AS address space which must be active.
04 Parameter ServerPort of the E/AS message adapter configuration file.
   Will be extracted from running system via command F JOBEAS,settings, task=messagea
```

05 Parameter ServerLocation of the E/AS message adapter configuration file.
 Will be extracted from running system via command F jobeas,settings,task-messagea

06 Parameters PPI from E/AS initialization configuration file.
 Will be extracted from running system via command F jobeas,display, status

07 PPI queue name used to send events to the end-to-end automation adapter via E/AS.
 Default name is INGEV2E or extracted from the following AAO: AOF_E2E_EAS_PPI

08 Parameter eif-receive-from-port extracted from USS file
 ing.adapter.properties

09 Parameter eif-receive-from-host-hostname extracted from USS file
 ing.adapter.properties

10 Parameter GRPID extracted from USS file

11 Parameter plugin-domain-name extracted from USS file
 ing.adapter.plugin.properties

12 Automation operators E2EOPER and E2EOPRnn must be defined and all active.
 At least 4 must be active or if more defined
 all must be active. See command DISPAOPS and see
 also entry E2EADPT_AUTOOPS in *E2E best practices policy.

13 Parameter E2E_CUSTOM_ROOT extracted from ingadapter.properties

14 Parameter E2E_INSTALL_ROOT extracted from ingadapter.properties

15 Parameter E2E_JAVA_HOME extracted from ingadapter.properties.
 If not specified then the string "default" is shown

16 Parameter eif-send-to-hostname extracted from USS file
 ing.adapter.properties

17 A NetView PING command is used to check if the TCP/IP connection to the SMU server
 host can be established. A firewall might cause an error. You may specify
 parameter PING=NO to avoid checking the ping.

18 Parameter eif-send-to-agent-port extracted from USS file
 ing.adapter.properties

19 A NetView PING command is used to check if the TCP/IP connection to the end-to-end
 agent can be established. A firewall might cause an error. You may specify parameter
 PING=NO to avoid checking the ping.

20 If all checks were successful, you will get this message.

Status mapping

This section gives details of the mapping of statuses between the SA z/OS automation adapter and SMU server.

Introduction

SMU maintains the following status information for each automated resource:

- Observed State
- Desired State
- Automation State
- Startable State
- Compound Status
- Health State

This must be mapped to the following end-to-end automation resource data model status information:

- Observed State
- Desired State
- Operational Status
- Compound Status

The following rule is applied to this mapping:

- The mapping between the values of the automation manager status fields and the status fields of the end-to-end resource data model is only based on the actual status values of the resource. The status history of any of the values does not influence the resulting status values.

Status mapping steps

Procedure

The mapping is carried out in three steps:

1. The SA z/OS statuses observed state, desired state and automation state are mapped to the observed state, desired state and operational state of the end-to-end automation resource data model.

See “Step 1” on page 105

Note: There is a direct one-to-one mapping of the SA z/OS desired state to the end-to-end automation desired state.

2. If, as a result of the first step, the operational status is OK, the second step is used to degrade the operational state, if the SA z/OS compound state reports a problem. See “Step 2” on page 106.
3. The compound state of the end-to-end resource data model is derived from the operational state. See “Step 3” on page 107.

Step 1

Procedure

The following sub-steps are made:

1. In this step, the health status is not directly mapped to the end-to-end automation resource data model. It is, however, implicitly mapped through the compound status (see “Step 2” on page 106).
2. The SA z/OS desired status values Available and Unavailable correspond to the values Available and Unavailable of the desired state displayed in SMU. Other desired state values of the end-to-end automation are ignored.

SA z/OS desired status

Available

⇒ Available

Unavailable

⇒ Unavailable

SMU desired state

3. An automation status of Unknown is mapped to the operational status Unknown, regardless of the other statuses.

SA z/OS automation state

Unknown

⇒ Unknown

SMU operational status

4. Table 13 on page 105 shows the SA z/OS automation manager status mapping. Note that the SA z/OS desired state is mapped directly to the SMU desired state (which is given in the table).

Table 13. SA z/OS automation manager status mapping					
SA z/OS automation manager status				SMU resource data model status	
Observed state	Desired state	Automation state	⇒	Observed state	Operational state
Available	*	Problem	⇒	Available	Error
Available	*	*	⇒	Available	Ok
WasAvail	*	*	⇒	Available	LostCommunication
SoftDown	*	Problem	⇒	Unavailable	Error
SoftDown	*	*	⇒	Unavailable	Ok
SoftDown	Available	Busy	⇒	Starting	InProgress
SoftDown	UnAvailable	Busy	⇒	Stopping	InProgress

Table 13. SA z/OS automation manager status mapping (continued)

SA z/OS automation manager status				SMU resource data model status	
Observed state	Desired state	Automation state	⇒	Observed state	Operational state
HardDown	*	*	⇒	Unavailable	For APLs: NonRecoverableError For other resource types: Error
Standby	*	Problem	⇒	Unavailable	Error
Standby	*	*	⇒	Unavailable	Dormant
Starting	*	Problem	⇒	Starting	Error
Starting	*	*	⇒	Starting	InProgress
Degraded	*	*	⇒	Available	Degraded
Degraded	Unavailable	Compound=Awaiting & type=APG	⇒	Stopping	Degraded
Stopping	*	Problem	⇒	Stopping	Error
Stopping	*	*	⇒	Stopping	InProgress
Problem	Available	*	⇒	Starting	Error
Problem	Unavailable	*	⇒	Stopping	Error
SysGone	*	*	⇒	Unavailable	SupportingEntityInError
Unknown	*	*	⇒	Unknown	NoContact

* indicates that this can be any valid value.

Step 2

The second step is used to degrade the operational state if the SA z/OS compound state reports a problem. The degradation is done in combination with the desired and observed status according to [Table 14 on page 106](#).

Table 14. Status mapping: second step if the SA z/OS compound status reports a problem

SA z/OS compound status	SMU desired status	SMU observed status	⇒	Operational state
Degraded	*	*	⇒	Degraded
Awaiting	Available	Unavailable	⇒	StartReqPending
Awaiting	Unavailable	Available	⇒	StopReqPending
Denied	Available	Unavailable	⇒	StartInhibited
Denied	Unavailable	Available	⇒	StopInhibited
Inhibited	Available	Unavailable Stopping	⇒	StartInhibited
Inhibited	Unavailable	Available Starting	⇒	StopInhibited
Inhibited	Available	Available	⇒	StopInhibited
Inhibited	Unavailable	Unavailable	⇒	StartInhibited
Problem	*	*	⇒	Error

* indicates that this can be any valid value.

For example, the operational state will be degraded in this second step, if the health status is degraded, which impacts the compound status of SA z/OS. Another reason for a degraded compound status of SA z/OS might be an automation flag that is switched off.

Step 3

The compound status of the end-to-end resource data model is derived from the operational state according to the following table:

SMU operational state		SMU compound status
Unknown	⇒	Warning
Ok	⇒	Ok
Degraded	⇒	Warning
Error	⇒	Error
NonRecoverableError	⇒	Fatal
NoContact	⇒	Warning
LostCommunication	⇒	Warning
Dormant	⇒	Ok
SupportingEntityInError	⇒	Error
StartInhibited	⇒	Warning
StopInhibited	⇒	Warning
StartReqPending	⇒	Warning
StopReqPending	⇒	Warning
InProgress	⇒	Warning

Appendix E. Messages

This information lists the messages that are issued either by the automation adapter or the automation adapter plug-in.

Note that within NetView an additional * may be appended to the end of the message text for EEZNnnnn, INGX97nn, INGX98nn, and INGX99nn messages.

End-to-end automation adapter messages

The following messages are issued by the automation adapter.

EEZA0001E **Syntax error on line *line number***

System action

The automation adapter stops.

Explanation

A syntax error has occurred in the configuration file, for example, a leading = on a line.

Operator response

Supply a value for the key in the configuration file.

System action

The automation adapter stops.

EEZA0004E **Integer out of bounds in key "*the key*". Expected value between *the lower bound expected* and *the upper bound expected*, found the value *parsed***

Operator response

Analyze the configuration file for invalid syntax.

Explanation

The system expected an integer value between the given bounds (inclusive) for the given key, but found a value outside these bounds.

EEZA0002E **Wrong datatype in key *the key*. Expected *the desired type*, found value "*the value that was found*"**

Explanation

The value of the given key cannot be interpreted as the desired type. For example, the system expected a boolean value but found the string "hello".

System action

The automation adapter stops.

System action

The automation adapter stops.

Operator response

Supply a value within the given bounds for the key.

Operator response

Analyze the configuration file for invalid key/value pairs.

EEZA0005I **At least one system symbol cannot be resolved: *text-line***

Explanation:

The text line in error contains the system symbol that cannot be resolved. A system symbol is considered unresolved if it is either not defined or empty.

EEZA0003E **The key "*the key that was not found*" was not found and no default value was given**

System action:

The automation adapter stops.

Explanation

The system wanted to retrieve from the configuration file a value that did not exist and no default value was given.

Operator response:

If available, refer to message EEZA0031E that specifies the name of the configuration file in error. Check the text line in error for an invalid system symbol. The system symbol in the corresponding z/OS IEASYMxx parmlib member might not be defined. If you are authorized to do so, enter the z/OS DISPLAY SYMBOLS command to display the system symbols

and associated substitution texts that are in effect, and then define the corresponding system symbol if it's missing.

EEZA0006E **Cannot create an instance of the class because class not found: *class name***

Explanation

The automation adapter cannot load the class.

System action

The automation adapter rejects the request.

Operator response

Check whether the class name is valid and is available in the corresponding classpath.

EEZA0007E **Cannot create an instance of the class because method not found: *class name***

Explanation

The automation adapter can load the class but cannot create an instance.

System action

The automation adapter rejects the request.

Operator response

Check whether the class is valid.

EEZA0008E **Cannot create an instance of the class because of an unknown error: *class name***

Explanation

The automation adapter cannot load the class or create an instance.

System action

The automation adapter rejects the request.

Operator response

Check whether the class is valid and analyze the attached original exception.

EEZA0009E **Invocation of adapter plug-in failed: *plug-in=plug-in name, method=method name, internalRetcode=internal return***

code, taskRetcode=task return code

Explanation

The automation adapter client API was called to execute a task on the remote automation adapter. There are three error categories:

- The client suffers an error on the connection
- The execution of the task within the automation adapter backend failed
- Execution failed in the automation adapter plug-in.

Automation adapter internal return codes:

- 3** Backend detects that the request expired before it was started.
- 4** Backend cannot be loaded.
- 22** Backend cannot load the plug-in Java class.
- 23** An exception occurred in the backend or the plug-in returns a non-zero task return code.
- 24** Backend rejects the execution of the task. See the automation adapter trace for details.
- 30** Backend detects invalid input arguments. Cannot execute task. This is an internal error.
- 40** Authentication failed. The user ID or password has not been specified or an internal error occurred.
- 41** Authentication failed. The user ID or password is invalid on the automation adapter host.
- 42** Authentication failed. The user ID has been revoked. JAAS login modules may not provide this information or may also use it if the password has expired.
- 43** Authentication failed. The password has expired. JAAS login modules may not provide this information.
The automation adapter client connection internal return codes:
- 101** Connection timed out while open.
- 102** Connection failed during open/read/write because peer closed.

- 103**
Not all data received. The data was read successfully but the number of bytes is too small. This might be an internal error.
- 104**
Unable to connect to the automation adapter.
- 105**
Unable to close connection. This might be an internal error.
- 106**
Unable to send data. Connection has been aborted due to, for example, SSL mismatch or the automation adapter stopped.
- 107**
Marshal error occurred while sending or receiving data. This is an internal error.
- 108**
Connection is invalid, for example, connection has not been opened or previously failed.
- 109**
Read response failed. Data is zero, for example, the connection has been aborted due to an SSL mismatch.
- 110**
Connection timed out during read/write.
Predefined task return codes from plug-in:
- <0**
Internally used task return codes have been mapped to one of the automation adapter return codes.
- 0**
The plug-in was successful.
- 4004**
The plug-in connection timed out.
- 4008**
The plug-in authorization is missing to execute the request.
- 4009**
The plug-in authentication failed.
- 4012**
The plug-in failed but might execute the next request successfully. This is the default return code if the plug-in throws an exception but the plug-in return code is zero.
- 4016**
The plug-in failed and might not execute the next request successfully.
- 4020**
The plug-in failed and forces the entire automation adapter to stop.

System action

Execution of the remote task fails.

Operator response

Analyze the return code description. If it is an internal error, check IBM Electronic Support for additional information - <http://www.ibm.com/support/entry/portal/>

EEZA0010E **Request expired before the adapter passes it to the adapter plug-in. Timeout period is *timeout value* seconds**

Explanation

All requests have an associated expiration date. The request is scheduled to an execution thread that detected that the expiration time had expired.

System action

The automation adapter rejects the request.

Operator response

Analyze the reason (for example, high working load). Increase the timeout period if necessary.

EEZA0011E **The backend program specification is invalid**

Explanation

The backend program is not a Java program or the Java program name was not specified.

System action

The automation adapter rejects the request.

Operator response

Check the program that called the automation adapter client API.

EEZA0012E **Invalid parameter list**

Explanation

The automation adapter detected a request that is associated with an invalid parameter list.

System action

The automation adapter rejects the request.

Operator response

Check the program that called the automation adapter client API.

EEZA0013E **Authentication for user ID *user name* was unsuccessful**

Explanation

The request is associated with a user ID and password that have been validated unsuccessfully.

System action

The automation adapter rejects the request.

Operator response

Check whether the user ID is authorized for the system and check the security policy. Also check if you have stored a user ID and password for this domain in the credential store of the Dashboard Application Services hub.

EEZA0014E **The original exception *original-class* needs to be transported to the remote caller**

Explanation

An exception from an underlying component needs to be transported to the remote caller.

System action

None.

Operator response

Analyze the original exception attached with this message.

EEZA0015E **Method not supported: *name of the missing method***

Explanation

The automation adapter detected an unknown method name. The list of all valid method names is defined in the EEZAdapterInteraction interface.

System action

The automation adapter rejects the request.

Operator response

Check IBM Electronic Support for additional information - <http://www.ibm.com/support/entry/portal/>

EEZA0017E **Request not supported: *name of the unsupported request***

Explanation

The automation adapter plug-in does not support the specified request.

System action

The request might be rejected depending on the behavior of the plug-in.

Operator response

Check whether the automation domain supports this type of request. Check whether you have installed the latest E2E automation adapter version.

EEZA0022E **Adapter client is unable to connect to the adapter at *host:port* due to exception: *the exception that was caught***

Explanation

The automation adapter client cannot connect to the server at the given host and port. The original exception text is provided.

System action

The connection is not established.

Operator response

Analyze the original exception. For example, check firewall settings.

EEZA0023E **Cache directory is invalid**

Explanation

The EIF cache directory is not a directory.

System action

The automation adapter stops.

Operator response

Correct the configuration file.

EEZA0024E **EIF sender and receiver must not be equal**

Explanation

The EIF configuration parameters are not allowed to point to each other.

System action

The automation adapter stops.

Operator response

Correct the master configuration file. For example, compare parameter *eif-receive-send-hostname* with *eif-receive-from-hostname* and compare parameter *eif-send-from-port* with *eif-receive-from-port*.

EEZA0025E **Cannot find the plug-in configuration file: *configuration file name***

Explanation

The master configuration file contains the name of a plug-in configuration file that cannot be found.

System action

The automation adapter stops.

Operator response

Correct the configuration file. Check parameter *plugin-configfile-xxx*, where for example *xxx* stands for *sa4zos*.

EEZA0026E **No plug-in configuration file was specified**

Explanation

The master configuration file must contain at least one plug-in configuration file.

System action

The automation adapter stops.

Operator response

Correct the configuration file. Compare parameter *plugin-impl-class* with the IBM provided settings.

EEZA0027E **Cannot load configuration file: *configuration file name***

Explanation

The specified configuration file cannot be loaded.

System action

The automation adapter stops.

Operator response

Make sure that the configuration file resides in your customized E2E automation adapter configuration directory.

EEZA0028E **Plug-in configuration file does not contain all mandatory parameters: *configuration file name***

Explanation

The specified configuration file does not contain all mandatory parameters. The plug-in is not used.

System action

The automation adapter does not deploy the plug-in.

Operator response

Compare the configuration file with the IBM provided settings.

EEZA0029E **Cannot create the first instance of the plug-in class: *class name***

Explanation

An attempt was made to create the first instance of the plug-in during initialization. Creation failed.

System action

The automation adapter does not deploy the plug-in.

Operator response

Compare parameter *plugin-impl-class* with the IBM provided settings.

EEZA0030E **Cannot set up event subscription list for plug-in configuration file: *plug-in configuration file name***

Explanation

The specification of the EIF event classes in the plug-in configuration file is invalid.

System action

The automation adapter does not deploy the plug-in.

Operator response

Compare parameter *plugin-event-classes* with the IBM provided settings.

EEZA0031E **Cannot load configuration file from: *plug-in configuration file name***

Explanation

The automation adapter cannot load the specified configuration file because either no configuration file or an invalid one was specified.

System action

The automation adapter stops.

Operator response

Check whether the name of the configuration file is correct. Compare parameter *plugin-configfile-xxx* with the IBM provided settings.

EEZA0032E **Initialization of the adapter failed: *original exception***

Explanation

An error occurred in the initialization step of the automation adapter.

System action

The automation adapter stops.

Operator response

Analyze the associated exception. If there is no exception text for this message, try to find additional messages in the automation adapter log.

EEZA0033E **Unable to create *type of factory* SocketFactory**

Explanation

The automation adapter server or client cannot create a socket factory for the remote contact.

System action

The automation adapter client cannot create a connection or the automation adapter server cannot receive connections.

Operator response

Analyze the reason using previous messages.

EEZA0036E **The adapter suffered an unexpected interruption: *original exception***

Explanation

The automation adapter waits for a termination command. An unexpected interruption occurred.

System action

The automation adapter stops.

Operator response

Analyze the original exception.

EEZA0037E **The adapter stops because no plug-in has been successfully initialized**

Explanation

At least one plug-in must have been successfully initialized otherwise the automation adapter stops.

System action

The automation adapter stops.

Operator response

Analyze previous messages and exceptions issued by the failing plug-in.

EEZA0038E **A (SSL) socket configuration error occurred: *exception text***

Explanation

An error occurred during the loading or processing of (SSL) socket-related configuration data. An SSL handshake exception will only be reported during initial contact.

System action

The automation adapter client cannot create a connection or the automation adapter server cannot receive connections.

Operator response

Analyze the exceptions text. Check the SSL configuration file, *ing.adapter.ssl.properties*, if necessary.

EEZA0039E **Not all data was read from socket: number of bytes read bytes read, number of bytes expected bytes expected to be read**

Explanation

The incoming request has a length in bytes, but not all bytes can be read.

System action

The automation adapter rejects the request.

Operator response

Check why the socket connection was broken while transferring data.

EEZA0040E **The adapter client cannot establish connection to the adapter: string representation of the connection**

Explanation

Opening the connection failed. A request cannot be sent to the automation adapter. The string representation of the connection contains details about the connection.

System action

The automation adapter frontend failed.

Operator response

Analyze the connection information.

EEZA0041E **The adapter client cannot invoke an adapter request: InternalRC=internal return code, TaskRC=task return code**

Explanation

A connection to the automation adapter has been successfully established. The automation adapter frontend might have sent a request to the automation adapter but the request failed. If the internal or task return codes are not applicable (n/a), some other unexpected exception occurred.

System action

The automation adapter frontend failed.

Operator response

Analyze the internal and task return codes (see EEZA0009E for an explanation of the return codes).

EEZA0042E **The adapter has thrown a remote exception: InternalRC=internal return code, TaskRC=task return code. The original message was: message text**

Explanation

A connection to the automation adapter has been successfully established. The automation adapter frontend has sent a request to the automation adapter but the plug-in has thrown an exception.

System action

None.

Operator response

Analyze the internal and task return codes (see EEZA0009E for an explanation of the return codes).

EEZA0043E **A required command line parameter is missing**

Explanation

One of the required command line parameters is missing (such as -start,-stop or -terminate).

System action

The automation adapter frontend failed.

Operator response

Specify the required command-line parameters and try again.

EEZA0045E **The adapter cannot establish a server socket due to illegal arguments: exception text**

Explanation

The automation adapter cannot establish a receiver thread and cannot accept incoming connections.

System action

The automation adapter stops.

Operator response

Analyze the configuration file for an invalid IP address.

EEZA0047E **The adapter is unable to accept connections due to socket exception "exception"**

Explanation

An exception occurred as the automation adapter was about to accept an incoming connection.

System action

The automation adapter stops.

Operator response

Analyze the exception text.

EEZA0051W **Termination of the adapter failed due to exception: *error message***

Explanation

The attempt to stop the receiver thread failed because an exception occurred.

System action

None.

Operator response

Analyze the exception text.

EEZA0052E **Cannot create an in-storage EIF configuration file: *exception text***

Explanation

An instance of the Java class ByteArrayInputStream cannot be created or written.

System action

The automation adapter stops.

Operator response

This is probably an internal error. The exception text might give the reason for the problem.

EEZA0053E **Missing argument for command line parameter "*the parameter*"**

Explanation

A required argument for a command line parameter (such as `-start`) is missing. For example, `AdapterCmd -start` would be incorrect because `-start` requires an argument. A

correct example would be: `AdapterCmd -start com.ibm.ing.saplugin.INGXPluginInvocation`

System action

Processing of this command ends.

Operator response

Check the documentation for information about valid command line arguments and their parameters.

EEZA0055E **Remote Contact inactivity threshold exceeded: elapsed seconds=*elapsed seconds* threshold=*threshold***

Explanation

The automation adapter calculates the elapsed time since the last synchronous request was received. The automation adapter stops itself if this time exceeds the number specified in the parameter `eez-remote-contact-activity-interval-seconds`. Any incoming event is used as a trigger for the calculation.

System action

The automation adapter stops.

Operator response

You might want to increase the number of seconds specified by the parameter `eez-remote-contact-activity-interval-seconds`. Setting this parameter to 0 (zero) means it never expires.

EEZA0056I **Initial contact was enabled and the connection to the management server has been established**

Explanation

The parameter `eez-initial-contact` was set to true and the automation adapter attempted to connect the management server. The handshake to the management server was successful.

System action

None.

Operator response

No action required.

EEZA0057E **The connection to the management server cannot be established**

Explanation

The automation adapter stops attempting to connect to the management server because the timeout interval is over.

System action

The automation adapter stops.

Operator response

You might want to increase the number of minutes specified by the parameter `eez-initial-contact-retry-interval-minutes`. Specify the value 0 (zero) in order to retry forever.

EEZA0058E **The plug-in has not been deployed or not yet started: *name of the Java plug-in class***

Explanation

An attempt was made to issue a request against a non-deployed plug-in or a plug-in that has not been started.

System action

The automation adapter rejects the request.

Operator response

Check the plug-in configuration file and deploy the missing plug-in class. Search for message EEZA0115I.

EEZA0059E **An internal error occurred**

Explanation

The automation adapter detected an internal error.

System action

None.

Operator response

Check IBM Electronic Support for additional information - <http://www.ibm.com/support/entry/portal/>

EEZA0060I **The termination of the adapter is delayed for *duration of the delay in seconds* seconds**

Explanation

Stopping the automation adapter is delayed for a short while until it has sent the appropriate domain leave

events. You can configure the duration of this delay with the `eez-stop-delay-seconds` parameter.

System action

The automation adapter attempts to send domain leave events.

Operator response

No action required.

EEZA0061E **Unable to bind a socket to address *eez-remote-contact-hostname* at port *eez-remote-contact-port*. Reason: *message of the exception***

Explanation

The automation adapter was unable to use this address or port. Possible causes of the problem are:

- The port is already in use by another program.
- The address could not be assigned.

System action

The automation adapter stops.

Operator response

Make sure that no program is using this port (that is, an automation adapter that is already running). If another program needs this port, then configure the automation adapter to use another port (with the `eez-remote-contact-port` parameter in the master configuration file). Ensure that the address is valid.

EEZA0062I **The start command of the automation plug-in *name of the Java plug-in class* was successful**

Explanation

The selected automation plug-in was successfully started.

System action

The automation adapter has started the automation plug-in.

Operator response

No action required.

EEZA0063I **The stop command of the automation plug-in *name of the Java plug-in class* was successful**

Explanation

The selected automation plug-in was successfully stopped.

System action

The automation adapter has stopped the automation plug-in.

Operator response

No action required.

EEZA0064I **The termination command for the adapter was successful**

Explanation

The automation adapter was successfully stopped.

System action

The automation adapter stops.

Operator response

No action required.

EEZA0070E **The host name *eez-remote-contact-hostname* is unknown**

Explanation

The automation adapter was unable to resolve the host name.

System action

The automation adapter stops.

Operator response

Specify a host name in parameter *eez-remote-contact-hostname*.

EEZA0071E **The domain name is either null or empty**

Explanation

The plug-in returned an invalid domain name since it is either null or empty.

System action

The plug-in cannot be started.

Operator response

Specify a domain name in the plug-in configuration file. Use parameter *plugin-domain-name* or comment it out and use the default.

EEZA0100I **The adapter has been started**

Explanation

This is the first of a sequence of three messages until the automation adapter is ready. The automation adapter starts initialization and will try to connect to the management server if *eez-initial-contact=true*.

System action

None.

Operator response

No action required.

EEZA0101I **The adapter is active**

Explanation

The automation adapter becomes "active" after a connection has been successfully established to the management server. The automation adapter continues initialization, finds and starts up all plug-ins.

System action

None.

Operator response

No action required.

EEZA0102I **The adapter is ready**

Explanation

The automation adapter startup sequence is complete.

System action

None.

Operator response

No action required.

EEZA0103I **The adapter is stopping**

Explanation

An internal or an external stop command has been received.

System action

The automation adapter is about to stop.

Operator response

No action required.

EEZA0104I The adapter has been stopped

Explanation

The automation adapter termination is complete. All possible stop delay periods are over. The process stops immediately.

System action

The automation adapter has stopped.

Operator response

No action required.

EEZA0105I The adapter has been stopped due to a failure, rc=return code

Explanation

The automation adapter stopped because an error occurred. All possible stop delay periods are over. The process stops immediately. The return code might be:

- 12** if initial contact failed
- 13** if the remote contact activity threshold is exceeded
- 16** if a plug-in forced termination of the automation adapter
- 20** if initialization failed
- 24** if an error occurred after initialization was successful
- 28** if an unsupported environment is detected

System action

The automation adapter stops.

Operator response

Search for error messages that were issued previously. On z/OS return code 28 might be caused by the 64-bit JVM. You should use the 32-bit JVM instead.

EEZA0111I The plug-in is starting: name of the Java plug-in class

Explanation

The automation adapter has already successfully created an instance of the plug-in class and will now call function INIT_DOMAIN.

System action

None.

Operator response

No action required.

EEZA0112I The plug-in has been started: name of the Java plug-in class

Explanation

The automation adapter plug-in has successfully initialized the domain (INIT_DOMAIN).

System action

None.

Operator response

No action required.

EEZA0113I The plug-in is stopping: name of the Java plug-in class

Explanation

The automation adapter will call plug-in function TERM_DOMAIN.

System action

None.

Operator response

No action required.

EEZA0114I The plug-in has been stopped: name of the Java plug-in class

Explanation

The automation adapter plug-in has successfully stopped the domain (TERM_DOMAIN).

System action

None.

Operator response

No action required.

EEZA0115I **The plug-in startup failed: *name of the Java plug-in class***

Explanation

This message might follow after EEZA0111I, but the attempt to start the plug-in via function INIT_DOMAIN failed. The automation adapter plug-in will not be started automatically.

System action

The plug-in will be disabled. A join event was not sent.

Operator response

You might want to restart the plug-in using the automation adapter start command. Analyze further plug-in messages.

EEZA0116I **The status of the event sender changed: *Address=address, Port=port, Status=status***

Explanation

This message occurs if the status of the EIF connection changed. The reason could be that a new EIF connection is created or an existing EIF connection is lost. The reason can be found in the status. A status='connection timed out' is expected if the management server is stopped, for example if the management server moves to another system and therefore the automation adapter needs to change the EIF sender destination. The *status* can be:

- 1 connection created
- 2 connection changed
- 4 connection closed
- 8 connection timed out

System action

None.

Operator response

No action required.

EEZA9991E **The message file is not installed**

Explanation

The English message file must be available.

System action

The automation adapter stops.

Operator response

Make sure that the message file is in the class path

EEZA9992E **EEZAdapterLogger is not available**

Explanation

The automation adapter logging component has not been initialized.

System action

The automation adapter stops. Other processes using the automation adapter client API will be unable to write messages to log and trace files.

Operator response

Check IBM Electronic Support for additional support - <http://www.ibm.com/support/entry/portal/>

EEZC0001I **Setting up Tivoli Common Directory at *location where Tivoli Common Directory is being set up.***

Explanation

The Tivoli Common Directory path was set to its default value, as shown in the message text.

System action

No system action required.

Operator response

No operator action required.

EEZC0002I **Unable to determine Tivoli Common Directory. Diverting**

serviceability related output to alternative location.

Explanation

The system was not able to determine the Tivoli Common Directory.

System action

Processing continues. The application will attempt to divert serviceability related output to another location for this session.

Operator response

In order to manage its serviceability related output, the application should be granted read/write permission to the location /etc/ibm/tivoli/common (UNIX) or <Program_Files_Dir>\ibm\tivoli\common (Windows).

EEZC0003I **Base output directory for serviceability related files (for example, message log files and trace files) has been set to new output directory.**

Explanation

The output directory for serviceability related files was set to its default value, as shown in the message text.

System action

From now on the application will write serviceability related information to the directory that is contained in the message text.

Operator response

No action is required if the base output directory for serviceability related files is acceptable. Otherwise, if it is required to relocate the base output directory, modify the entry in log.properties which should be located at <Program_Files_dir>\ibm\tivoli\common\cfg (Windows) or /etc/ibm/tivoli/common/cfg/log.properties (UNIX). Changes to this file will take effect once the corresponding component is restarted.

EEZC0004I **Changing base output directory for serviceability related files of name of logger from old output directory to new output directory.**

Explanation

Due to changes in configuration settings the output directory of serviceability related files has been relocated.

System action

From now on the system will write serviceability related information to the new location.

Operator response

No action is required if the base output directory for serviceability related files is acceptable. Otherwise, if it is required to relocate the base output directory, modify the entry in log.properties which should be located at <Program_Files_dir>\ibm\tivoli\common\cfg (Windows) or /etc/ibm/tivoli/common/cfg/log.properties (UNIX). Changes to this file will take effect once the corresponding component is restarted.

EEZK0003E **String *someString* is too long: the maximum length of *nameOfTheString* Strings is *maxLength*.**

Explanation

Setting the String to the specified value did not succeed due to string length.

System action

The current task ends.

Operator response

Verify the input parameters. If the problem persists, provide the trace file and this message text to IBM support.

EEZK0004E **String named *someStringName* must not be null and must not exceed the maximum length of *maxLength*.**

Explanation

Setting the String to null is not allowed.

System action

The current task ends.

Operator response

Verify the input parameters. If the problem persists, provide the trace file and this message text to IBM support.

EEZK0005E **An exception, which is not an instance of EEZApplicationException has been passed to the EEZApplicationTransientException . The type of the message is exceptionType. The exception message is: exceptionMessage.**

Explanation

This is an unexpected behavior.

System action

The current task will continue. The exception will be processed.

Operator response

Provide the logs and traces for more details, please, if any error other error occurs.

End-to-end automation adapter plug-in messages

The following messages are issued by the automation adapter plug-in.

ING230E **Addressed consumer *consumer* not registered. Event not forwarded.**

Explanation

An event was received with a consumer who was not registered as the target address. The event cannot therefore be forwarded to the addressed consumer.

The variable *consumer* shows the name of the event consumer who was specified as the target address of the event that the event should have been forwarded to.

System action

Processing of the event is terminated.

Operator response

None.

System programmer response

Check whether event notification has been correctly initialized and end-to-end automation has been registered as an event consumer.

ING231E **event notification service returns *rc*. Event not forwarded to *receiver_id*.**

Explanation

The event could not be forwarded to the addressed consumer via the PIPE stage TECROUTE.

The variable *rc* shows the return code to PIPE stage TECROUTE.

The variable *receiver_id* shows the name of the PPI receiver ID.

System action

Processing of the event is stopped.

Operator response

None.

System programmer response

Check whether the status of the PPI receiver ID is active. The status of PPI receivers can be listed with the command DISPPPI.

ING232I **PPI receiver ID for E2E adapter is set to *receiver_id*.**

Explanation

The PPI receiver ID for the message adapter of the event/automation service, which is used to forward events to the automation adapter, has been set to the specified receiver ID.

System action

None.

Operator response

None.

System programmer response

None.

ING233E **Event forwarding task EVTOPER for E2E not defined.**

Explanation

The primary agent cannot be started because the automated function EVTOPER, which is used to forward the events to the automation adapter, has not been defined in the automation policy.

System action

Processing stops.

Operator response

None.

System programmer response

Define the automated function EVTOPER.

ING234E Request '*request*' failed:
RC=*rc* command='*command*'
reason='*reason*'

Explanation

The specified command abended.

The variable *request* shows the request that was received from the automation adapter for execution.

The variable *rc* shows the return code of the command.

The variable *command* shows the command that implements the request.

The variable *reason* shows text that describes the error.

System action

Execution of the request stops.

Operator response

None.

System programmer response

Analyze the return code and reason code for the abending command.

ING235W No information from automation manager. *event* event not created for *consumer*.

Explanation

An error occurred when requesting information from the automation manager that needed to be included in a domain event. The event has not been created.

- The variable *event* specifies the event to be created. Possible values are: READY, REFRESH, CHANGE.

A READY event is created at initialization time of the automation domain and after having recovered from miscellaneous error situations.

A REFRESH event is created after having refreshed the configuration policy.

A CHANGE event is created after a takeover of the automation manager if the outage has not exceeded a given time interval.

- The variable *consumer* shows the name of the event consumer that the event should have been forwarded to.

System action

Processing of the event is stopped.

Operator response

None.

System programmer response

Analyze the NetView log for further SA z/OS messages concerning the communication between the primary agent and the automation manager.

ING236W Task execution request
request processed with
warnings: COMMAND='*command*'
REASON='*reason*'

Explanation

A warning was issued when executing a task execution request. The message variables are:

request

The name of the task execution request

command

The command that failed.

reason

Text that describes the error.

System action

The request is executed with warnings.

Operator response

None.

System programmer response

Analyze the reason code of the warning message.

ING237W Request '*request*' processed with warnings: **COMMAND='command'**
REASON='reason'

Explanation

A warning was issued when executing an end-to-end request.

The variable *request* shows the name of the end-to-end request.

The variable *command* shows the issued command that implements the request.

The variable *reason* specifies the warning message.

System action

The end-to-end request executes.

Operator response

Analyze the reason code of the warning message.

System programmer response

None.

ING238E Task execution request *tex_request* failed: **ERROR=error_code**
SERVICE='service' **RC=return_code**
REASON='reason'

Explanation

The REXX function INGRXTX0 and common REXX macros issue this message if there is an error. The message variables are:

tex_request

The name of the task execution request

error_code

The REXX function INGRXTX0 and the common REXX macros issue the following error codes:

- 1** RPC service failed: either INGPXSND, INGPXRCV or INGPXDEL.
- 2** Read from or write to the task execution data stream failed.
- 3** Unknown task execution operation name.
- 4** Execution of a task execution request started but failed. The REXX macro that implemented the execution of the request did not issue an error message. This is the default return

code and the standard ING238E message is written with SERVICE=OperationName and RC=OperationReturnCode.

- 5** Authorization error: the user that made the request is not authorized to make it.
 - 6** At least one request in the list of requests failed.
 - 7** The execution of a task execution request failed for any other reason.
 - 8** The execution of a task execution request failed because service INGPYAMC failed.
 - 9** Execution of a task execution request was rejected because the expiration time had elapsed. See also The system programmer response for message ING249E.
 - 10** User command executed but failed because return code is greater than zero.
 - 11** Cannot start the user command because user command not found on target system.
 - 12** Cannot start the user command because command user ID not defined on target system.
 - 13** Cannot start the user command.
 - 24** Task execution request contains no meta data.
 - 28** The task execution command handler aborts because there is not enough time to send back the response data. No response is returned.
 - 32** The task execution command handler aborts because of a REXX signaling error.
 - 36** The automation adapter is not supported by this product.
- service**
The service that failed.
- return_code**
The return code of the service that failed.
- reason**
Text that describes the error.

System action

Execution of the request failed for the reason given in the message.

Operator response

None.

System programmer response

Check the netlog and correct the error.

ING239W **Value in advanced automation option *name* not valid. Default *value* used.**

Explanation

The value provided in the advanced automation option is not accepted as a valid value.

The variable *name* shows the name of the advanced automation option.

The variable *value* shows the value of the advanced automation option.

System action

Processing continues with the default value for the specified advanced automation option.

Operator response

None.

System programmer response

Correct the value provided in the advanced automation option.

ING240E **Unexpected task execution request name *tex_request*.**

Explanation

The REXX macro implemented to execute the task execution request was called with an unexpected task execution request name.

The variable *tex_request* shows the name of the task execution request.

System action

Processing of the task execution request fails.

Operator response

Contact IBM Support.

System programmer response

None.

ING241E **Unexpected task execution element type *type*.**

Explanation

The REXX macro implemented to execute the task execution request was called with an unexpected task execution request type.

The variable *type* shows the type of task execution element.

System action

Processing of the task execution request fails.

Operator response

Contact IBM Support.

System programmer response

None.

ING242E **Cannot read task execution element from data stream.**

Explanation

The REXX macro implemented to execute the task execution request cannot read the task execution element from the data stream.

System action

Processing of the task execution request fails.

Operator response

Contact IBM Support.

System programmer response

None.

ING243E **Cannot write task execution element to data stream.**

Explanation

The REXX macro implemented to execute the task execution request cannot write to the data stream to return data to the calling routine.

System action

Processing of the task execution request fails.

Operator response

Contact IBM Support.

System programmer response

None.

ING244E Invalid filter element name *filter*.

Explanation

The REXX macro implemented to execute the task execution request was passed an invalid filter element name.

The variable *filter* shows the name of the filter element.

System action

Processing of the task execution request fails.

Operator response

Contact IBM Support.

System programmer response

None.

ING245E Duplicate filter element name *filter*.

Explanation

The REXX macro implemented to execute the task execution request was passed duplicate filter element names.

The variable *filter* shows the name of the filter element.

System action

Processing of the task execution request fails.

Operator response

Contact IBM Support.

System programmer response

None.

ING246E Invalid combination of filter element names *filter1* and *filter2*.

Explanation

The REXX macro implemented to execute the task execution request was passed an invalid combination of filter element names.

The variables *filter1* and *filter2* show the names of the two filter elements.

System action

Processing of the task execution request fails.

Operator response

Contact IBM Support.

System programmer response

None.

ING247E Invalid filter element *filter*. Filter value unknown: *value*

Explanation

The REXX macro implemented to execute the task execution request was passed an unknown filter value.

The variable *filter* shows the name of the filter.
The variable *value* shows the value of the filter.

System action

Processing of the task execution request fails.

Operator response

Contact IBM Support.

System programmer response

None.

ING248E Invalid filter element *filter*. Filter operator unknown: *operator*

Explanation

The REXX macro implemented to execute the task execution request was passed a filter element with an unknown operator comparing the filter name and filter value.

The variable *filter* shows the name of the filter.
The variable *operator* shows the filter operator.

System action

Processing of the task execution request fails.

Operator response

Contact IBM Support.

System programmer response

None.

ING249E	Execution of a task execution request timed out. The expiration time <i>time</i> was too short – <i>missing_time</i>.
----------------	--

Explanation

The expiration time that was provided for the task execution request was too short.

The variable *time* shows the expiration time that was provided for the task execution request.

The variable *missing_time* specifies the missing time interval.

System action

Processing of the task execution request fails.

Operator response

Contact your system programmer.

System programmer response

Check the defined timeout interval for the end-to-end automation request. The expiration time of an end-to-end automation request is determined by where it is issued, as follows:

Request issued from:	Expiration time determined by:
The SMU WebUI or System Automation Application Manager	The timeout interval that is defined in the WebSphere Admin Console for the environment variable <i>com.ibm.eez.aab.invocation-timeout-seconds</i>
The automation adapter (such as INIT_DOMAIN)	The timeout interval that is defined in the automation adapter <i>ing.adapter.plugin.properties</i> configuration file
E2E agent	The timeout interval <i>INGAGT_CON_TIMEOUT</i> that is defined in the automation agent <i>inge2eagt.properties</i> configuration file

It is possible that an end-to-end automation request that has been issued might be queued in the

automation adapter or the NetView operator task for a certain period of time. You can therefore increase the timeout interval to take this delay into account.

If the timeout occurs during *INIT_DOMAIN*, you can also increase the *TIMEOUT* parameter in *ing.adapter.plugin.properties*.

ING250E	Execution of a service timed out. The expiration time <i>time</i> was reduced to <i>number</i> seconds.
----------------	--

Explanation

A service routine timed out without exploiting the expiration time that was provided for the execution of the task execution request. The execution time was reduced to the maximum value for the *WAIT* parameter of the executed service.

The variable *time* shows the expiration time that was provided for the task execution request.

The variable *number* shows the value for the *WAIT* parameter of the executed service.

System action

Processing of the task execution request fails.

Operator response

None.

System programmer response

None.

ING251E	Invalid combination of filter elements <i>filter1</i> and <i>filter2</i>. Values <i>value1</i> and <i>value2</i> are mutually exclusive.
----------------	---

Explanation

The REXX macro implemented to execute the task execution request was passed an invalid combination of values for the filter elements.

The variables *filter1* and *filter2* show the names of the two filter elements.

The variables *value1* and *value2* show the values of the two filter elements.

System action

Processing of the task execution request fails.

Operator response

Contact IBM Support.

System programmer response

None.

ING252E	Domain initialization failed. Automation environment not initialized.
----------------	--

Explanation

Initialization of the automation domain failed because initialization of the system automation environment of the primary agent has not yet been completed.

System action

The start of the automation adapter fails.

Operator response

Restart the automation adapter after the initialization of the automation environment of the primary agent has completed.

System programmer response

None.

ING253E	Domain initialization failed. Primary agent already started on system <i>system</i>.
----------------	---

Explanation

Initialization of the automation domain failed because the primary agent is already running on another system within the subplex.

The variable *system* shows the system name.

System action

The start of the automation adapter fails.

Operator response

Check whether the automation adapter should be stopped on the other system.

System programmer response

None.

ING254E	Domain initialization failed. PPI receiver <i>receiver</i> of EAS not ACTIVE.
----------------	--

Explanation

Initialization of the automation domain failed because the PPI receiver ID of the NetView event/automation service is not active.

The variable *receiver* shows the name of the PPI receiver.

System action

The start of the automation adapter fails.

Operator response

Analyze the status of the PPI receivers by using command DISPPI. Check whether the event/automation service has been started and has successfully initialized its receiver ID at the PPI.

System programmer response

None.

ING255E	Requesting information from Automation Manager of domain <i>domain</i> failed.
----------------	---

Explanation

Requesting information from automation manager of an automation domain failed. Requesting information from automation is necessary during the initialization of the automation domain or when requesting information about the automation domain.

The variable *domain* shows the name of the automation domain.

System action

The task execution request fails.

Operator response

None.

System programmer response

Analyze the problem. Examine the NetView log for further information. If the problem persists contact IBM Support.

ING256E	Health check failed. Event forwarding switched off.
----------------	--

Explanation

Events were lost when trying to forward them from the primary agent to the automation adapter. Event

forwarding has therefore been switched off, which results in failing the health check of the automation domain.

System action

The automation adapter will stop. A subsequent restart enforces the re-initialization of the automation adapter and a refresh of the status information about first-level automation resources on the end-to-end server.

Operator response

Check the communication path from the primary automation agent to the NetView event/automation service. As soon as the communication path is available again, the automation adapter can be restarted.

System programmer response

None.

ING257E	Health check failed. PPI receiver receiver of EAS not ACTIVE.
----------------	--

Explanation

The PPI receiver identifier of the event/automation service is not in the status ACTIVE. This results in failing the health check of the automation domain.

The variable *receiver* shows the PPI receiver ID of the NetView event/automation service.

System action

The automation adapter does not send a heartbeat event, and so the communication status of the domain on the SMU server and E2E agent changes.

Operator response

Analyze the status of the PPI receivers by using command DISPPI. Check whether the event/automation service has been started and has successfully initialized its receiver ID at the PPI.

System programmer response

None.

ING258E	Execution of service <i>service</i> failed. Unexpected data received.
----------------	--

Explanation

The requested service did not return the expected data.

The variable *service* shows the name of the requested service.

System action

Processing of the task execution request fails.

Operator response

Notify your system programmer.

System programmer response

Determine why the service routine fails. Examine the NetView log for additional information.

ING259E	Invalid request element. Request name unknown: <i>name</i>
----------------	---

Explanation

An end-to-end request with an unknown name was received.

The variable *name* shows the request name.

System action

Processing of the task execution request fails.

Operator response

Contact IBM Support.

System programmer response

None.

ING260E	Invalid request element. Request type unknown: <i>type</i>
----------------	---

Explanation

An end-to-end request with an unknown type was received.

The variable *type* shows the request type.

System action

Processing of the task execution request fails.

Operator response

Contact IBM Support.

System programmer response

None.

ING261E	Invalid request element. Multiple solicited requests are not supported.
----------------	--

Explanation

Multiple solicited end-to-end requests were received to be executed by the primary agent. Only one solicited end-to-end request can be accepted.

System action

Processing of the task execution request fails.

Operator response

Contact IBM Support.

System programmer response

None.

ING262W	An attempt was made to overwrite at least one restricted parameter. Input parameters have been ignored: <i>parameters</i>
----------------	--

Explanation

An end-to-end request was received with parameters that attempted to overwrite restricted parameters. The parameters are ignored.

The variable *parameters* shows a list of the ignored parameters.

System action

Processing of the task execution request fails.

Operator response

Contact IBM Support.

System programmer response

None.

ING263E	Request rejected because mandatory parameter SOURCE is missing.
----------------	--

Explanation

An end-to-end request was received without the specified originator of the request via parameter SOURCE.

System action

Processing of the task execution request fails.

Operator response

Contact IBM Support.

System programmer response

None.

ING264E	Reset_from_NRE rejected because agent status of resource <i>resource</i> is CTLDDOWN.
----------------	--

Explanation

A reset-from-non-recoverable-error request was received for a resource that is in agent status CTLDDOWN. Such a request is not supported for resources in agent status CTLDDOWN.

The variable *resource* shows the name of the resource that the request was issued for.

System action

Processing of the task execution request fails.

Operator response

Check the agent status of the resource that the request was issued for.

System programmer response

None.

ING265E	Reset_from_NRE rejected because observed status of resource <i>resource</i> is not HARDDOWN.
----------------	---

Explanation

A reset-from-non-recoverable-error request was received for a resource that is not in status HARDDOWN. Such a request is only supported for resources in status HARDDOWN.

The variable *resource* shows the name of the resource that the request was issued for.

System action

Processing of the task execution request fails.

Operator response

Check the status of the resource.

System programmer response

None.

ING266E	Reset_from_NRE failed because no data to resource <i>resource</i> received from Automation Manager.
----------------	--

Explanation

A reset-from-non-recoverable-error request was received for a resource. An error occurred when requesting information from the automation manager about the resource that the request was issued against.

The variable *resource* shows the name of the resource that the request was issued for.

System action

Processing of the task execution request fails.

Operator response

Check whether the resource that the request was issued for is known to the automation manager.

System programmer response

None.

ING267E	Reset_from_NRE rejected because resource <i>resource</i> is not of class APL.
----------------	--

Explanation

A reset-from-non-recoverable-error request was received for a resource that is not of class APL. This request is only supported for resources of type APL.

The variable *resource* shows the name of the resource that the request was issued for.

System action

Processing of the task execution request fails.

Operator response

Check the type of the resource that the request was issued for.

System programmer response

None.

ING268E	<i>request</i> rejected because system name is missing.
----------------	--

Explanation

An end-to-end request concerning a system was issued, but the system name was not specified.

The variable *request* shows the name of the end-to-end request.

System action

Processing of the task execution request fails.

Operator response

Check that the request was issued against a resource of class SYS.

System programmer response

None.

ING269E	Native command contains an invalid value for parameter \$WAIT: <i>wait</i>
----------------	---

Explanation

An invalid value was specified as the WAIT parameter when issuing a native command.

The variable *wait* shows the value for the WAIT parameter that was received.

System action

Processing of the task execution request fails

Operator response

Contact IBM Support.

System programmer response

None.

ING270E	User <i>user</i> is not authorized to execute command: <i>command</i>
----------------	--

Explanation

The execution of the specified command is rejected by the authorization user exit AOFEXE2E.

The variable *user* shows the user ID that is associated with the command.

The variable *command* shows the rejected command resulting from the related end-to-end request.

System action

Processing of the task execution request fails

Operator response

Contact your system programmer to obtain the necessary authorization to execute the command.

System programmer response

If it is appropriate, authorize the user to issue the specified command.

ING271E	Command execution failed. Waiting for asynchronous output timed out.
----------------	---

Explanation

A timeout occurred when waiting for asynchronous output generated by a command that was issued from within a PIPE.

System action

Processing of the task execution request fails

Operator response

Retry the operation. If the problem persists, contact your system programmer.

System programmer response

Determine why the command does not return its output within the expected time. Examine the NetView log for additional information.

ING272E	More than one request in a list failed.
----------------	--

Explanation

More than one request in a list of requests that were included in a task execution request failed.

System action

Processing of the related requests fails.

Operator response

Examine the NetView log for additional information about the single requests that were processed with errors.

System programmer response

None.

ING273W	More than one request in a list processed with warnings.
----------------	---

Explanation

More than one request in a list of requests that were included in a task execution request were processed with warnings.

System action

Processing of the task execution request continues.

Operator response

Examine the NetView log for additional information about the single requests that were processed with warnings.

System programmer response

None.

ING274E	A query to the automation manager failed due to unsuccessful execution of service INGPYAMC.
----------------	--

Explanation

An error occurred when requesting information from the automation manager via the automation manager API INGPYAMC.

System action

Processing of the task execution request fails.

Operator response

Check the communication path from the primary agent to the automation manager and examine the NetView log for additional information. If the problem persists contact IBM Support.

System programmer response

None.

ING275E **Processing failed for command:
command**

Explanation

A command was called that ended with a non-zero return code.

The variable *command* shows the failed command.

System action

Processing of the command stops.

Operator response

Determine why the command failed and correct the problem.

System programmer response

None.

ING276E **Execution of a service timed out.**

Explanation

The return code of the requested service indicates a timeout problem.

System action

The requested service does not complete successfully.

Operator response

Retry the operation. If the problem persists, contact your system programmer.

System programmer response

Determine why the service routine timed out. Examine the NetView log for additional information.

INGX9701I **ingadapter.sh [-](start|stop|
generateSampleKeys|
IBMSupport|traceON|traceOFF|
traceMIN) [suffix]**

Explanation

Describes the syntax of the automation adapter start-stop command. It is also shown if there are invalid arguments or no arguments specified.

System action

None.

Operator response

Specify command syntax as described.

INGX9702I **Script ingadapter.sh successfully
finished processing.**

Explanation

Processing of the end-to-end automation adapter script ended successfully.

System action

None.

Operator response

None.

INGX9703E **Script ingadapter.sh terminated
unsuccessfully.**

Explanation

An error occurred during script execution.

System action

Script execution is interrupted.

Operator response

Check previous error messages for more details.

INGX9704I **Preparing the environment to start
the automation adapter.**

Explanation

The script *ingadapter.sh* performs a validation and a cleanup step before starting the automation adapter. The script determines whether an automation adapter on the same port is already running, or if there are any open connections left by a failed automation adapter shutdown. In the former case the script aborts and in the latter case the connections are dropped. Moreover, any existing pid-file of a failed automation adapter shutdown is deleted.

System action

None.

Operator response

If the automation adapter cannot be started or the connections cannot be dropped, check whether the user ID that was used to run the automation adapter has the correct access rights. For details see [Chapter 11, “Operating the End-to-End Automation Adapter,”](#) on page 53.

INGX9705E **Cannot start automation adapter with the same configuration: hostname:port**

Explanation

The automation adapter cannot be started twice with the same port number and hostname as specified in the automation adapter master configuration file. This is determined by examining an existing pid-file whose filename is constructed from the hostname and port number. The containing process-id is used to determine whether another automation adapter is running or if the pid-file has been left over by a failed automation adapter shutdown. If the corresponding automation adapter USS process does not exist then delete the pid file and restart automation adapter. The pid file resides in the automation adapter data directory with extension '.pid'.

System action

Script execution is interrupted.

Operator response

Start the automation adapter with a different configuration.

INGX9706I **Trying to drop possible open, unused connections. Processing port: port_number**

Explanation

The script tries to close possible open and unused connections that have been left by a failed automation adapter shutdown. These connections are detected with the netstat command. The connection will be dropped using the command `netstat -d conid`, if specific access rights have been granted.

System action

Connections are dropped.

Operator response

If the `ingadapter.sh` script cannot drop connections check whether the user ID that was used to run the

automation adapter has the correct access rights. For details see [Chapter 11, “Operating the End-to-End Automation Adapter,”](#) on page 53.

INGX9707I **Status of the automation adapter: adapter is running.**

Explanation

Information about the status of the automation adapter.

System action

None.

Operator response

None.

INGX9708I **Status of the automation adapter: adapter is not running.**

Explanation

Information about the status of the automation adapter.

System action

None.

Operator response

None.

INGX9709E **Rights to access process list have not been granted. Change permissions.**

Explanation

The necessary rights to show all processes have not been granted to the current user. The script cannot correctly determine the automation adapter status.

System action

Script execution is interrupted.

Operator response

Change access rights to show all processes. For more information, see [Chapter 9, “Customizing USS and TCP/IP,”](#) on page 41.

INGX9710I **Could not retrieve installed codepage. Trying with default codepage:**

Explanation

The script was not able to determine the installed codepage because environment variable `CODEPAGE` has not been set correctly. It attempts to use the default codepage.

System action

The system uses the default codepage.

Operator response

Set the environment variable `CODEPAGE` in the configuration file `ing.adapter.plugin.properties` to the desired codepage.

INGX9711E **The automation adapter configuration file is missing.**

Explanation

The automation adapter master configuration file, `ing.adapter{suffix}.properties`, was not found. The suffix is optional and can be specified as the second argument of `ingadapter.sh`.

System action

Script execution is interrupted.

Operator response

If you use a suffix make sure you have specified it as the second argument for `ingadapter.sh`. The automation adapter master configuration file is expected to be located in the configuration directory. Also check the customization section of the script `ingadapter.sh`. Check manual for corresponding information.

INGX9712I **Generating Sample Keys for test purposes.**

Explanation

The script generates sample keystore and truststore files for SSL. These keys are for testing purpose only.

System action

None.

Operator response

None.

INGX9713E **Keytool was not found. Please install a java-sdk and customize ingadapter.sh accordingly.**

Explanation

The automation adapter script, `ingadapter.sh`, attempted to use the `keytool` from the Java SDK but the tool was not found. The `keytool` is used to generate sample keys for SSL.

System action

Cannot create sample keys.

Operator response

Install a Java SDK and adapt the path to the `keytool` using the variable `JAVA_KEYTOOL` in the customization section of the script `ingadapter.sh`.

INGX9714I **Trace was set to: *level*.**

Explanation

The script has successfully set the specified level. *level* is either `OFF` or `DEBUG_MAX`.

System action

None.

Operator response

None.

INGX9715E **An error occurred while trying to set log level.**

Explanation

The script failed in at least one case to set the specified log level. Use the counterpart of the command to reset the logger to its previous state. Ignore failures that are reported by the script in this undo step. Those commands that succeeded the first time will succeed in the undo step and those that did not will not succeed the second time.

System action

None.

Operator response

Undo changes by invoking the opposite trace command: for example, `traceOFF` if `traceON` has failed.

INGX9733I **BAD java version foundVersion,
required is requiredVersion**

Explanation

The USS script compares the required minimum java version with the actual version found by *java version*.

System action

Processing terminates.

Operator response

None.

System programmer response

Install and set up at least the required java version.

INGX9800E **Cannot find message file or
message ID *file_or_msgid***

Explanation

The message file cannot be loaded or the message ID cannot be found in the message file.

The variable *file_or_msgid* shows the name of the associated message file or ID.

System action

None.

Operator response

Check the class path.

System programmer response

None.

INGX9801E **INGXLogger cannot setup
com.ibm.log.PDLogger**

Explanation

Initialization of the message and trace log service failed.

System action

The message or trace data cannot be written to the log file.

Operator response

Check the classpath or the logger configuration file, *eezjlog.properties*.

System programmer response

None.

INGX9802I **INGXLogger has successfully been
initialized using configuration file
config_file from path *path*.**

Explanation

Initialization of the message and trace log service was successful. The logger configuration data were read from the specified configuration file.

The variable *config_file* shows the name of the configuration file that the logger configuration data was read from.

The variable *path* shows the path of the configuration file.

System action

Message and trace data will be written to the log file.

Operator response

None.

System programmer response

None.

INGX9803E **INGXLogger is not available :
*message***

Explanation

An attempt was made to write a message to the logger but the logger was not initialized.

The variable *message* shows the message that could not be written to the logger.

System action

The message is written to the *stderr.txt* file.

Operator response

None

System programmer response

None.

INGX9810I **Timeout after *interval* seconds.****Explanation**

A connection from the JVM to the SA z/OS communication manager (INGPXDST) timed out.

The variable *interval* shows the duration of the timeout.

System action

The response from the associated request is not received.

Operator response

Contact your system programmer.

System programmer response

Check the defined timeout interval for the request. If necessary, increase it as described in “[Requests issued by the SMU and the end-to-end automation manager](#)” on page 97.

INGX9820E **JNI function *function* failed with return code *return_code*.****Explanation**

A function call to the specified JNI DLL failed.

- The variable *function* shows the function call that failed.
- The variable *return_code* shows the return code that specifies the reason for the failure:

10 Internal—invalid field ID (attribute not available in Java class)

20 Internal—handle cannot be saved in Java long since long too small

21 Internal—handle is invalid, for example, null pointer

22 Handle cannot be created since no memory

23 Internal—handle iterator already used (number of slots too small)

25 The ASCB address is no longer correct. Recycle the End-to-end automation adapter.

26 Internal—invalid array size (Java array size is zero or too small)

27 Internal—invalid object (null pointer or an unexpected class)

30 Internal—invalid parameter (unexpected input argument)

31 Invalid buffer length (input data buffer has unexpected length)

32 A Java class that was accessed via JNI has thrown an exception

33 Creation of a new Java array failed

34 Creation of a new Java string failed

35 Out of memory

36 A Java class cannot be found

40 Catastrophic error—an unexpected exception occurred

1000 PPI receive timeout

11nn PPI subsystem not available

12nn PPI initialization of receiver queue failed

13nn The PPI receiver queue exists already

14nn PPI NetView posted PPI ECB due to subsystem error

15nn PPI cannot receive data

16nn PPI cannot delete receiver queue

17nn PPI cannot send data

where *nn* denotes the PPI-specific return code as described in the *NetViewApplication Programmer's Guide*, for example:

04 The PPI receiver is not active. For example, the SA z/OS agent is not available or is not properly customized.

24

The PPI is not active.

25

The ASCB address is no longer correct. Recycle the automation adapter.

26

The receiver program is not defined, for example, PPI=YES has not been added to INGXINIT.

28

An active subsystem interface address space was found, but an active PPI address space was not found.

32

No NetView storage is available.

35

The receiver buffer queue is full.

System action

The corresponding request failed.

Operator response

Analyze the return code. If necessary, contact your system programmer.

System programmer response

Analyze the return code. For example:

- RC=1300 indicates that another automation adapter is running on the system. Verify with D OMVS, A=nnnn whether there is another automation adapter running, where nnnn is the ASID of INGE2Exx that you obtained with the NetView DISPPI command.
- RC=1726 indicates that there might be a mismatch of the XCF group ID in ing.adapter.plugin.properties. Verify whether it matches the XCF group ID that was defined in INGXINIT for the automation agent and HSAPRMxx for the automation manager. Alternatively, the statement PPI=YES might be missing in the INGXINIT member.

INGX9821E **Object of class *class* has already been destroyed.**

Explanation

Internal error.

System action

The corresponding request failed.

Operator response

Contact IBM Support.

System programmer response

None.

INGX9822E **Environment error. Check handle failed with return code *return_code*.**

Explanation

Internal error.

System action

The corresponding request failed.

Operator response

Contact IBM Support.

System programmer response

None.

INGX9823E **Class *class* cannot load DLL *dll*.**

Explanation

The JNI DLL cannot be loaded from the specified Java class.

System action

All communication services will fail.

Operator response

Check the libpath.

System programmer response

None.

INGX9830E **INGXStream failed reading from the data stream. Attempted to read *expected* bytes and got *actual* bytes.**

Explanation

The data stream decoder expected to read *expected* number of bytes but got only *actual* bytes.

System action

The corresponding request fails.

Operator response

Check why connection was broken.

System programmer response

None.

INGX9831E	INGXStream failed reading from the data stream. Found unexpected end-of-data.
------------------	--

Explanation

The data stream decoder did not find the end-of-data marker. Data might be corrupted. The connection might have broken or there is an internal data stream error.

System action

The corresponding request fails.

Operator response

Check why connection was broken.

System programmer response

None.

INGX9833E	INGXStream failed due to invalid header.
------------------	---

Explanation

The data stream decoder found an invalid header in the data stream.

System action

The corresponding request fails.

Operator response

Contact IBM Support.

System programmer response

None.

INGX9840E	Execution of a remote REXX command handler aborted.
------------------	--

Explanation

The corresponding request cannot be executed within the SA z/OS NetView backend.

System action

The corresponding request fails.

Operator response

Analyze the NetView log for more SA z/OS messages or REXX syntax errors, or both.

System programmer response

None.

INGX9901E	INGXPluginLogger cannot initialize com.ibm.log.EEZLoggerJLog.
------------------	--

Explanation

An attempt to create a logger for the SA z/OS plug-in component was unsuccessful.

System action

No message or trace data is written to the logger output destination.

Operator response

None.

INGX9902I	INGXPluginLogger has successfully been initialized using configuration file <i>file</i> from path <i>path</i>.
------------------	---

Explanation

The message and trace data of the SA z/OS plug-in will be written to the logger output destination.

System action

None.

Operator response

None.

INGX9904E	Adapter plug-in initialization was unsuccessful.
------------------	---

Explanation

The initialization of the SA z/OS plug-in was not successful.

System action

Start up of plug-in is stopped.

Operator response

Installation or customization was erroneous, for example, the logger or the plug-in configuration file. Also check whether the automation adapter customization for NetView is correct.

INGX9905E **Adapter function *function* failed due to exception from underlying component: *exception text***

Explanation

The specified automation adapter plug-in function was unsuccessful. An exception was caught from an underlying component.

System action

None.

Operator response

Read the original exception and follow its explanation.

INGX9906E **Error message from SA z/OS: SA z/OS message**

Explanation

The execution of a command using SA z/OS was unsuccessful. This message presents the extracted error message.

System action

None.

Operator response

For details about the extracted error message see *IBM System Automation for z/OS Messages and Codes* or *Tivoli NetView for z/OS Messages and Codes*.

INGX9907I **Adapter plug-in cannot contact the System Automation NetView Agent, elapsed time is *time* seconds.**

Explanation

The SA z/OS plug-in detected that a send request failed with message INGX9820E and return code 1704 and this error situation cannot be resolved within certain time period. It is assumed that the SA z/OS NetView Agent is no longer available.

System action

The automation adapter stops intentionally.

Operator response

Restart the SA z/OS NetView Agent. See also the TIMEOUT_SA_NOTAVAIL parameter in the plug-in configuration file.

INGX9910I **The command *cmd* will be performed for resource *res* on behalf of user *usr* with comment: *cmt***

Explanation

A user has issued a command that relates to a resource. This command has been received by the automation adapter and will be performed by automation on the indicated resource.

The variable *cmd* shows the end-to-end command that was issued against the automation adapter.

The variable *res* shows the name of the SA z/OS resources.

The variable *usr* shows the end-to-end operator name that issued the command.

The variable *cmt* shows the comment that was entered by the operator.

System action

The command has been received and will be executed for the resource.

Operator response

None.

INGX9911I **The command *cmd* will be performed for system *sys* on behalf of user *usr***

Explanation

A user has issued a command that relates to a system. This command has been received by the automation adapter and will be performed by automation on the specified system.

The variable *cmd* shows the end-to-end command that was issued against the automation adapter.

The variable *sys* shows the name of the system that is involved with the command.

The variable *usr* shows the end-to-end operator name that issued the command.

System action

The command has been received and will be executed.

Operator response

None.

INGX9912I **The command to move resource group *grp* away from system *sys* will be performed on behalf of user *usr* with comment: *cmt***

Explanation

A user has issued a request to move a resource group away from specified system. This command has been received by the automation adapter and will be performed by automation.

The variable *grp* shows the name of the group that is involved with the command.

The variable *sys* shows the name of the system that is involved with the command.

The variable *usr* shows the end-to-end operator name that issued the command.

The variable *cmt* shows the comment that was entered by the operator.

System action

The command has been received and will be executed.

Operator response

None.

INGX9913I **The command to move resource group *grp* to the home system will be performed on behalf of user *usr* with comment: *cmt***

Explanation

A user has issued a command that relates to a resource. This command has been received by the automation adapter and will be performed by automation on the indicated resource.

The variable *grp* shows the name of the group that is involved with the command.

The variable *usr* shows the end-to-end operator name that issued the command.

The variable *cmt* shows the comment that was entered by the operator.

System action

The command has been received and will be executed.

Operator response

None.

INGX9914I **The command to move resource group *grp* to system *sys* will be performed on behalf of user *usr* with comment: *cmt***

Explanation

A user has issued a command that relates to a resource. This command has been received by the automation adapter and will be performed by automation on the indicated resource.

The variable *grp* shows the name of the group that is involved with the command.

The variable *sys* shows the name of the system that is involved with the command.

The variable *usr* shows the end-to-end operator name that issued the command.

The variable *cmt* shows the comment that was entered by the operator.

System action

The command has been received and will be executed.

Operator response

None.

INGX9915I **Command reset from non-recoverable error will be performed on resource *res* on behalf of user *usr***

Explanation

A user found a non-recoverable resource, which means the resource is not automated until a user issues the reset command. The reset command has been received by the automation adapter and will be performed by automation on the indicated resource.

The variable *res* shows the name of the SA z/OS resources.

The variable *usr* shows the end-to-end operator name that issued the command.

System action

The command has been received and will be executed for the resource.

Operator response

None.

Appendix F. Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

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

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year).

Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. _enter the year or years_.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Index

A

- AdapterJoinEvent [89](#)
- additional information, automation domain [93](#)
- advanced automation option
 - AOF_E2E_EAS_PPI [38](#), [74](#)
 - AOF_E2E_EVT_RETRY [38](#)
 - AOF_E2E_EXREQ_NETLOG [38](#)
 - AOF_E2E_TKOVN_TIMEOUT [38](#)
 - customizing [38](#)
 - INGREQ_PRI.E2EMGR [38](#), [99](#)
 - setting request priority [99](#)
 - setting the PPI receiver ID [74](#)
- AOF_E2E_EAS_PPI advanced automation option [38](#)
- AOF_E2E_EVT_RETRY advanced automation option [38](#)
- AOF_E2E_EXREQ_NETLOG advanced automation option [38](#)
- AOF_E2E_TKOVN_TIMEOUT advanced automation option [38](#)
- AOF233E message, diagnosing domain initialization failure [71](#)
- AOFEXE2E sample exit and overwriting commands [99](#)
- application definition
 - automation adapter [81](#)
 - event/automation service [81](#)
- Application Manager
 - connection [79](#)
 - library [79](#)
- authentication [51](#)
- authentication checking [57](#)
- authentication, operator [46](#)
- authorization checking [57](#)
- authorization exit [57](#)
- authorization user exit, sample [58](#)
- Auto Operators entry type [30](#), [37](#)
- automated applications defined to SA z/OS [87](#)
- automated function [48](#), [89](#)
- automated operator functions
 - defining [30](#), [37](#)
 - EVTOPER, checking status [71](#)
- automation adapter
 - application definition [81](#)
 - asynchronous communication [5](#)
 - communication [4](#)
 - communication flow [46](#)
 - configuration files [45](#)
 - configuring [45](#)
 - customizing [45](#)
 - dependencies [15](#)
 - event/automation service [6](#)
 - failover to a backup system [87](#)
 - grouping considerations [87](#)
 - high availability [81](#)
 - host name [46](#), [89](#)
 - host name port [89](#)
 - ingadapter.sh [49](#)
 - initial contact timeout interval [89](#)
 - initial contact with SMU server [89](#)
 - automation adapter (*continued*)
 - introduction [1](#), [3](#)
 - IP address [46](#), [89](#)
 - IP address port [89](#)
 - Java Virtual Machine access [46](#)
 - master configuration file [46](#)
 - messages [109](#)
 - non-disruptive recovery from errors [86](#)
 - operating [53](#)
 - plug-in domain name [48](#), [89](#)
 - plug-in, messages [122](#)
 - prerequisites [15](#)
 - problem determination [71](#)
 - purpose and rules [85](#)
 - quick startup [29](#)
 - recycling after stopping NetView [76](#)
 - recycling after stopping SSI subsystem [76](#)
 - recycling in the same system [86](#)
 - remote contact activity interval [89](#)
 - restart [87](#)
 - resynchronization of [86](#)
 - security [57](#)
 - start-stop script [49](#)
 - start-stop script, and SSL configuration file [49](#)
 - start-stop script, using [53](#)
 - startup using INGXADPT [32](#)
 - stop delay [89](#)
 - synchronous communication [5](#)
 - TCP/IP port relationship [46](#)
 - two in same sysplex [6](#)
- automation agent, primary [6](#)
- automation domain
 - additional information [93](#)
 - general information [93](#)
 - policy information [93](#)
 - remaining unavailable after error recovery [76](#)
- automation manager status
 - mapping to resource data model status [105](#)
- automation manager, problem with automation domain after restarting [76](#)
- automation policy, defining [81](#)
- AUTOPFN parameter, description [89](#)

B

- backup system, failover of the automation adapter to [87](#)
- bound socket, dropping with NETSTAT [75](#)
- buffer queue limit
 - description [48](#), [89](#)

C

- CEE3501S error message, and link list errors [74](#)
- certificate alias name [89](#)
- certificates [49](#)
- cfgsmu
 - Service Management Unite utility [50](#)

- CHECK_HEALTH function failure [76](#)
- code page, Java [89](#)
- CODEPAGE parameter, description [89](#)
- commands, overwriting for resource objects [99](#)
- common global variable
 - customizing [38](#)
 - INGREQ_PRI.E2EMGR [99](#)
 - setting request priority [99](#)
- communication
 - automation adapter, asynchronous [5](#)
 - automation adapter, introduction [4](#)
 - automation adapter, synchronous [5](#)
 - event/automation service [6](#)
- communication flow, automation adapter [46](#)
- concurrent working threads
 - and automated operator functions [37](#)
 - automated operator functions
 - and concurrent working threads [37](#)
- configuration file
 - example plug-in [48](#)
 - JAAS [51](#)
 - Java logger [71](#)
 - master [46](#)
 - plug-in [48](#)
 - SSL [49](#)
- configuration files [45](#)
- configuring
 - automation adapter [45](#)
- Custom-Root directory [16](#)
- customizing
 - advanced automation options [38](#)
 - automation adapter [45](#)
 - UNIX System Services [41](#)

D

- defining
 - application for event/automation service [81](#)
 - application for the automation adapter [81](#)
 - automated operator functions [30](#), [37](#)
 - automation policy [81](#)
 - E2OPER automated function [37](#)
- delaying the stopping of the automation adapter [89](#)
- DISPAOPS command response [71](#)
- DISPPI command, using for problems with program-to-program interface [73](#)
- domain initialization failure, diagnosing [71](#)
- domain object [93](#)
- domain state [93](#)
- dropping bound socket with NETSTAT [75](#)
- DSIPARM data set
 - customizing [30](#)

E

- E2OPER automated function, defining [37](#)
- eez-initial-contact parameter, description [89](#)
- eez-initial-contact-retry-interval-minutes parameter, description [89](#)
- eez-max-connections parameter, description [89](#)
- eez-operator-authentication parameter [57](#)
- eez-operator-authentication parameter, description [89](#)

- eez-plugin-classpath parameter, description [89](#)
- eez-remote-contact-activity-interval-seconds parameter, description [89](#)
- eez-remote-contact-hostname parameter, description [89](#)
- eez-remote-contact-over-ssl parameter, description [89](#)
- eez-remote-contact-port parameter, description [89](#)
- eez-ssl-keystore parameter, description [89](#)
- eez-ssl-keystorealias parameter, description [89](#)
- eez-ssl-keystorepassword parameter, description [89](#)
- eez-ssl-truststore parameter, description [89](#)
- eez-stop-delay parameter, description [89](#)
- EEZA0001E [109](#)
- EEZA0002E [109](#)
- EEZA0003E [109](#)
- EEZA0004E [109](#)
- EEZA0005I [109](#)
- EEZA0006E [110](#)
- EEZA0007E [110](#)
- EEZA0008E [110](#)
- EEZA0009E [110](#)
- EEZA0010E [111](#)
- EEZA0011E [111](#)
- EEZA0012E [111](#)
- EEZA0013E [112](#)
- EEZA0014E [112](#)
- EEZA0015E [112](#)
- EEZA0017E [112](#)
- EEZA0022E [112](#)
- EEZA0023E [112](#)
- EEZA0024E [113](#)
- EEZA0025E [113](#)
- EEZA0026E [113](#)
- EEZA0027E [113](#)
- EEZA0028E [113](#)
- EEZA0029E [113](#)
- EEZA0030E [113](#)
- EEZA0031E [114](#)
- EEZA0032E [114](#)
- EEZA0033E [114](#)
- EEZA0036E [114](#)
- EEZA0037E [114](#)
- EEZA0038E [114](#)
- EEZA0039E [115](#)
- EEZA0040E [115](#)
- EEZA0041E [115](#)
- EEZA0042E [115](#)
- EEZA0043E [115](#)
- EEZA0045E [115](#)
- EEZA0047E [116](#)
- EEZA0051W [116](#)
- EEZA0052E [116](#)
- EEZA0053E [116](#)
- EEZA0055E [116](#)
- EEZA0056I [116](#)
- EEZA0057E [116](#)
- EEZA0058E [117](#)
- EEZA0059E [117](#)
- EEZA0060I [117](#)
- EEZA0061E [117](#)
- EEZA0061E message [75](#)
- EEZA0062I [117](#)
- EEZA0063I [118](#)
- EEZA0064I [118](#)
- EEZA0070E [118](#)

- EEZA0071E [118](#)
- EEZA0100I [118](#)
- EEZA0101I [118](#)
- EEZA0102I [118](#)
- EEZA0103I [118](#)
- EEZA0104I [119](#)
- EEZA0105I [119](#)
- EEZA0111I [119](#)
- EEZA0112I [119](#)
- EEZA0113I [119](#)
- EEZA0114I [119](#)
- EEZA0115I [120](#)
- EEZA0116I [120](#)
- EEZA9991E [120](#)
- EEZA9992E [120](#)
- EEZAdapterDefaultLogin [57](#)
- EEZC0001I [120](#)
- EEZC0002I [120](#)
- EEZC0003I [121](#)
- EEZC0004I [121](#)
- EEZK0003E [121](#)
- EEZK0004E [121](#)
- EEZK0005E [122](#)
- EIF cache directory [89](#)
- EIF caching, switching on and off [89](#)
- EIF receive host name [89](#)
- EIF receive port [46, 89](#)
- EIF send host name [46, 89](#)
- EIF send port [89](#)
- EIF socket reconnect interval [89](#)
- eif-cache parameter, description [89](#)
- eif-cache-size parameter, description [89](#)
- eif-receive-from-hostname parameter, description [89](#)
- eif-receive-from-port parameter, description [89](#)
- eif-retry-interval-seconds parameter, description [89](#)
- eif-send-to-hostname parameter, description [89](#)
- eif-send-to-port parameter, description [89](#)
- end-to-end automation
 - limitations [81](#)
 - login module [51](#)
 - Operations Console [93](#)
 - resource data model [93](#)
- end-to-end automation adapter
 - startup using INGADPT [32](#)
- end-to-end automation management, introduction [1](#)
- end-to-end automation manager
 - requests, overruling hold flag for [100](#)
- end-to-end automation manager, requests issued by [97](#)
- Environment variables [15](#)
- error recovery, automation domain remaining unavailable after [76](#)
- errors
 - link list [74](#)
 - non-disruptive recovery from [86](#)
- event log file [71](#)
- event/automation service
 - application definition [81](#)
 - communication with automation adapter [6](#)
 - configuring the global initialization file [36](#)
 - customizing the sample procedure [35](#)
 - problems, using MODIFY command for [73](#)
- event/automation service, starting [35](#)
- EVTOPER automated operator function, checking status [71](#)
- example plug-in configuration file [48](#)

examples of RACF profiles [59](#)

F

failover of the automation adapter to a backup system [87](#)
 functions, automated operator [37](#)

G

generateSampleKeys parameter, of ingadapter.sh start script [53](#)
 Global initialization file (IHSAINIT) [36](#)
 graphical user interface [71](#)
 group ID in INGXINIT [48, 89](#)
 grouping considerations for the automation adapter [87](#)
 GRPID parameter

- description [89](#)

H

high availability, automation adapter [81](#)
 hold flag, disregarding [100](#)
 host name

- automation adapter [46, 89](#)
- EIF receive [89](#)
- EIF send [46, 89](#)

I

IBMSupport parameter, of ingadapter.sh start script [53](#)
 IHSAEVNT [35](#)
 IHSAMCFG [36](#)
 ing.adapter.properties file [46](#)
 ING008I message and automation domain timeout [76](#)
 ING230E [122](#)
 ING231E [122](#)
 ING232I [122](#)
 ING233E [122](#)
 ING234E [123](#)
 ING235W [123](#)
 ING235W message and automation domain timeout [76](#)
 ING236W [123](#)
 ING237W [124](#)
 ING238E [124](#)
 ING239W [125](#)
 ING240E [125](#)
 ING241E [125](#)
 ING242E [125](#)
 ING243E [125](#)
 ING244E [126](#)
 ING245E [126](#)
 ING246E [126](#)
 ING247E [126](#)
 ING248E [126](#)
 ING249E [127](#)
 ING250E [127](#)
 ING251E [127](#)
 ING252E [128](#)
 ING253E [128](#)
 ING254E [128](#)
 ING255E [128](#)
 ING256E [128](#)
 ING257E [129](#)

[ING258E 129](#)
[ING259E 129](#)
[ING260E 129](#)
[ING261E 130](#)
[ING262W 130](#)
[ING263E 130](#)
[ING264E 130](#)
[ING265E 130](#)
[ING266E 131](#)
[ING267E 131](#)
[ING268E 131](#)
[ING269E 131](#)
[ING270E 131](#)
[ING271E 132](#)
[ING272E 132](#)
[ING273W 132](#)
[ING274E 132](#)
[ING275E 133](#)
[ING276E 133](#)
[ING276E message and automation domain timeout 76](#)
[ingadapter.sh shell script](#)
 [and nohup command 53](#)
 [and SSL configuration file 49](#)
 [parameters 53](#)
 [prerequisites 41](#)
 [syntax 53](#)
 [using 53](#)
[INGAMS command, using to check primary NetView agent 72](#)
[INGE2E command 100](#)
[INGE2E PREPARE command 31](#)
[INGE2E RESYNC command and automation domain timeout 76](#)
[INGE2E Verify command 33](#)
[INGEVOBS command, using to check end-to-end automation registration](#)
 [consumer registration, checking with INGEVOBS command 74](#)
[INGEVOBS LIST command response 74](#)
[INGPX DST communication task](#)
 [problem with automation domain after restarting 76](#)
[INGREQ_PRI.E2EMGR advanced automation option 38](#)
[INGREQ_WAIT common global variable, setting timeout with 76](#)
[INGX9701I 133](#)
[INGX9702I 133](#)
[INGX9703E 133](#)
[INGX9704I 133](#)
[INGX9705E 134](#)
[INGX9706I 134](#)
[INGX9707I 134](#)
[INGX9708I 134](#)
[INGX9709E 134](#)
[INGX9710I 134](#)
[INGX9711E 135](#)
[INGX9712I 135](#)
[INGX9713E 135](#)
[INGX9714I 135](#)
[INGX9715E 135](#)
[INGX9733I 136](#)
[INGX9800E 136](#)
[INGX9801E 136](#)
[INGX9802I 136](#)
[INGX9803E 136](#)
[INGX9810I 137](#)
[INGX9820 message and recycling automation adapter 76](#)
[INGX9820E 137](#)
[INGX9821E 138](#)
[INGX9822E 138](#)
[INGX9823E 138](#)
[INGX9830E 138](#)
[INGX9831E 139](#)
[INGX9833E 139](#)
[INGX9840E 139](#)
[INGX9901E 139](#)
[INGX9902I 139](#)
[INGX9904E 139](#)
[INGX9905 message and recycling automation adapter 76](#)
[INGX9905E 140](#)
[INGX9906E 140](#)
[INGX9907I 140](#)
[INGX9910I 140](#)
[INGX9911I 140](#)
[INGX9912I 141](#)
[INGX9913I 141](#)
[INGX9914I 141](#)
[INGX9915I 141](#)
[INGXADPT](#)
 [starting up end-to-end automation adapter 32](#)
[INGXINIT](#)
 [PPI setting 30](#)
[INGXINIT initialization member](#)
 [customizing 30](#)
[INGXINIT, group ID 48, 89](#)
[initial contact, automation adapter](#)
 [timeout interval 89](#)
 [with SMU server 89](#)
[Install-Root directory 16](#)
[introducing the automation adapter 1](#)
[IP address](#)
 [automation adapter 46, 89](#)
 [master configuration file 46](#)

J

[JAAS configuration file](#)
 [end-to-end automation login module 51](#)
 [overview 51](#)
 [sample 51](#)
 [z/OS login module 51](#)
[JAAS login module 57](#)
[Java](#)
 [classpath and security 89](#)
 [code page 89](#)
 [JIT compiler, and USS problems 76](#)
 [loading DLLs, problems with 76](#)
 [logger configuration file 71](#)
 [system properties, and SSL configuration file 49](#)
[Java Runtime](#)
 [setting up 15](#)
[Java Virtual Machine 46](#)

K

[keyboard xiii](#)
[keystore](#)
 [creating for use by SSL 51](#)

keystore file, password [89](#)
keystore filename [89](#)
keystore password
 encrypting, using cfgsmu [50](#)

L

limitations to end-to-end automation [81](#)
log file
 event [71](#)
 message [71](#)
 trace [71](#)
logger configuration file, Java [71](#)
login module
 end-to-end automation [51](#)
 JAAS [51](#)
 z/OS [51](#)

M

master configuration file
 introduction [46](#)
 IP addresses [46](#)
 parameters [89](#)
 sample [46](#)
message log file [71](#)
messages
 AOF233E and domain initialization failure [71](#)
 automation adapter [109](#)
 automation adapter plug-in [122](#)
 CEE3501S and link list errors [74](#)
 ING008I and automation domain timeout [76](#)
 ING235W and automation domain timeout [76](#)
 ING276E and automation domain timeout [76](#)
 INGX9820 and recycling automation adapter [76](#)
 INGX9905 and recycling automation adapter [76](#)
MODIFY command, for problems with event/automation service [73](#)
MVS MODIFY command, for problems with event/automation service [73](#)

N

NETSTAT command, using to drop bound socket [75](#)
NetView
 checking primary agent with INGAMS [72](#)
 event/automation service [35](#)
 event/automation service, enabling [32](#)
 event/automation service, problems with [73](#)
 event/automation service, verifying [33](#)
 operator task name [48, 89](#)
 PIPE default safe [57](#)
 planned outages [81](#)
 stopping, recycling automation adapter after [76](#)
NetView message adapter service
 configuring [36](#)
NetView SSI JCL procedure [32](#)
node, general information [94](#)
nohup command, and ingadapter.sh start script [53](#)
non-disruptive recovery from errors [86](#)

O

online, switching automation domain to after error recovery [76](#)
operating the automation adapter
 as a SA z/OS defined resource [55](#)
 from USS console [53](#)
 with customizable procedure, INGXADPT [30, 55](#)
operations console
 requests issued by [97](#)
operator authentication [46](#)
operator task name, NetView [48, 89](#)
overruling hold flag for end-to-end automation manager requests [100](#)
overwriting commands, for resource objects [99](#)

P

parameters
 AUTOPFN [89](#)
 CODEPAGE [89](#)
 eez-data-directory [89](#)
 eez-initial-contact [89](#)
 eez-initial-contact-retry-interval-minutes [89](#)
 eez-max-connections [89](#)
 eez-operator-authentication [57, 89](#)
 eez-plugin-classpath [89](#)
 eez-remote-contact-activity-interval-seconds [89](#)
 eez-remote-contact-hostname [89](#)
 eez-remote-contact-over-ssl [89](#)
 eez-remote-contact-port [89](#)
 eez-ssl-keystore [89](#)
 eez-ssl-keystorealias [89](#)
 eez-ssl-keystorepassword [89](#)
 eez-ssl-truststore [89](#)
 eez-stop-delay [89](#)
 eif-cache [89](#)
 eif-cache-size [89](#)
 eif-receive-from-hostname [89](#)
 eif-receive-from-port [89](#)
 eif-retry-interval-seconds [89](#)
 eif-send-to-hostname [89](#)
 eif-send-to-port [89](#)
 GRPID [89](#)
 master configuration file [89](#)
 plug-in configuration file [89](#)
 plugin-configfile-sa4zos [48, 89](#)
 plugin-domain-name [89](#)
 PPIBQL [89](#)
 SSL configuration file [89](#)
 TIMEOUT [89](#)
 TIMEOUT_SA_NOTAVAIL [89](#)
password [57](#)
plug-in configuration file
 example [48](#)
 introduction [48](#)
 name, setting [89](#)
 parameters [89](#)
plug-in domain name [48, 89](#)
plugin-configfile-sa4zos parameter, description [48, 89](#)
plugin-domain-name parameter, description [89](#)
policy information, automation domain [93](#)
port
 automation adapter host name [89](#)

- port (*continued*)
 - EIF receive [46, 89](#)
 - EIF send [89](#)
- PPI
 - enabling for the NetView subsystem interface address space (SSI) [32](#)
- PPI buffer queue limit [48, 89](#)
- PPI communication
 - setting in INGXINIT [30](#)
- PPI queue, number of elements [48, 89](#)
- PPIBQL parameter
 - description [89](#)
- prerequisites for UNIX System Services [41](#)
- prerequisites, automation adapter [15](#)
- primary automation agent [6](#)
- primary NetView agent, problems selecting [72](#)
- problem determination
 - automation domain remaining unavailable after error recovery [76](#)
 - consumer registration [74](#)
 - domain initialization failure [71](#)
 - introduction [71](#)
 - link list errors [74](#)
 - NetView event/automation service [73](#)
 - program-to-program interface [73](#)
 - selecting the primary agent [72](#)
 - TCP/IP bind socket [75](#)
 - timeout of requests [75](#)
- profile
 - RACF [58](#)
 - RACF, examples of [59](#)
- program-to-program interface, problems with [73](#)

Q

- Quick startup
 - automation adapter [29](#)

R

- RACF
 - profile [58](#)
 - profile, examples [59](#)
- reconnect interval, EIF socket [89](#)
- recycling the automation adapter in the same system [86](#)
- REFRESHSTART STARTUP policy and CHECK_HEALTH function failure [76](#)
- relations to resource objects [96](#)
- relationship, TCP/IP port and automation adapter [46](#)
- remote contact activity interval [89](#)
- request mapping to SA z/OS commands for resource objects [97](#)
- request priority, for resource objects [99](#)
- requests
 - issued by end-to-end automation manager [97](#)
 - issued by operations console [97](#)
 - issued to resource objects [96](#)
 - overruling hold flag for [100](#)
 - overwriting commands from [99](#)
 - timeout problems with [75](#)
- resource data displayed by WebUI [93](#)
- resource data model
 - automation domain, additional information [93](#)

- resource data model (*continued*)
 - automation domain, policy information [93](#)
 - compound status [107](#)
 - domain object [93](#)
 - domain state [93](#)
 - node [94](#)
 - node, general information [94](#)
 - operational state [107](#)
 - relations to resource objects [96](#)
 - request mapping to SA z/OS commands [97](#)
 - resource object [95](#)
 - resource object, additional information [95](#)
 - resource object, general information [95](#)
 - resource object, overwriting commands for [99](#)
 - resource object, relationships information [96](#)
 - resource object, request priority [99](#)
 - resource object, requests issued to [96](#)
 - status [104](#)
 - status mapping [104](#)
 - status mapping rules [104](#)
 - status mapping steps [105](#)
 - top level resource [96](#)
 - WAIT parameter [97](#)
- resource object
 - additional information [95](#)
 - general information [95](#)
 - overwriting commands for [99](#)
 - relations to [96](#)
 - relationships information [96](#)
 - request mapping to SA z/OS commands [97](#)
 - request priority [99](#)
 - requests issued to [96](#)
- restart of the automation adapter [87](#)
- resynchronization of the automation adapter [86](#)

S

- SA z/OS initialization member ING.DSIPARM (INGXINIT) [37](#)
- sample authorization user exit [58](#)
- sample master configuration file [46](#)
- security
 - authorization exit [57](#)
 - automation adapter [57](#)
 - JAAS configuration file [51](#)
 - Java classpath [89](#)
 - keystore and truststore for use by SSL [51](#)
 - keystore password [50](#)
 - NetView PIPE default safe [57](#)
 - password [57](#)
 - RACF profiles [58](#)
 - SSL configuration file [49](#)
 - user ID [57](#)
- setting
 - request priority for resource objects [99](#)
 - timeout with INGREQ_WAIT common global variable [76](#)
- shortcut keys [xiii](#)
- SMU server, initial contact by automation adapter [89](#)
- socket connection [89](#)
- socket, binding problem [75](#)
- SSI subsystem
 - stopping, recycling automation adapter after [76](#)
- SSL configuration file
 - certificate alias name [89](#)

- SSL configuration file (*continued*)
 - example [49](#)
 - Java system properties [49](#)
 - keystore file, password [89](#)
 - keystore filename [89](#)
 - missing parameters [49](#)
 - parameters [89](#)
 - truststore filename [89](#)
- SSL socket connection [89](#)
- start parameter, of ingadapter.sh start script [53](#)
- start-stop script
 - JAAS option [51](#)
 - using [53](#)
- status mapping
 - for problem SA z/OS compound status [106](#)
 - introduction [104](#)
 - rules [104](#)
 - steps [105](#)
- stop parameter, of ingadapter.sh start script [53](#)
- subplex, XCF group ID [48](#), [89](#)
- switching on and off EIF caching [89](#)
- System Automation Agent
 - customizing [37](#)

T

- TCP/IP bind socket problem [75](#)
- TIMEOUT parameter, description [89](#)
- TIMEOUT_SA_NOTAVAIL parameter, description [89](#)
- top level resource [96](#)
- trace log file [71](#)
- traceMIN parameter, of ingadapter.sh start script [53](#)
- traceOFF parameter, of ingadapter.sh start script [53](#)
- traceON parameter, of ingadapter.sh start script [53](#)
- truststore
 - creating for use by SSL [51](#)
- truststore file [49](#)
- truststore filename [89](#)

U

- unavailable, automation domain remaining after error recovery [76](#)
- UNIX System Services
 - automatic configuration [31](#)
 - console [53](#), [55](#)
 - Custom-Root directory [16](#)
 - customizing [41](#)
 - default file system [16](#)
 - Install-Root directory [16](#)
 - manual configuration [31](#)
 - prerequisites [41](#)
 - problems loading DLLs [76](#)
 - setting up environment variables [15](#)
 - shared file system [17](#)
 - writing to stdout and stderr files [76](#)
- user ID [57](#)
- USS [41](#)

W

- WAIT parameter, for SA z/OS commands [97](#)

- WebSphere authentication and end-to-end automation login module [51](#)
- WebUI
 - displays the resource data [93](#)

X

- XCF group ID [48](#), [89](#)

Z

- z/OS login module [51](#)



SC34-2750-02

