



End-to-End Automation Adapter

Note!

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

Fifth Edition (December 2006)

This edition applies to IBM Tivoli System Automation for z/OS Version 3, Release 1 (Program Number 5698-SA3), an IBM licensed program, and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

IBM welcomes your comments. A form for readers' comments appears at the back of this publication. If the form has been removed, address your comments to:

IBM Deutschland Entwicklung GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany

If you prefer to send comments electronically, use one of the following methods:

FAX (Germany): 07031 + 16-3456
FAX (Other Countries): (+49)+7031-16-3456
Internet: s390id@de.ibm.com

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

© Copyright International Business Machines Corporation 2005, 2006. All rights reserved.

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

Contents

Figures v

Tables vii

Notices ix

Web site disclaimer ix

Trademarks ix

About this document xi

Who should use this document xi

Where to find more information xi

 The end-to-end automation library xi

 The System Automation for z/OS Library xi

 Related Product Information xii

 Using LookAt to look up message explanations xii

Accessibility xiii

 Using assistive technologies xiii

 Keyboard navigation of the user interface xiii

 z/OS information xiii

Chapter 1. Introducing the end-to-end automation adapter 1

End-to-end automation management 1

The end-to-end automation adapter 2

 Purpose 2

 Communication 3

Chapter 2. Prerequisites and dependencies 5

Chapter 3. Preparing NetView 7

Enabling the event/automation service 7

 Starting the event/automation service 7

 Configuring the global initialization file 7

 Configuring the NetView message adapter service 8

Chapter 4. Installing the end-to-end automation adapter 9

Preparing to install the automation adapter 9

Installation steps 9

The hierarchical file system 9

 The default hierarchical file system 9

 User-defined HFS 10

Chapter 5. Customizing NetView and SA z/OS 15

Customize the NetView DSIPARM data set 15

Customize NetView tasks 15

 Automated operator functions 15

 Automation operator tasks 16

 Customize advanced automation options 16

Chapter 6. Customizing the end-to-end automation adapter 19

Configuration files 19

 The automation adapter master configuration file 19

 The automation adapter plug-in configuration file 21

 The SSL configuration file 23

 The JAAS configuration file 24

 The Java logger configuration file 25

Customizing USS 25

 Using the data directory for log files 25

 Specifying the Tivoli Common Directory for log files 25

 Setting UNIX access privileges 26

 Setting access privileges for TCP/IP connections 26

Customize the link list 27

Chapter 7. Security 29

Authentication 29

Authority checking by the automation adapter 29

 Authorization checking 29

Chapter 8. Operating the end-to-end automation adapter 33

Setting the PPI receiver ID for EAS 33

How to use the automation adapter 33

 USS script 33

 BPXBATCH 34

 SA z/OS defined resource 35

Chapter 9. Operating the end-to-end automation adapter via SA z/OS 37

Automation purpose and rules 37

 Non-disruptive recovery from errors 37

 Resynchronization of the automation adapter 38

 Recycling the automation adapter in the same system 38

 Restarting the automation adapter or failover to a backup system 38

 Automated applications 39

 Grouping considerations 39

Defining automation policy 40

 Application definition for the NetView event/automation service (E2E_EAS) 40

 Application definition for the end-to-end automation adapter (E2E_ADPT) 41

 MOVE group for the end-to-end automation adapters (E2E_ADPT_X) 47

Limitations to the automation 48

Chapter 10. Problem determination 49

Diagnosis of the SA z/OS tasks for the adapter 49

Problems with selecting the primary agent 49

Problems with the program-to-program interface 50

Problems with NetView event/automation service 50

Examples	51
Problems with consumer registration	51
Link list errors	52
TCP/IP bind socket problem	52
Timeout problems	53
USS problems	53
CHECK_HEALTH function failure.	53
Automation domain remaining offline after error recovery	53

Appendix A. Configuration file parameters 55

Appendix B. SA z/OS specific implementation of the end-to-end automation resource data model . . . 59

Resource data displayed by the operations console	59
Domain object	59
Resource object	63

Node	68
Requests issued by the operations console and the end-to-end automation manager	68
Overwriting commands	70
Request priority	71
Disregarding the hold flag	71
Definition of references for SA z/OS resources in an end-to-end automation policy	71
INGE2E	71
Status mapping	72
Introduction	72
Status mapping steps	73

Appendix C. Messages 77

End-to-end automation adapter messages	77
End-to-end automation adapter plug-in messages	85

Index 97

Figures

1.	Parts of the end-to-end automation management component	2	14.	INGEVOBS LIST command response	51
2.	End-to-end automation adapter communication	3	15.	Operations console topology tree	59
3.	Sample ingadapter.sh USS shell script.	11	16.	General information for an automation domain	60
4.	Sample automation adapter master configuration file, ing.adapter.properties	20	17.	Policy information for an automation domain	60
5.	TCP/IP port relationship for the end-to-end automation components	21	18.	Additional information for an automation domain, page 1	61
6.	Example automation adapter plug-in configuration file, ing.adapter.plugin.properties	22	19.	Additional information for an automation domain, page 2	62
7.	Example SSL configuration file, ing.adapter.ssl.properties	23	20.	General information for a resource object	64
8.	Sample JAAS configuration file, ing.adapter.jaas.properties.	24	21.	Additional information for a resource object	65
9.	Sysplex group relationships	39	22.	Forward relationships information for a resource object	66
10.	DISPAOPS command response	49	23.	Backward relationships information for a resource object	67
11.	INGAMS command dialog panel	50	24.	request list for a resource object.	67
12.	Event/automation service STATUS information	51	25.	General information for a node	68
13.	Event/automation service QSTATS information	51	26.	Example end-to-end automation resource reference	71

Tables

1.	End-to-end automation library	xi	6.	SSL configuration file	
2.	System Automation for z/OS Library	xii		(ing.adapter.ssl.properties) parameters	58
3.	Automation adapter advanced automation		7.	Mapping of end-to-end automation requests to	
	options	16		SA z/OS commands	69
4.	Master configuration file		8.	SA z/OS automation manager status mapping	73
	(ing.adapter.properties) parameters.	55	9.	Status mapping: second step if the SA z/OS	
5.	Plug-in configuration file			compound status reports a problem	74
	(ing.adapter.plugin.properties) parameters	57			

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM product, program, or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Deutschland Entwicklung GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Web site disclaimer

Any pointers in this publication to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement. IBM accepts no responsibility for the content or use of non-IBM Web sites specifically mentioned in this publication or accessed through an IBM Web site that is mentioned in this publication.

Trademarks

The following terms, used in this book, are trademarks of the IBM Corporation in the United States or other countries:

AIX	RACF
CICS	S/390
DB2	SP

IBM	Tivoli Enterprise Console
MVS	VSE/ESA
MVS/ESA	VTAM
NetView	WebSphere
OS/390	z/OS
Resource Link	zSeries

Java is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

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

UNIX is a registered trade mark in the United States and other countries, licensed exclusively through X/Open Company Limited.

About this document

This document describes the IBM® Tivoli® System Automation for z/OS™ end-to-end automation adapter (also called the automation adapter). The automation adapter enables SA z/OS to take advantage of the end-to-end automation that is provided by the IBM Tivoli System Automation for Multiplatforms. End-to-end automation provides continuous availability for heterogeneous distributed IT business applications and reduces the total cost of ownership.

Who should use this document

This document 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 end-to-end automation library

The following table shows the information units in the end-to-end automation library:

Table 1. End-to-end automation library

Title	Order Number
<i>System Automation for Multiplatforms End-to-End Automation Management</i>	SC33-8211

The System Automation for z/OS Library

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

Table 2. System Automation for z/OS Library

Title	Order Number
<i>IBM Tivoli System Automation for z/OS Planning and Installation</i>	SC33-8261
<i>IBM Tivoli System Automation for z/OS Customizing and Programming</i>	SC33-8260
<i>IBM Tivoli System Automation for z/OS Defining Automation Policy</i>	SC33-8262
<i>IBM Tivoli System Automation for z/OS User's Guide</i>	SC33-8263
<i>IBM Tivoli System Automation for z/OS Messages and Codes</i>	SC33-8264
<i>IBM Tivoli System Automation for z/OS Operator's Commands</i>	SC33-8265
<i>IBM Tivoli System Automation for z/OS Programmer's Reference</i>	SC33-8266
<i>IBM Tivoli System Automation for z/OS CICS Automation Programmer's Reference and Operator's Guide</i>	SC33-8267
<i>IBM Tivoli System Automation for z/OS IMS Automation Programmer's Reference and Operator's Guide</i>	SC33-8268
<i>IBM Tivoli System Automation for z/OS TWS Automation Programmer's Reference and Operator's Guide</i>	SC23-8269
<i>IBM Tivoli System Automation for z/OS End-to-End Automation Adapter</i>	SC33-8271

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

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

SA z/OS Home Page

For the latest news on SA z/OS, visit the SA z/OS home page at <http://www.ibm.com/servers/eserver/zseries/software/sa>

Related Product Information

You can find books in related product libraries that may be useful for support of the SA z/OS base program by visiting the z/OS Internet Library at <http://www.ibm.com/servers/eserver/zseries/zos/bkserv/>

Using LookAt to look up message explanations

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

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

- The Internet. You can access IBM message explanations directly from the LookAt Web site at <http://www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/>.
- Your z/OS TSO/E host system. You can install code on your z/OS or z/OS.e systems to access IBM message explanations using LookAt from a TSO/E command line (for example: TSO/E prompt, ISPF[®], or z/OS UNIX[®] System Services).
- Your Microsoft[®] Windows[®] workstation. You can install LookAt directly from the *z/OS Collection* (SK3T-4269) or the *z/OS and Software Products DVD Collection*

(SK3T4271) and use it from the resulting Windows graphical user interface (GUI). The command prompt (also known as the DOS > command line) version can still be used from the directory in which you install the Windows version of LookAt.

- Your wireless handheld device. You can use the LookAt Mobile Edition from <http://www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/lookatm.html> with a handheld device that has wireless access and an Internet browser (for example: Internet Explorer for Pocket PCs, Blazer or Eudora for Palm OS, or Opera for Linux handheld devices).

You can obtain code to install LookAt on your host system or Microsoft Windows workstation from:

- A CD-ROM in the *z/OS Collection* (SK3T-4269).
- The *z/OS and Software Products DVD Collection* (SK3T4271).
- The LookAt Web site (click **Download** and then select the platform, release, collection, and location that suit your needs). More information is available in the LOOKAT.ME files available during the download process.

Accessibility

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

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

Using assistive technologies

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

Keyboard navigation of the user interface

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

z/OS information

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

<http://www.ibm.com/servers/eserver/zseries/zos/bkserv/>

Chapter 1. Introducing the end-to-end automation adapter

This chapter provides a brief introduction to concepts and architecture of end-to-end automation management and the end-to-end automation adapter (also called the automation adapter in this document). For more detailed information, refer to *System Automation for Multiplatforms End-to-End Automation Management*.

End-to-end automation management

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

Figure 1 on page 2 shows the main parts that make up the end-to-end automation management component (the end-to-end automation manager and the operations console) and the relationship of the first-level domain to it.

The *operations console* is a Web-based graphical user front-end to the end-to-end automation domain and to the first level automation domains (resources managed by SA z/OS or SA for Multiplatforms, for example).

The *end-to-end automation manager* has a role equivalent to that of the automation manager in SA z/OS and consists of the following:

- Automation J2EE framework in WebSphere® Application Server (WAS)
- Automation engine
- Automation engine resource adapter
- First-level automation manager resource adapter (that communicates with the automation adapter)
- End-to-end automation policy

For details about these parts, see *System Automation for Multiplatforms End-to-End Automation Management*.

Note about terminology:

Terms that are used in discussing end-to-end automation have the following meaning in SA z/OS:

First-level automation domain

This is the group of SA z/OS agents and managers that belong to the *same XCF group ID* (GRPID).

Node This is the SA z/OS equivalent of a *system*.

Each first-level automation domain has an end-to-end automation adapter and a primary SA z/OS automation agent that together communicate with the end-to-end automation manager. The automation adapter and primary automation agent must run on the same system (that is, node) and must be linked to the sysplex group. The automation adapter can run on only one system in the sysplex.

The primary SA z/OS automation manager can run on any system in the sysplex, other than the system that the automation adapter is running on.

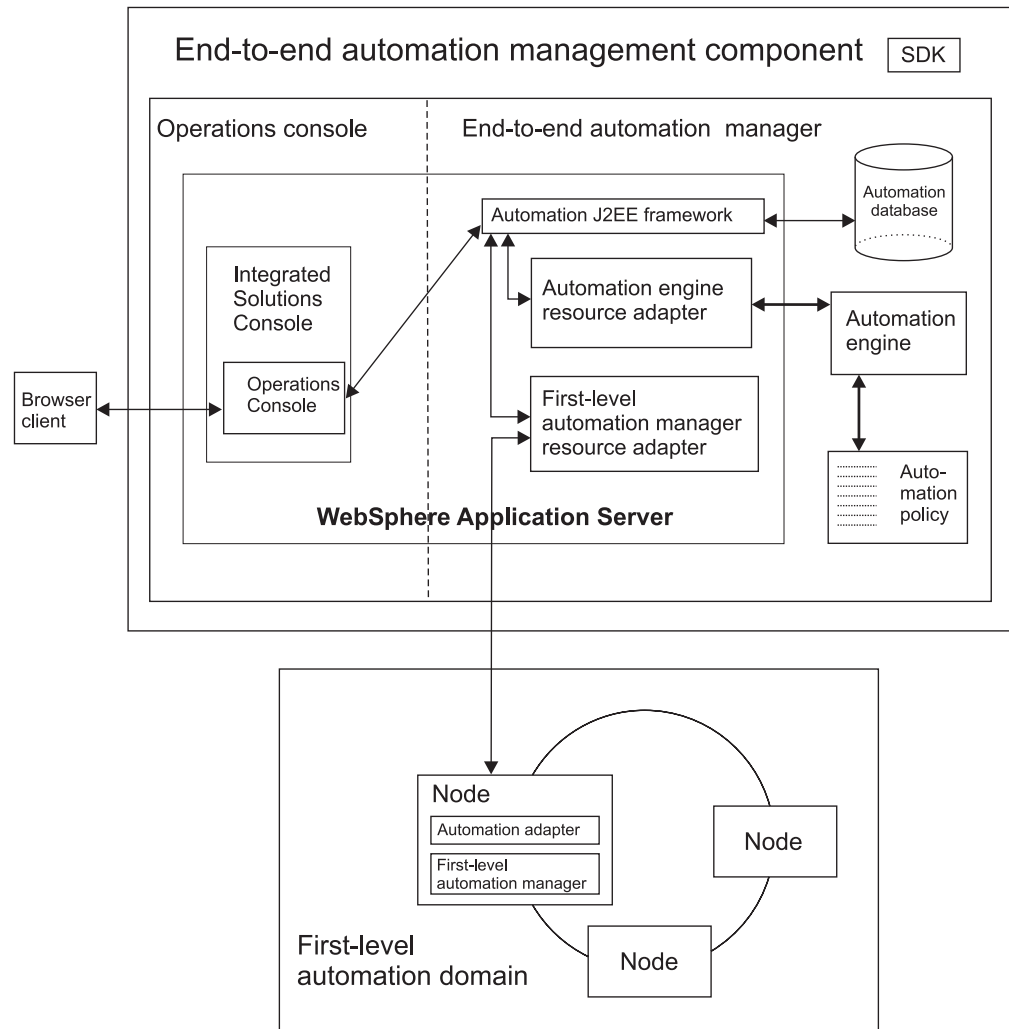


Figure 1. Parts of the end-to-end automation management component

The end-to-end automation adapter

The automation adapter acts as the link between the end-to-end automation manager and its first-level automation domain (that is, the SA z/OS sysplex group). You should note that the automation adapter can only run on one system in a domain, although you may have more than one domain in a sysplex. There can however be a maximum of only one automation adapter per z/OS system.

Note: The automation adapter must be located on the same system as the primary SA z/OS agent.

Purpose

The purpose of the automation adapter is to:

- Monitor resources within its first-level automation domain
- Propagate resource attribute changes to the end-to-end automation manager

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

Communication

The automation adapter can communicate either synchronously or asynchronously (as shown in Figure 2).

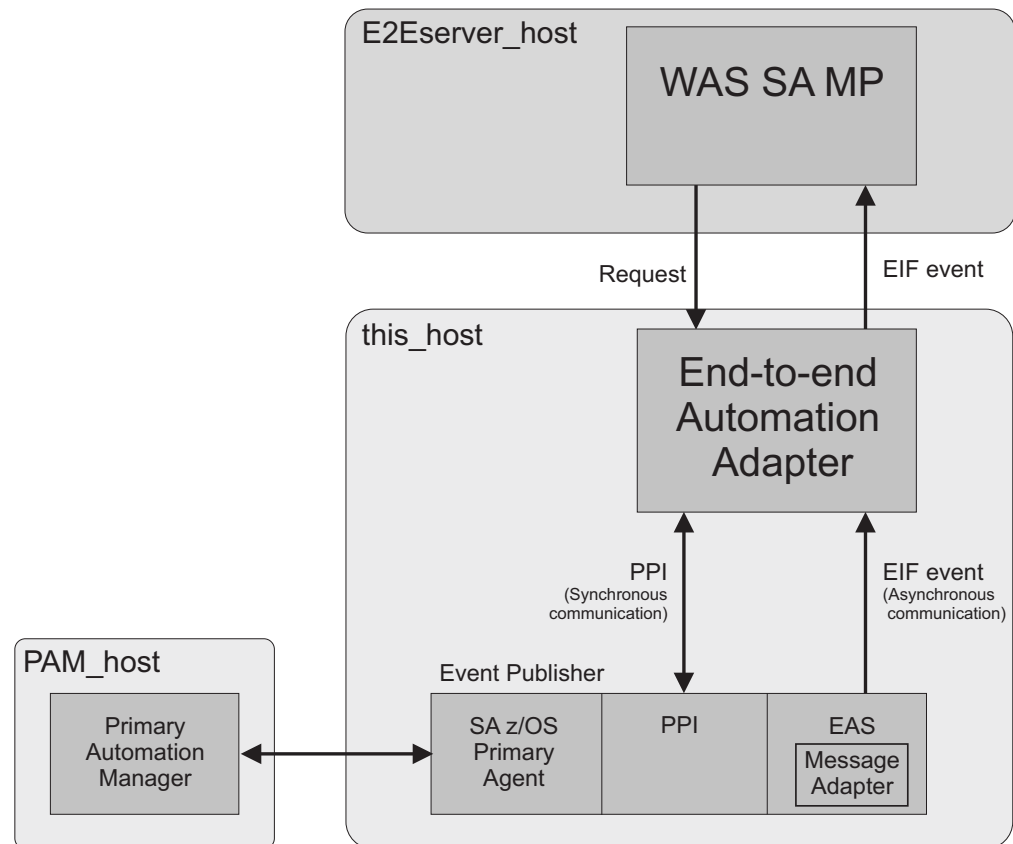


Figure 2. End-to-end automation adapter communication

Synchronous communication

The automation adapter schedules an *SA z/OS task execution request* via PPI synchronous communication (see Figure 2) to a task execution command handler that runs the automated operator function E2EOPER or E2EOPRnn.

The task execution request contains one or more end-to-end automation requests. The abbreviation for task execution request used in this document is *tex_request*.

The program-to-program interface: 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. It is an optional facility of the NetView subsystem address space and can be initialized with its PPIOPT start option.

The PPI option can only be requested for one subsystem address space of a system.

PPI is used for synchronous communication between the primary SA z/OS automation agent and the automation adapter.

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 to the automation adapter's EIF event receiver. This then delegates the events to the automation adapter's event mapping function. Once an event has been mapped and not rejected by the automation adapter it is sent to the end-to-end automation management server via the EIF emitter component.

For more information about EIF, refer to *Tivoli Enterprise Console Event Integration Facility Reference*.

The event/automation service: The message adapter service of the event/automation service (EAS) is used to convert and forward messages from the NetView message automation to a designated event server, such as the end-to-end automation adapter.

So that the automation adapter can receive events from SA z/OS, EAS registers with the NetView PPI. SA z/OS sends automation adapter-specific events over PPI to the message adapter service. The EAS message adapter service converts the messages into EIF events using the message adapter format file. The resulting events are forwarded to the automation adapter. See "Enabling the event/automation service" on page 7 for further details.

A separate EAS address space has to be started for each event destination. From NetView, the designated event destination is addressed by the unique PPI receiver name of the appropriate event/automation service.

The primary SA z/OS automation agent

The successful initialization of the automation adapter on a system makes the automation agent on that system into the *primary automation agent*. The primary automation agent enables the automation adapter to communicate with the primary SA z/OS automation manager (PAM).

The primary automation agent forwards events to the automation adapter after it has registered with the primary automation agent and subscribed for them.

After a system failure, event subscriptions may have been lost and so the automation adapter has to re-subscribe.

The primary automation manager sends events to the primary automation agent over XCF.

Note: There is a small chance that if two automation adapters are started in the same sysplex, 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. See Chapter 9, "Operating the end-to-end automation adapter via SA z/OS," on page 37 for more details.

Chapter 2. Prerequisites and dependencies

The following prerequisites are required for the automation adapter:

- SA z/OS 3.1
- IBM Tivoli NetView® 1.4 or higher
- z/OS 1.3 or higher
- Java™ Runtime Environment (JRE) 1.4.2 installed on z/OS (it is recommended that you use IBM Java)
Note: The automation adapter supports only the 32-bit version.
- The JRE Software Development Kit (SDK) if you choose to use the facility to create sample keys for test purposes (see “USS script” on page 33)

You need to configure the following components of these prerequisites:

- The Event/Automation Service (EAS) component of NetView. See the section “Enabling Event/Automation Service” in the NetView manual, *Installation: Configuring Additional Components* and “Enabling the event/automation service” on page 7
- Full z/OS UNIX System Services (USS) with hierarchical file system
- SSI address space with PPI function of NetView
- TCP/IP

Chapter 3. Preparing NetView

Because end-to-end automation uses events to communicate, you need to enable the event/automation service of NetView.

Enabling the event/automation service

The event/automation service (EAS) component of NetView and the steps to enable it are described in Chapter 7, "Setting Up UNIX System Services for the NetView Program," in *Tivoli NetView for z/OS Installation: Configuring Additional Components*.

This section only provides additional information about how to enable the NetView message adapter service of EAS for end-to-end automation.

End-to-end automation uses the NetView message adapter service of EAS as a gateway between the primary agent and the automation adapter. The NetView message adapter service of EAS converts incoming messages from Tivoli NetView into EIF events and forwards them to the automation adapter.

Starting the event/automation service

The event/automation service can be started either with a job from an MVS™ system console, or from a UNIX System Service command shell. In either case, startup parameters have to be provided in form of the following initialization files:

- Global initialization file (Default: IHSAINIT)
- Message adapter configuration file (Default: IHSAMCFG)

A sample for starting EAS as a job is located in NETVIEW.V5R1M0.SCNMUXMS as the member IHSAEVNT.

The initialization files are assumed to be located in a data set allocated to DD name IHSSMP3.

Perform the following updates to the sample to meet the requirements of your installation:

1. If you don't use the default name IHSAINIT for the global initialization file, pass the name of your file via the parameter INITFILE.
2. If you don't use the default name IHSAMCFG for the message adapter configuration file, pass the name of your file via the parameter MSGCFG.
3. In the DD statement, specify the data set names of your installation.

Configuring the global initialization file

To configure the global initialization file:

1. Make sure that the NetView message adapter service is also started when you start EAS. This is done by commenting out the statement:

```
NOSTART TASK=MESSAGEA
```

The other services are not needed by automation adapter, so prevent them from starting.

2. Specify INGEVE2E as the name of the PPI receiver ID in the following statement:

```
PPI=INGEVE2E
```

You can also pass the PPI receiver ID as a parameter when starting EAS.

If you want to use your own PPI receiver ID, you have to define it to SA z/OS as an advanced automation option before starting the automation adapter. See “Setting the PPI receiver ID for EAS” on page 33 for more details.

Configuring the NetView message adapter service

Configuration of the NetView message adapter service is done in the message adapter configuration file, as follows:

1. Provide the IP address or host name and, optionally, the port address of the automation adapter as the server location.
Specify the address 127.0.0.1 as the server host name.
2. Specify `connection_oriented` as the connection mode so that the connection is established at initialization time and closed at adapter termination time.
3. Switch off caching of events.
4. The default value of 4096 for the maximum event size can be used unchanged.
5. Specify the name of the NetView message adapter format file. The version of this file that is to be used by end-to-end automation is delivered in ING.SINGSAMP(INGMFMTE). Note that no configuration of this file is required.

If this is in its own data set, copy it to a data set that is concatenated to IHSSMP3.

Chapter 4. Installing the end-to-end automation adapter

This chapter describes the steps to follow in planning and preparing for the installation of the automation adapter.

Preparing to install the automation adapter

The automation adapter is provided in Java Archive (JAR) file format and requires Java Runtime Environment (JRE) to be installed on z/OS.

Installation steps

After the necessary prerequisites have been installed, you are now ready to install the automation adapter itself. The adapter is delivered as a PTF.

To install the automation adapter, carry out the following steps:

- Step 1. **Customize the hierarchical file system (HFS).** This is only necessary if you decide to use your own HFS or a shared HFS. For more details, see “User-defined HFS” on page 10.
- Step 2. **Customize the NetView DSIPARM data set.** See “Customize the NetView DSIPARM data set” on page 15.
- Step 3. **Customize NetView tasks.** See “Customize NetView tasks” on page 15.
- Step 4. **Customize the automation adapter configuration files.** The automation adapter includes a set of configuration files that you must manually configure for your system. For more details, see Chapter 6, “Customizing the end-to-end automation adapter,” on page 19.
- Step 5. **Customize USS.** This step is optional. It allows you to access message and trace logs in the Tivoli Common Directory. For more details, see “Customizing USS” on page 25.
- Step 6. **Define security.** This step is important for the authorization of each end-to-end automation request. The default is that every end-to-end automation request can start or stop SA z/OS resources. For more details about how to change the authorization, see Chapter 7, “Security,” on page 29.
- Step 7. **Enable high availability for the automation adapter.** You can define a high availability setup for the automation adapter. For more details, see Chapter 9, “Operating the end-to-end automation adapter via SA z/OS,” on page 37.

The hierarchical file system

The automation adapter is delivered with a default HFS, but you can also define your own HFS or a shared HFS (see “User-defined HFS” on page 10 for details).

The default hierarchical file system

The SMP/E installation creates a *default* file structure for the various files that are associated with the automation adapter, with the following directories:

```
1 /usr/lpp/ing/adapter
2                       /config
3                       /data
4                       /lib
5                       /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, the cache and log files, etc. This directory is initially empty.
- 4** JAR files and DLLs for the automation adapter
- 5** Security certificates. This directory is initially empty.

The default settings of all configuration files refer to the SA z/OS HFS subdirectories:

```
/usr/lpp/ing/adapter/config
/usr/lpp/ing/adapter/data
```

User-defined HFS

If you want to use your own HFS or a shared HFS you have to modify the start script and the configuration files.

It is expected that there is only one automation adapter per automation domain and at most one automation adapter per system.

Figure 3 on page 11 shows the `ingadapter.sh` shell script with the installation path and configuration files that are created by the SMP/E installation process.


```

#-----
# Customization begins here
#-----

# Root of the adapter installation
1  INSTALL_DIR=/usr/lpp/ing/adapter

# If needed, activate the steplib, replace HLQ by your high qualifier
# and add SINGMOD1 to the STEPLIB in order to be able to load IOC
# shipped by System Automation.
2  #export STEPLIB="HLQ.SINGMODx":$STEPLIB

# Password for the truststore generated by function GenerateSampleKeys
# Needed only if it is planned to generate sample keys.
3  SSL_PASSW=passphrase

# Redirection of stdout and stderr as specified in sample proc SAM(INGXADPT).
# Specify only filename without path.
4  REDIRSTDOUT=stdout.txt
4  REDIRSTDERR=stderr.txt

# Customize debugging DLLs (only if IBM Support requires you to do so)
5  export INGXDEBUG=1,2,1

#-----
# Additional customization needed if your installation differs from
# the default installation, e.g. different config or data directory
#-----
# Customize Java runtime.
# - path in order to invoke java runtime.
# - maximum memory allocated by JVM in MB (e.g. change 128M to 256M).
6  JAVA="java -Xmx128M"
# Java SDK. Customize path in order to invoke java keytool
6  JAVA_KEYTOOL=keytool

# Export DLLs and executables
INGINST_DLL=$INSTALL_DIR/lib
export LIBPATH=$INGINST_DLL:$LIBPATH
export PATH=$INGINST_DLL:$PATH

# Data directory must match eez-data-directory within ing.adapter.properties
# Directory access must be read-write
7  DATA_DIR=$INSTALL_DIR/data

# This statement sets the jlog output directory directly
# and skips the search for Tivoli Common Directory (TCD).
# Comment this statement out if you want to use the concept of
# Tivoli Common Directory.
# Using the concept of TCD the data directory must also match
# the log path in ing.adapter.jlogdir.properties
8  JLOG_OUTPUT=$DATA_DIR

# Used by function CreateSampleKeys in order to create and store
# the keystore and truststore files. See also ing.adapter.ssl.properties
# Directory access must be read-write otherwise keys remain in data directory
9  SSL_DIR=$INSTALL_DIR/ssl

# Configuration directory
10 CONFIG_DIR=$INSTALL_DIR/config

# Master config file for the automation adapter.
# The suffix will be added automatically if required.
11 ADAPTER_CONFIG=$CONFIG_DIR/ing.adapter${SUFFIX}.properties

# Java Logger configuration file name and
# file name that contains the path to the log-directory
12 JLOG_FILENAME_LOGPATH=ing.adapter.jlogdir.properties
13 JLOG_FILENAME=ing.adapter.jlog.properties
14 JLOG_CONFIG_DIR=$CONFIG_DIR

# Other configuration files
15 JAAS_CONFIG=$CONFIG_DIR/ing.adapter.jaas.properties
16 SSL_CONFIG=$CONFIG_DIR/ing.adapter.ssl.properties

#-----
# Customization section ends here.
#-----

```

Figure 3. Sample *ingadapter.sh* USS shell script

Where the path and files that you might need to customize are:

- 1 The automation adapter installation path.

- 2 If you need to load the IOC that is shipped by SA z/OS, activate the steplib and replace HLQ with your high level qualifier to add SINGMOD1 to the STEPLIB. If you prefer to add the SINGMOD1 library to your link list, you do not need to edit this line. See “Customize the link list” on page 27 for more details.
Note: If the steplib statement is not customized correctly you will receive the CEE3501S message that indicates that the INGIOC module has not been found.
- 3 The password for the truststore that is generated by the generateSampleKeys function. This is only needed if you plan to generate sample keys for test purposes.
- 4 Customize these parameters so that the filenames for the standard output and error streams are the same as those specified in the sample procedure INGSAMP(INGXADPT). Specify only the filename without the path.
- 5 The debugging setting for the DLLs. You only need to change this if asked to by IBM Support. For more details see Chapter 10, “Problem determination,” on page 49.
- 6 The Java run time and SDK paths to invoke Java and the Java keytool respectively. You can specify the maximum memory allocated by JVM in MB. Customize these paths if necessary.
- 7 The automation adapter data directory. This must match the eez-data-directory parameter in the file ing.adapter.properties.
- 8 This statement allows you to directly set the Java log output directory and skip the search for the Tivoli Common Directory (TCD). Comment this statement out if you want to use the Tivoli Common Directory. If you choose to use the TCD, the data directory must also match the log path in ing.adapter.jlogdir.properties. See “Customizing USS” on page 25.
- 9 The automation adapter SSL directory. This is used by the CreateSampleKeys function to create and store the keystore and truststore files. See also “The SSL configuration file” on page 23. The directory access must be read-write otherwise the keys remain in the data directory.
- 10 The automation adapter configuration directory
- 11 The automation adapter master configuration file, ing.adapter.properties, or ing.adaptersuffix.properties where *suffix* is a suffix that follows your naming conventions and is provided by the start script (see “USS script” on page 33 for details), for example, ing.adapterSYS1.properties.
- 12 The file name that contains the path to the log directory, the default is ./data
- 13 The Java logger configuration file name, ing.adapter.jlog.properties
- 14 Java logger configuration file path
- 15 Java Authentication and Authorization Service (JAAS) security configuration file, ing.adapter.jaas.properties
- 16 SSL security configuration file, ing.adapter.ssl.properties

To use your own HFS:

1. You must customize the automation adapter installation path in the start script.

Note: If you do not want to change the default file structure, you do not need to make any further changes. Proceed to Chapter 6, “Customizing the end-to-end automation adapter,” on page 19.

2. If you want to have your configuration files or log files in a *different location* (that is, you want to set up a different file structure), you must enter the fully qualified path name in the path variables and *whenever* a path is referred to in the configuration files, for example:
 - If the data directory is in a different path (say *mypath*), you must also change:
 - a. In the sample start script, *ingadapter.sh*,
`DATA_DIR= mypath/data`
 - b. In the master configuration file, *ing.adapter.properties*,
`eez-data-directory = mypath/data`
 - c. In the Java logger directory configuration file,
ing.adapter.jlogdir.properties,
`eez_output_dir = mypath/data`
 - If the configuration files are in a different path (say *mypath*), you must also change:
 - a. In the sample start script, *ingadapter.sh*,
`CONFIG_DIR= mypath/config`
 - b. In the master configuration file, *ing.adapter.properties*,
`plugin-configfile-sa4zos = mypath/config/ing.adapter.plugin.properties`
3. You must customize the *eezjlogdir.properties* and *eez_output_dir=* parameters to specify where the log files are written to. (For more details see “Customizing USS” on page 25.)

Although it is possible to customize the installation path and file names, throughout this document it is expected that the defaults are kept.

Shared HFS

The naming convention for the automation adapter master configuration file is
`ing.adaptersuffix.properties`

where *suffix* is the suffix specified as the second argument of the automation adapter start script. You may use the MVS system name as the suffix.

If you want to use a shared HFS and you do not use a virtual IP for the automation adapter remote contact IP address (see “IP address” on page 20) you need to create one automation adapter master configuration file per system of the first-level automation domain.

If you want to use a shared HFS and you use a virtual IP for the automation adapter remote contact IP address, you need only one automation adapter master configuration file for all systems.

If you use a shared HFS the automation adapter always writes messages and traces to the same log files regardless of which system it is currently running on.

Chapter 5. Customizing NetView and SA z/OS

This chapter provides details of how you should customize NetView and SA z/OS for the automation adapter.

Customize the NetView DSIPARM data set

The SA z/OS communication task INGPXDST has an initialization member DSIPARM (INGXINIT) that is used to specify automation adapter-specific parameters. Add the following parameters to this member:

```
PPI=YES  
1 PPIBQL=nnnn | 3000  
2 GRPID=xx
```

```
1 PPIBQL=nnnn | 3000
```

All input requests flow into this queue so the buffer queue limit, PPIBQL, should match this. If this 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 return code 1735:

```
INGX9820E JNI function ingjppi 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.

If you receive these error messages, increase the buffer queue limit.

Requests are lost but the end-to-end automation operator will receive exception reports. For more details see Chapter 10, "Problem determination," on page 49.

```
2 GRPID=xx
```

You must set the 2-byte XCF group ID of the SA z/OS subplex that you want to use. The value of the GRPID must be the same as in the automation adapter plug-in configuration file.

Customize NetView tasks

Automated operator functions

End-to-end automation uses dedicated automated operator functions in the primary agent to execute requests and to forward events. (You can find more details about the primary agent in Chapter 1, "Introducing the end-to-end automation adapter," on page 1.)

If defined, automated operator functions (E2EOPER and E2EOPR01 through E2EOPR*nm*) 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 will fail.

Use the customization dialogs to define the automated operator functions for the automation adapter in entry type Auto Operators of the SA z/OS automation

policy. Note that the names of the Automation Operators must match those that you define in the DSIPARM data set (see “Automation operator tasks”).

Automated operator function	Automation Operator
E2EOPER	AUTE2E
E2EOPR01 ... E2EOPR nn	AUTE2E01 ... AUTE2E nn
EVTOPER	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 19).

The definition of the automated operator functions E2EOPER, and E2EOPR01 through E2EOPR nn is optional but is recommended for performance reasons.

After defining the automated operator functions for end-to-end automation, rebuild the automation control files and refresh them for the automation manager and automation agent.

Automation operator tasks

The automation operator tasks are defined in the Tivoli NetView DSIPARM data set. Definition statements for these tasks are included in member `ING.SINGNPRM(AOFOPFSO)`. You must customize this sample for the automated operator functions that you have defined.

The sample AOFOPFSO contains the following NetView tasks for the E2EOPER automated operator functions:

- AUTE2E
- AUTE2E01
- AUTE2E02
- AUTE2E03

Customize advanced automation options

Advanced automation options can be used to change the default settings of the primary automation agents. For this purpose assign the user-defined value to the common global variable that is provided. Table 3 shows the common global variables that are provided as advanced automation options.

Table 3. Automation adapter advanced automation options

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.	“Setting the PPI receiver ID for EAS” on page 33
AOF_E2E_EVT_RETRY	1 to n	Specifies the number of retries, at intervals of one second, that are used to transfer events via PPI <code>TECROUTE</code> 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 37

Table 3. Automation adapter advanced automation options (continued)

Variable	Value	Effect	Further details
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.	
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 37
INGREQ_PRI.E2EMGR	Priority	Specifies the priority that incoming requests from the end-to-end automation manager are executed at. Default: LOW	"Request issued to resource objects" on page 67

For further details on advanced automation options and how to enable them, see the appendix "Global Variables" in *IBM Tivoli System Automation for z/OS Customizing and Programming*.

Chapter 6. Customizing the end-to-end automation adapter

Having installed the automation adapter files you will need to configure them for your setup. This chapter provides details of how you should do this.

The configuration steps that you need to carry out for the automation adapter are described in the following sections.

Configuration files

The following configuration files are delivered with the automation adapter:

- The automation adapter master configuration file, `ing.adapter.properties`
- The automation adapter plug-in configuration file, `ing.adapter.plugin.properties`
- The SSL configuration file, `ing.adapter.ssl.properties`
- The JAAS configuration file, `ing.adapter.jaas.properties`
- The Java logger configuration file, `ing.adapter.jlog.properties`

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

The automation adapter master configuration file

The automation adapter needs a master configuration file

```
./config/ing.{adaptersuffix|adapter}.properties
```

where *suffix* can be used to denote different systems if you have a shared HFS.

Without a properly verified master configuration file the automation adapter will not start. The name of this configuration file is defined in the USS shell script, `ingadapter.sh` (see “The automation adapter master configuration file definition” on page 12).

The master configuration file contains IP addresses and ports that might be different for each system within the sysplex. This means that if the automation adapter is moved to another system these parameters might be different. If you do not use a virtual IP address you need to make sure that the various master configuration files on the different systems within the sysplex contain the correct IP addresses.

If the configuration files are located on a shared HFS, you may need to define multiple configuration files. Use a suitable suffix as the locator. See “USS script” on page 33 for details.

Figure 4 on page 20 shows the sample automation adapter master configuration file that is delivered.

```

# --- Adapter Configuration -----

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

# --- EIF Configuration -----

   eif-cache                   = true
   eif-cache-size              = 500
   eif-retry-interval-seconds  = 30
4 eif-send-to-hostname       = E2Eserver_host
   eif-send-to-port            = 2002
   eif-receive-from-hostname   = 127.0.0.1
5 eif-receive-from-port     = 5529

# --- Plugin Configuration -----

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

```

Figure 4. 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 end-to-end automation server (WebSphere Application Server—WAS).

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 24.)

3 eez-data-directory

The directory that is to be used for the EIF cache and other adapter-specific data. This directory will be used for receiver and sender cache files and for automation adapter log files.

The default is the current working directory.

Note: If you change this parameter value, you must also change the `eez_output_dir=` parameter in the file `ing.adapter.jlogdir.properties` and the `DATA_DIR=` parameter in the start script, `ingadapter.sh`.

4 eif-send-to-hostname

The host name or IP address of the end-to-end automation server. Use the dotted format for IP_address.

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 plugin-configfile-sa4zos

This is the location of the plug-in configuration file. If your configuration

files are located on a non-default HFS, you must change this parameter so that it has the same path as the CONFIG_DIR= parameter in the start script, ingadapter.sh.

Table 4 on page 55 gives details of all the parameters that can be specified in the master configuration file.

Figure 5 shows the communication flow and port addresses that correspond to the automation adapter master configuration file in Figure 4 on page 20. The file IHSAMCFG is the NetView message adapter configuration member in data set SCNMUXCL (this data set is provided with NetView).

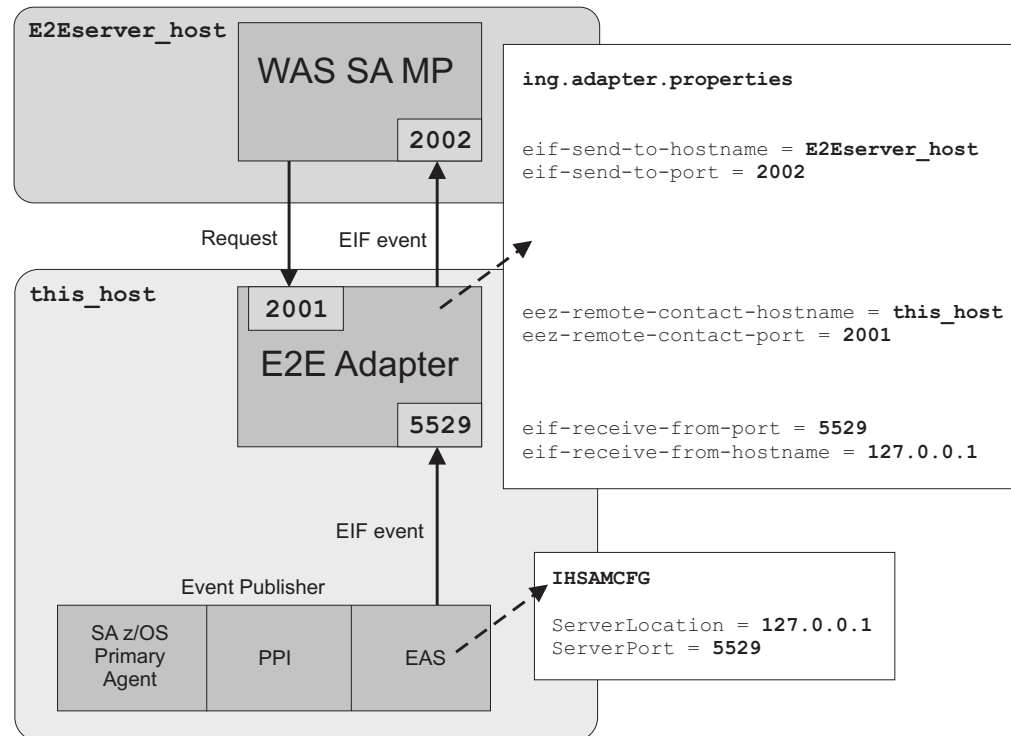


Figure 5. TCP/IP port relationship for the end-to-end automation components

Events are sent to the automation adapter after it has initialized and registered with the end-to-end automation manager. If the automation adapter stops, there is no communication flow so that events are not lost.

Synchronous communication between the primary SA z/OS automation agent and the automation adapter (for example, queries or requests to start or stop resources) takes place via PPI. Asynchronous communication takes place via the message adapter service of the NetView event/automation service using EIF events.

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

The automation adapter plug-in configuration file

The master configuration file refers to the plug-in configuration file, `ing.adapter.plugin.properties`. This file contains, for example, the Java class name of

the adapter plug-in application, or the default domain name. Without at least one proper plug-in configuration file the adapter will not start.

In the adapter master configuration file, the property `plugin-configfile-sa4zos` points to the configuration file of a specific plug-in application. This plug-in application configuration file must contain at least the properties that are shown in Figure 6.

Figure 6 shows an example of an automation adapter plug-in configuration file and highlights the entries that you must customize.

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

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

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

5 #plugin-domain-name = MyDomainName

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

Figure 6. 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. This 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 is 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 “Automated operator functions” on page 15.

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 time period that is specified, it detects that the SA z/OS NetView Agent is no longer available.

The automation adapter must receive two `CHECK_HEALTH` requests or other end-to-end automation server 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.

Table 5 on page 57 gives details about the parameters that you must include in your plug-in configuration file.

The SSL configuration file

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

Note:

The initial SSL directory is empty. You can create sample keys for use by SSL for test purposes. See “USS script” on page 33 for more details.

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. The automation adapter retrieves the name of the SSL configuration file from the system property `com.ibm.eez.adapter.configSSL` (this is set in the `ingadapter.sh` script).

If a parameter is not set in the SSL configuration file, the automation adapter start-stop script uses the corresponding (default) value from the following Java system properties:

- `-Djavax.net.ssl.trustStore`
- `-Djavax.net.ssl.keyStore`
- `-Djavax.net.ssl.keyStorePassword`
- `-Dcom.ibm.eez.adapter.ssl.keyStoreAlias`

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.

This plug-in application configuration file must contain at least the properties that are shown in Figure 7. Table 6 on page 58 gives details about these parameters.

Figure 7 shows an example of an SSL configuration file.

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

Figure 7. Example SSL configuration file, `ing.adapter.ssl.properties`

- 1** Note: These files are *not* delivered with the automation adapter but can be created for test purposes, see “USS script” on page 33 for more details.

When you create the sample SSL keys, the sample keystore for both the automation adapter and the end-to-end automation server are created. The generated truststore can be used for the automation adapter and the end-to-end automation server.

Copy the truststore and the keystore for the end-to-end automation server to the following directory on the WAS server:

- `etc/inst_root/cfg/ssl` for UNIX servers
- `inst_root/cfg/ssl` for Windows servers

Modify the appropriate end-to-end automation server SSL configuration file (`etc/inst_root/cfg/eez fla.ssl.properties` or `inst_root/cfg/eez fla.ssl.properties`) accordingly.

The JAAS configuration file

The JAAS definitions for the automation adapter are defined in the JAAS configuration file, `ing.adapter.jaas.properties`. This file specifies two entries for a login module that is required because all incoming requests will be authenticated using JAAS. Figure 8 shows a sample JAAS configuration file.

The start script for the automation adapter refers to the JAAS configuration file via the `JAAS_CONFIG=` parameter in the shell script, `ingadapter.sh` (see “JAAS security configuration file definition” on page 12).

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

Figure 8. Sample JAAS configuration file, `ing.adapter.jaas.properties`

The login modules are:

- 1 The end-to-end automation login module is provided for test purposes or when you can trust the end-to-end automation WebSphere authentication. It thus accepts all user IDs and passwords.
- 2 The z/OS login module rejects all users unless they are defined in RACF®.

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.

The entry `EEZAdapterDefaultLogin` uses the default automation adapter login module that accepts any user and any password. The entry `EEZAdapterLogin` uses the z/OS login module (RACF).

The z/OS administrator may overwrite the JAAS login module with any third party security login module.

The Java logger configuration file

The Java logger configuration file, `ing.adapter.jlog.properties`, defines the targets for the following log files:

- Two message log files (one in XML format, the other a flat file)
- One trace log file as a flat file, to be used by IBM Support
- One event log file as a flat file, to be used by IBM Support

The message log file can be accessed via the automation adapter graphical user interface (GUI).

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
```

Note: There is no need for you to configure the `ing.jlog.properties` file unless IBM Support asks you to change the trace level.

Customizing USS

Using the data directory for log files

You only need to specify the Tivoli Common Directory (TCD) if you comment out the `JLOG_OUTPUT` statement in the `ingadapter.sh` USS shell script (see “User-defined HFS” on page 10). This statement allows you to directly set the Java log output directory and skip the search for the TCD. The default is the automation adapter data directory.

Specifying the Tivoli Common Directory for log files

Message and trace logs for Tivoli products are located under a common parent directory called the Tivoli Common Directory (TCD). The TCD is defined in the properties file

```
/etc/ibm/tivoli/common/cfg/log.properties
```

This file contains the path to the Tivoli Common Directory via the entry `tivoli_common_dir=tivoli_common_dir`.

The common directory structure centralizes log files and first failure data capture files for all Tivoli products. The directory structure used is:

```
tivoli_common_dir/eez/logs  
tivoli_common_dir/eez/ffdc
```

If the automation adapter cannot access `/etc/ibm/tivoli/common/log.properties` it issues message `EEZU002I`. It then uses the log directory that is specified in the entry `eez_output_dir=./data`, which is defined in the file

```
/usr/lpp/ing/adapter/config/ing.adapter.jlogdir.properties
```

During the installation you can modify the entry `eez_output_dir` to your needs. For example, the entry `eez_output_dir=./data` generates the following log directory for the default installation directory structure:

```
/usr/lpp/ing/adapter/data/log  
/usr/lpp/ing/adapter/data/ffdc
```

If the properties file `log.properties` does not exist and the automation adapter has **write** access, it creates the `log.properties` file using the contents of `ing.adapter.jlogdir.properties`.

If the properties file `log.properties` exists and the automation adapter has **read** access, the contents of `ing.adapter.jlogdir.properties` will be overwritten by the TCD.

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:

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

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

Step 2. Define the RACF profile for the privileges:

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

Step 3. To grant READ access to the user OPER1:

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

Step 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

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
```

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

```
D PROG,LNKLST
```

Chapter 7. Security

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

Authentication

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

You must define user IDs and passwords with RACF. These define who is allowed to send an end-to-end automation request.

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

1. Define the JAAS logon module provided by RACF. (If you want to use RACF, the user IDs that you define are authorized.) This is equivalent to setting the parameter `eez-operator-authentication` to `true` in the master configuration file, `ing.adapter.properties`.
2. Define the JAAS logon module provided by `EEZAdapterDefaultLogin` that always validates as `true`. This is equivalent to setting the parameter `eez-operator-authentication` to `false`.
3. Define the JAAS logon module provided by any third party product.

Authority checking by the automation adapter

Authorization checking

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 request handler performs the authorization check for each command that is to be executed, except for query requests. The authorization check is to be done by an authorization user exit, which is an external REXX program that you must provide.

The name of the REXX program must be `AOFEXE2E`. A sample of this 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

For further details about the command parameters, see “Requests issued by the operations console and the end-to-end automation manager” on page 68.

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_UN SOLICITED_REQUEST FAILED: ERROR=5
SERVICE='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 an RACF checking routine that uses the following RACF FACILITY class and the profile pattern to check whether the user has ALTER access to it:

```
FACILITY.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 for 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 FACILITY.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 with the authority to ALTER the resource, the user is allowed to execute the command for the resource *resourceName.resourceType.resourceSystem*.

Examples of profiles:

FACILITY. INGE2E. INGREQ. DB2. APL. KEY1

The user is allowed to start or stop application DB2[®] on system KEY1.

FACILITY. INGE2E. INGREQ. DB2. **

The user is allowed to start or stop DB2 resources of type application or group on all systems.

FACILITY. INGE2E. INGREQ. **

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

Chapter 8. Operating the end-to-end automation adapter

This chapter provides details of how to operate the automation adapter manually.

Setting the PPI receiver ID for EAS

If the event/automation service (EAS) does not use the default PPI receiver, INGEVE2E, use the advanced automation option to change it, before starting the automation adapter. To do this, assign the chosen PPI receiver ID as a value to the common global variable AOF_E2E_EAS_PPI.

How to use the automation adapter

There are three ways that you can operate the automation adapter:

- With a USS script from the USS console, see “USS script”
- As a batch job or started task, see “BPXBATCH” on page 34
- As a defined resource in SA z/OS, see “SA z/OS defined resource” on page 35

USS script

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 26.
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 26.
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 26.
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. This can be done for modules of the HFS by setting the extended attribute `p`. Use the `extattr` shell command with the `+p` option for all libraries (*.so) in the automation adapter data directory. You may also have to define LE-provided run time libraries, such as CEERUN2, 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. For example, if you use the script as a batch job, it would be similar to the following:

```
//INGXADPT PROC F=START,SX=  
//ADAPTER EXEC PGM=BPXBATCH,REGION=0M,  
//          PARM='SH /usr/lpp/ing/adapter/ingadapter.sh &F &SX'  
//STDOUT DD PATH='/usr/lpp/ing/adapter/data/stdout.txt',
```

```
//      PATHOPTS=(OWRONLY,OCREAT),PATHMODE=SIRWXU
//STDERR DD PATH='/usr/lpp/ing/adapter/data/stderr.txt',
//      PATHOPTS=(OWRONLY,OCREAT),PATHMODE=SIRWXU
```

See also “User-defined HFS” on page 10.

Syntax

You can start the automation adapter with a USS script that SA z/OS provides:

```
ingadapter.sh function suffix
```

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

This stops the automation adapter.

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.

traceON | TRACEON

Starts the maximum trace log service.

traceOFF | TRACEOFF

Stops the trace log service.

traceMIN | TRACEMIN

Starts the minimum trace log service.

suffix This parameter can be omitted. It might be needed to make the automation adapter master configuration files unique for each system within the sysplex, for example, if you have a shared HFS (see “Shared HFS” on page 13). The default names of the configuration files will be used, optionally with *suffix* as a suffix.

If you want to start in the USS shell, you can use the nohup command to put the startup in the background, for example:

```
nohup ingadapter.sh start &
```

BPXBATCH

There is a sample procedure, INGXADPT, 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.

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

SA z/OS defined resource

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 Chapter 9, “Operating the end-to-end automation adapter via SA z/OS,” on page 37 for further details.

Chapter 9. Operating the end-to-end automation adapter via SA z/OS

This chapter describes how to define the automation adapter as application to be automated by SA z/OS for simplified operating and high availability.

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”
- “Resynchronization of the automation adapter” on page 38
- “Recycling the automation adapter in the same system” on page 38
- “Restarting the automation adapter or failover to a backup system” on page 38

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 about the offline automation domain. 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_TKOVROTIMEOUT.

Recycling the automation adapter in 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
- ACF COLD
- An outage of NetView event/automation service, if events have been lost
- 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

The automation solution for the end-to-end automation 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 automation adapters on the different systems in the subplex. To prevent a quick move of the automation adapter from one system to another, a Makeavailable/WhenObservedDown relation from the automation adapter to the MOVE group is defined, as shown in Figure 9. Thus the automation adapter is not started on a selected system until it is down on all other systems that are included in the MOVE group.

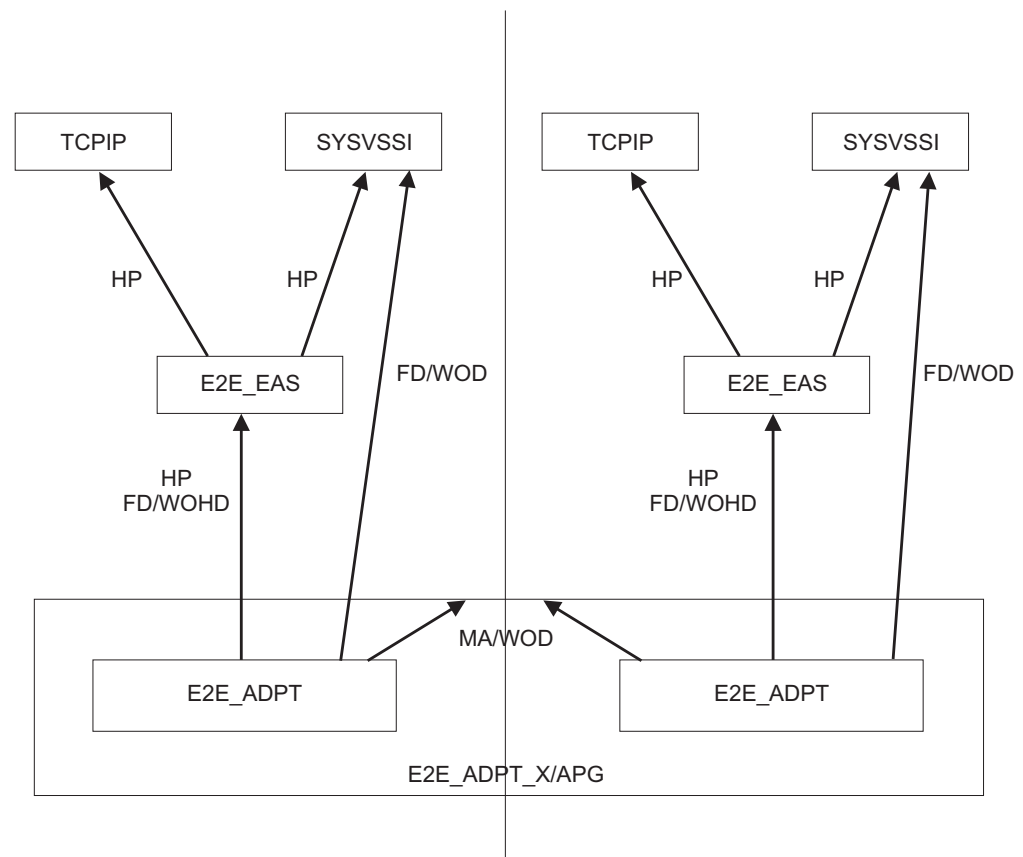


Figure 9. Sysplex group relationships

The following abbreviations for relationship types are used:

HP	HASPARENT
FD/WOHD	FORCEDOWN/WhenObservedHardDown
FD/WOD	FORCEDOWN/WhenObservedDown
MA/WOD	MAKEAVAILABLE/WhenObservedDown

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.

Application definition for the NetView event/automation service (E2E_EAS)

Define an application for the NetView event/automation service for each system in the sysplex using the description of the policy items in the SA z/OS automation policy that follows.

Policy item DESCRIPTION

Specify an entry name of your choice and a description of the application.

Field name	Specified value
Entry Name	E2E_EAS
Object Type	INSTANCE
Short Description	Event Automation Service
Long Description	Event Automation Service

Policy item APPLICATION INFO

Specify subsystem name, job name and JCL procedure name of your choice.

Field name	Specified value
Application Type	STANDARD
Subsystem Name	E2E_EAS
Job Name	E2E#EAS
JCL Procedure Name	IHSAEVNT
Restart Option *	ALWAYS
* This option can be omitted, if the default value has not been changed.	

Policy item RELATIONSHIPS

Define HASPARENT relationships from the E2E_EAS 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.

Field name	Specified value
Description	E2E_EAS depends on TCPIP
Relationship Type	HASPARENT
Supporting Resource	TCPIP/APL/=

Field name	Specified value
Description	E2E_EAS depends on SYSVSSI
Relationship Type	HASPARENT

Field name	Specified value
Supporting Resource	SYSVSSI/APL/=

Policy item STARTUP

The event/automation service can be started with the default start up command of SA z/OS for applications. Therefore no definitions to this policy item are needed.

Policy item SHUTDOWN

Specify shutdown commands for shutdown phase NORM.

Pass	Automated Function/*	Command text
1		MVS P &SUBSJOB
4		MVS C &SUBSJOB

The shutdown command to PASS1 causes the E2E#EAS job to halt all activity and end normally. The shutdown command to PASS4 ends the started task immediately.

Policy item THRESHOLDS

Specify a critical threshold to limit the number of times that the restart in the same system of the event/automation service is attempted. The critical threshold has to be specified as the number of times the error must occur within a certain time interval.

Policy item WHERE USED

Select an appropriate basic group, with or without an automation name, to link the application to all the systems in the sysplex.

Application definition for the end-to-end automation adapter (E2E_ADPT)

Define an application for the automation adapter for each system in the sysplex using the description of the policy items in the SA z/OS automation policy that follows.

Policy item DESCRIPTION

Specify an entry name of your choice and a description of the application.

Field name	Specified value
Entry Name	E2E_ADPT
Object Type	INSTANCE
Short Description	E2E Automation Adapter
Long Description	E2E Automation Adapter

Policy item APPLICATION INFO

Define the automation adapter as an application of job type MVS to be started by a procedure. Specify subsystem name, job name and JCL procedure name of your choice.

Field name	Specified value
Application Type	STANDARD

Field name	Specified value
Subsystem Name	E2E_ADPT
Job Name	E2E#ADPT
JCL Procedure Name	INGXADPT
Start Timeout	00:05:00
Restart Option *	ALWAYS
* This option can be omitted, if the default value has not been changed.	

If the parameter `eez-initial-contact` in the master configuration file of the automation adapter is set to true, so that the automation adapter tries to establish initial contact with the end-to-end automation server during startup, specify a value for Start Timeout that is consistent with the parameter `eez-initial-contact-retry-interval-minutes` in the master configuration file of the automation adapter.

Start Timeout less than `eez-initial-contact-retry-interval-minutes`: Specify a value for Start Timeout that is less than parameter `eez-initial-contact-retry-interval-minutes`. The agent status of the automation adapter changes to `STARTED2` after the Start Timeout has elapsed while the automation adapter was still trying to establish initial contact with the end-to-end automation server.

As soon as contact is acknowledged by the end-to-end automation server and the automation adapter successfully completes its initialization, the status changes to `UP`.

If no contact can be established within the time that is set in `eez-initial-contact-retry-interval-minutes`, the automation adapter stops itself and will be restarted by SA z/OS as long as the critical threshold for failed restarts has not yet been exceeded.

The advantage of specifying a Start Timeout value less than `eez-initial-contact-retry-interval-minutes` is to indicate that there is a problem when the specified Start Timeout interval to establish the initial contact has nearly elapsed.

Start Timeout greater than `eez-initial-contact-retry-interval-minutes`: If the value for Start Timeout is greater than parameter `eez-initial-contact-retry-interval-minutes`, the automation adapter might stop before the agent status has indicated a problem.

Policy item RELATIONSHIPS

Define relationships starting from the automation adapter to different supporting resources.

The `HASPARENT` relationship from the automation adapter to `E2E_EAS` enforces the right start or stop order, because the automation adapter depends on the EAS message adapter service.

Field name	Specified value
Description	Automation Adapter depends on E2E_EAS
Relationship Type	HASPARENT
Supporting Resource	E2E_EAS/APL/=

The FORCEDOWN relationship from the automation adapter to E2E_EAS enforces a shutdown of the automation adapter if the EAS resource fails and cannot be restarted on the same system.

Field name	Specified value
Description	Force down adapter when E2E_EAS is harddown
Relationship Type	FORCEDOWN
Supporting Resource	E2E_EAS/APL/=
Condition	WhenObservedHardDown

The FORCEDOWN relationship from the automation adapter to the NetView SSI enforces 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.

Field name	Specified value
Description	Force down adapter when SYSVSSI is down
Relationship Type	FORCEDOWN
Supporting Resource	SYSVSSI/APL/=
Condition	WhenObservedDown

The MAKEAVAILABLE relationship from the automation adapter to the MOVE group prevents a quick move of the automation adapter. It makes sure that the automation adapter is not started on any system in the sysplex until it is down on all systems in the sysplex.

Field name	Specified value
Description	Prevent quick move of adapter
Relationship Type	MAKEAVAILABLE
Supporting Resource	E2E_ADPT_X/APG
Condition	WhenObservedDown

Policy item MESSAGES/USERDATA

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=ADAPTER, F=START
IEF695I START INGXADPT WITH JOBNAME E2E#ADPT IS ASSIGNED TO USER STCUSER ,
GROUP STCGROUP
\$HASP373 E2E#ADPT STARTED
- 2** IEF403I E2E#ADPT - STARTED - TIME=17.23.34
\$HASP373 BPXAS STARTED
IEF403I BPXAS - STARTED - TIME=17.23.35
- 3** EEZA0100I The adapter has been started
- 4** EEZA0056I Initial contact was enabled and the connection to the management
server has been established
- 5** 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.jlog.properties from path /u/aoc/310/instance/config.
INGX9902I INGXPluginLogger has successfully been initialized using configuration
file ing.adapter.jlog.properties from path /u/aoc/310/instance/config.
6 EEZA0112I The plug-in has been started: class com.ibm.ing.sam.INGXPlugin
7 EEZA0102I The adapter is ready

```

- 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** The first message that is issued by the automation adapter is also used to change the agent status of the automation adapter to ACTIVE.
- 4** If the eez-initial-contact parameter in the automation adapter master configuration file is set to true, the automation adapter tries to establish initial contact with the end-to-end automation server after it has issued the previous message EEZA0100I. Message EEZA0056I informs that the connection has been established successfully.
- 5** The automation adapter has successfully established the connection to the management server and now continues initialization by starting the plug-in.
- 6** Message EEZA0112I indicates that the plug-in has successfully initialized the domain. It is used to change the agent status of the automation adapter to UP.
- 7** Message EEZA0102I indicates that the automation adapter startup sequence is complete.

If the startup of the automation adapter or the plug-in fails, the automation adapter stops after having issued 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 a non-recoverable abend and automation will not try to restart the automation adapter.

During a failed startup of the automation adapter the following messages may be issued, for example:

```

1 S INGXADPT,JOBNAME=E2E#ADPT,F=START
  $HASP100 E2E#ADPT ON STCINRDR
  IEF695I START INGXADPT WITH JOBNAME E2E#ADPT IS ASSIGNED TO USER STCUSER ,
  GROUP STCGROUP
  $HASP373 E2E#ADPT STARTED
2 IEF403I E2E#ADPT - STARTED - TIME=19.26.31
3 EEZA0100I The adapter has been started
4 EEZA0057E The connection to the management server cannot be established
5 EEZA0103I The adapter is stopping
6 EEZA0105I The adapter has been stopped due to a failure, rc=12
  AOF571I 19:26:56 : ADAPTER SUBSYSTEM STATUS FOR JOB ADAPTER IS ABENDING - SUBSYSTEM
  HAS SUFFERED A RECOVERABLE ERROR
7 IEF404I ADAPTER - ENDED - TIME=19.26.56
  $HASP395 ADAPTER ENDED

```

- 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** The first message that is issued by the automation adapter is also used to change the agent status of the automation adapter to ACTIVE.
- 4** If the eez-initial-contact parameter in the automation adapter master configuration file is set to true, the automation adapter tries to establish

initial contact with the end-to-end automation server after having issued message EEZA0100I. Message EEZA0057E informs about the elapsed timeout interval that is provided to establish initial contact with the end-to-end automation server

- 5** The automation adapter stops itself. Message EEZA0103I is used as a status message to change the status of the automation adapter to TERMINATING.
- 6** Message EEZA0105I indicates the error situation and is used to change the status of the automation adapter to ABENDED.
- 7** 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

If you do not use easy message management for the automated build of the automation table based on these definitions, additional statements have to be specified in your automation table to call the generic routines ACTIVMSG and TERMMMSG with the appropriate parameters to update the resource status.

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.

To define messages to be captured, enter the message IDs and select Capture after having entered action AUTO.

Useful messages in analyzing problems are:

EEZA0037E EEZA0038E EEZA0055E EEZA0056I EEZA0057E EEZA0061E EEZA0101I
 EEZA0102I EEZA0111I EEZA0113I EEZA0114I EEZA9991E EEZA9992E
 INGX9705E INGX9707I INGX9709E INGX9711E

If you do not use easy message management for the automated build of the automation table based on these definitions, additional statements have to be specified in your automation table to call the generic routine AOFCPMSG to capture these messages.

Defining status commands: Define the following status commands to be executed when the status of the automation adapter changes to the specified value.

Enter the resource status as the message ID and specify the command after having entered CMD as the action.

Message ID	Pass/Selection	Automated Function/'**'	Command Text
IEF450I			INGE2E RESET
ACORESTART			INGE2E RECYCLE,E2E_ADPT_X/APG

The command for status BROKEN cleans up the primary agent to enable a new automation adapter initialization.

The command to ACORESTART enforces a recycle of the MOVE group. This results in a recycle in the same system of the automation adapter, after a recycle of the primary agent or after an ACF load.

Defining code processing: The following code processing definition adjusts the severity of captured messages.

Messages that indicate normal status changes during startup and shutdown are displayed with severity NORMAL, others with severity UNUSUAL.

Enter the message ID CAPMSGs and specify the codes and values after having entered action CODE.

Message ID	Code 1	Code 2	Code 3	Value Returned
CAPMSGs	EEZA0037E	*	*	UNUSUAL
	EEZA0038E	*	*	UNUSUAL
	EEZA0055E	*	*	UNUSUAL
	EEZA0057E	*	*	UNUSUAL
	EEZA0061E	*	*	UNUSUAL
	EEZA9991E	*	*	UNUSUAL
	EEZA9992E	*	*	UNUSUAL
	IEF450I	*	*	UNUSUAL
	INGX9705E	*	*	UNUSUAL
	INGX9707I	*	*	UNUSUAL
	INGX9709E	*	*	UNUSUAL
	INGX9711E	*	*	UNUSUAL
	EEZA0056I	*	*	NORMAL
	EEZA0101I	*	*	NORMAL
	EEZA0102I	*	*	NORMAL
	EEZA0111I	*	*	NORMAL
	EEZA0113I	*	*	NORMAL
	EEZA0114I	*	*	NORMAL

Policy item STARTUP

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.

Startup Parameters	,F=START,SX=&SYSNAME
--------------------	----------------------

The SX parameter is optional and might be needed if the system name has to be passed as a suffix parameter to the USS script ingadapter.sh.

Policy item SHUTDOWN

Specify shutdown commands for the different shutdown phases.

The start procedure of the automation adapter is also used to stop the automation adapter. To prevent confusing start or stop messages of the procedure when stopping the automation adapter, specify a different job name. If you have customized the sample procedure to accept further parameters, append them to the START command of the procedure.

For shutdown phase NORM specify:

Pass	Automated Function/*	Command text
1		MVS S &SUBSPROC,JOBNAME=ADPTSTOP,F=STOP
4		MVS C &SUBSJOB

For shutdown phase IMMED specify:

Pass	Automated Function/*	Command text
1		MVS S &SUBSPROC,JOBNAME=ADPTSTOP,F=STOP
2		MVS C &SUBSJOB

For shutdown phase FORCE specify:

Pass	Automated Function/*	Command text
1		MVS C &SUBSJOB

Policy item THRESHOLDS

Specify a critical threshold to limit the number of times that the restart in the same system of the automation adapter is attempted. The critical threshold has to be specified as the number of times the error must occur within a certain time interval.

Policy item WHERE USED

Select the sysplex group E2E_ADPT_X/APG.

MOVE group for the end-to-end automation adapters (E2E_ADPT_X)

Define an application group for the automation adapters on the different systems in the sysplex according to the following description.

Policy item DESCRIPTION

Specify an entry name of your choice and a description of the MOVE group.

Field name	Specified value
Entry Name	E2E_ADPT_X
Short Description	MOVE group for E2E Automation Adapter
Long Description	MOVE group for E2E Automation Adapter

Policy item APPLGROUP INFO

Define the application group as MOVE group belonging to a sysplex.

Field name	Specified value
Application Group Type	SYSPLEX
Nature	MOVE
Default Preference	*DEF
Automation Name	E2E_ADPT_X
Behaviour	ACTIVE

Policy item APPLICATIONS

Select E2E_ADPT as application members for the group.

Policy item RESOURCES

The default initial preference values can be taken unchanged.

Policy item WHERE USED

Select the sysplex to link the MOVE group.

Limitations to the automation

1. If the EAS message adapter is stopped separately with the command F E2E#EAS,STOP TASK=MESSAGEA, events might be lost unnoticed by the automation because EAS still receives incoming events and discards them.
To prevent losing events in this way, always stop the message adapter by stopping E2E_EAS. The automation then notices the outage and tries to make the E2E_EAS 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.

Chapter 10. Problem determination

This chapter describes methods for determining and solving various problems that you might encounter. It includes descriptions of methods to analyze:

- Problems with selecting the primary agent
- Problems with the program-to-program interface
- Problems with NetView event/automation service
- Problems with consumer registration
- Link list errors
- TCP/IP bind socket problem
- “Timeout problems” on page 53
- “USS problems” on page 53
- “CHECK_HEALTH function failure” on page 53
- “Automation domain remaining offline after error recovery” on page 53

Diagnosis of the SA z/OS tasks for the adapter

If initialization of the domain fails (and message AOF233E is issued), you can check the existence and status of the automated operator function EVTOPER with the command DISPAOPS, which will produce output similar to that shown in Figure 10.

```
A0FK2S0          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      E2EOPR01     AUTE2E01     ACTIV
KEY4      E2EOPR02     AUTE2E02     ACTIV
KEY4      E2EOPR03     AUTE2E03     ACTIV

Command ==>
PF1=Help      PF2=End      PF3=Return      PF6=Ro11
                PF8=Forward  PF9=Refresh      PF12=Retrieve
```

Figure 10. DISPAOPS command response

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 11 on page 50. 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 = 10/01/04
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
-----
-   KEY4    KEY4$$$$3 PAM   READY   KEY1PLEX  INGXS634  V2R4M0  XCF
-   KEY4    KEY4      AGENT READY   KEY1PLEX  INGXS634  V2R4M0  XCF  YES

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

```

Figure 11. 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 procname,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.

For more details concerning the information that is displayed see *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 12.

```
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 12. Event/automation service STATUS information

2. At the NetView console entering:

```
MVS F NETEVE2E,DISPLAY,QSTATS
```

displays information as shown in Figure 13. 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 ALRTTRAP 0 0 0
```

Figure 13. 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 14.

```
Consumer Events Exit AUTOOPER User data
-----
E2E 1,2 INGRTECS INGEVE2E
*** End of Display ***
```

Figure 14. 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 "Setting the PPI receiver ID for EAS" on page 33.

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 end-to-end automation server 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 may occur at the operations console if you display resource data by selecting an SA z/OS domain in the topology tree and navigating through the resource tree. 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:
The GUI or the end-to-end automation manager	The timeout interval that is defined in the end-to-end automation server's WebSphere Admin Console for the environment variable <code>com.ibm.eez.aab.invocation-timeout-seconds</code>
The automation adapter (such as <code>INIT_DOMAIN</code>)	The timeout interval that is defined in the automation adapter <code>ing.adapter.plugin.properties</code> 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" on page 69.

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.

CHECK_HEALTH function failure

When you stop NetView or the SSI Subsystem you should recycle the automation adapter. See the description of `ACORESTART` in "Defining status commands" on page 45.

If you do not recycle the automation adapter, any `CheckHealth` command that is periodically received from the end-to-end automation server will result in error messages `INGX9905` and `INGX9820`. This may cause many error messages to be written to the `SYSLOG`.

Automation domain remaining offline 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 offline automation domain. If there are problems notifying the

end-to-end automation manager about the recovered error situation, the domain may remain offline rather than become online.

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.

Appendix A. Configuration file parameters

The following tables provide details about all of the configuration file parameters.

Table 4. Master configuration file (*ing.adapter.properties*) parameters. In alphabetical order.

Parameter Name	Default	Description
eez-data-directory	./data	The directory that is to be used for the EIF cache and other automation adapter-specific data. This directory will be used for receiver and sender cache files and for automation adapter log files. The default is the current working directory.
eez-initial-contact ¹	true	Value: true or false. If this parameter is set to true, the automation adapter tries to establish initial contact with the end-to-end automation management server.
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 end-to-end automation management 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, <i>ing.adapter.jaas.properties</i> , 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>

Table 4. Master configuration file (*ing.adapter.properties*) parameters (continued). In alphabetical order.

Parameter Name	Default	Description
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.
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	none	The host name or IP address that is used by the automation adapter to receive synchronous requests from the end-to-end automation server (WebSphere Application Server—WAS).
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 end-to-end automation server.
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 4. Master configuration file (*ing.adapter.properties*) parameters (continued). In alphabetical order.

Parameter Name	Default	Description
eif-send-to-hostname	E2Eserver_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.
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 5 lists only those parameters that you can update.

Table 5. 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, etc. The name of the code page must match the code page names supported by java.io See for example http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html

Table 5. Plug-in configuration file (*ing.adapter.plugin.properties*) parameters (continued)

Parameter Name	Default	Description
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 SA for Multiplatforms.</p> <p>If you enter ? or omit the parameter, SA z/OS substitutes a name that consists of the sysplex name and the XCF group ID. You can however use your own naming strategy.</p>

Table 6. 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 -Djavax.net.ssl.trustStore</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 -Djavax.net.ssl.keyStore</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 -Djavax.net.ssl.keyStorePassword</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 7, "Security," on page 29.</p>

Appendix B. SA z/OS specific implementation of the end-to-end automation resource data model

This chapter describes the SA z/OS specific implementation of the end-to-end automation resource data model.

Note: The screen captures in this appendix are for illustration only and may not correspond exactly to the operations console for your installation.

Resource data displayed by the operations console

This section describes the information that is displayed by the operations console. The information that is available in the Information area depends on whether you have selected an SA z/OS domain in the *topology tree* or a resource (or resource group) in the resource tree.

Figure 15 shows the topology tree: it shows the name of the sysplex and its systems. These systems are those that have been defined in the automation policy and linked to the sysplex group. (Refer also to Figure 1 on page 2.)

Topology

Select	
<input type="radio"/>	 KEY1PLEX INGXSG34 
<input type="radio"/>	 KEY3/SYS/KEY3 
<input checked="" type="radio"/>	 KEY4/SYS/KEY4 

Figure 15. Operations console topology tree

The following elements of the operations console are described:

- Domain object
- Resource object
- Node

Domain object

Figure 16 on page 60 shows general information for an automation domain.

Information area

General	Automation domain
Policy	Name: KEY1PLEX INGXSG34
Additional Info	Status
	Domain state: Online
	Communication state: No communication problems
	✘ The automation domain contains top level resources with non-recoverable errors
	Log
	View log...

Figure 16. General information for an automation domain

This page provides the following status information:

Domain state

Online	SA z/OS is ready for automation.
Offline	SA z/OS is inhibited from automating resources or events are lost. The following are examples of error situations: <ul style="list-style-type: none">• 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 AOF_E2E_TKOVN_TIMEOUT.
Unknown	The automation adapter is not running.

Figure 17 shows policy information for an automation domain.

Information area

General	Active automation policy for KEY1PLEX INGXSG34
Policy	Policy name: BDOW.E2E.HA.ACF
Additional Info	Activation date: March 3, 2005
	Activation time: 2:05:18 PM CET

Figure 17. Policy information for an automation domain

This page provides the following information:

Policy name

The name of the automation policy database that is currently loaded by the automation manager. An AM cold or warm start, or a 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.

Figure 18 and Figure 19 on page 62 show additional information for an automation domain.

Information area

General	Automation product																										
Policy	Product name: System Automation for z/OS																										
Additional Info	Version: V3R1M0																										
	End-to-end automation management adapter																										
	Adapter name: System Automation Adapter																										
	Version: 1.1.0																										
	Node: BOEKEY4																										
	Additional properties for KEY1PLEX INGXSG34																										
	<table border="1"><thead><tr><th>Name ^</th><th>Value ^</th></tr></thead><tbody><tr><td>Primary Manager Start Type</td><td>COLD</td></tr><tr><td>Primary Manager Name</td><td>KEY4\$\$\$4</td></tr><tr><td>XCF Group Name</td><td>INGXSG34</td></tr><tr><td>Sysplex Name</td><td>KEY1PLEX</td></tr><tr><td>Primary Agent Version</td><td>V3R1M0</td></tr><tr><td>Primary Agent Name</td><td>KEY4</td></tr><tr><td>Primary Agent NetView Version</td><td>NV51</td></tr><tr><td>Adapter Operating System</td><td>z/OS 01.05.00</td></tr><tr><td>Adapter Java Runtime Version</td><td>1.4.2 jitc</td></tr><tr><td>Adapter Startup Time</td><td>2005-03-22 17:51:59 +0100</td></tr><tr><td>Adapter Plugin Build Version</td><td>SAFOS.3.1.0.D12.20050322.1004, NO_APAR</td></tr><tr><td>Adapter Plugin Startup Time</td><td>2005-03-22 17:52:02 +0100</td></tr></tbody></table>	Name ^	Value ^	Primary Manager Start Type	COLD	Primary Manager Name	KEY4\$\$\$4	XCF Group Name	INGXSG34	Sysplex Name	KEY1PLEX	Primary Agent Version	V3R1M0	Primary Agent Name	KEY4	Primary Agent NetView Version	NV51	Adapter Operating System	z/OS 01.05.00	Adapter Java Runtime Version	1.4.2 jitc	Adapter Startup Time	2005-03-22 17:51:59 +0100	Adapter Plugin Build Version	SAFOS.3.1.0.D12.20050322.1004, NO_APAR	Adapter Plugin Startup Time	2005-03-22 17:52:02 +0100
Name ^	Value ^																										
Primary Manager Start Type	COLD																										
Primary Manager Name	KEY4\$\$\$4																										
XCF Group Name	INGXSG34																										
Sysplex Name	KEY1PLEX																										
Primary Agent Version	V3R1M0																										
Primary Agent Name	KEY4																										
Primary Agent NetView Version	NV51																										
Adapter Operating System	z/OS 01.05.00																										
Adapter Java Runtime Version	1.4.2 jitc																										
Adapter Startup Time	2005-03-22 17:51:59 +0100																										
Adapter Plugin Build Version	SAFOS.3.1.0.D12.20050322.1004, NO_APAR																										
Adapter Plugin Startup Time	2005-03-22 17:52:02 +0100																										
	Page 1 of 2 <input type="button" value="▶"/> <input type="text" value="1"/> <input type="button" value="Go"/> Total: 14 Filtered: 14 Displayed: 12																										

Figure 18. Additional information for an automation domain, page 1

Information area

Automation product

Product name: System Automation for z/OS
Version: V3R1M0

End-to-end automation management adapter

Adapter name: System Automation Adapter
Version: 1.1.0
Node: BOEKEY4

Additional properties for KEY1PLEX INGXSG34

Name	Value
Primary Manager Startup Time	2005-03-17 16:55:04 +0100
Primary Manager System Name	KEY4

Page 2 of 2 2 Go Total: 14 Filtered: 14 Displayed: 2

Figure 19. Additional information for an automation domain, page 2

These pages provide the following additional information:

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 automation adapter.

Version

The version of the automation adapter.

Node The identifier for the system where the 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 run time version that the automation adapter is running in.

Adapter Startup Time

The timestamp when the automation adapter was started.

Adapter Plugin Startup Time

The timestamp when the automation adapter plug-in was initialized.

Adapter Plugin Build Version

A unique identifier that defines the version of the 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.

Resource object

Figure 20 on page 64 shows general information for a resource object.

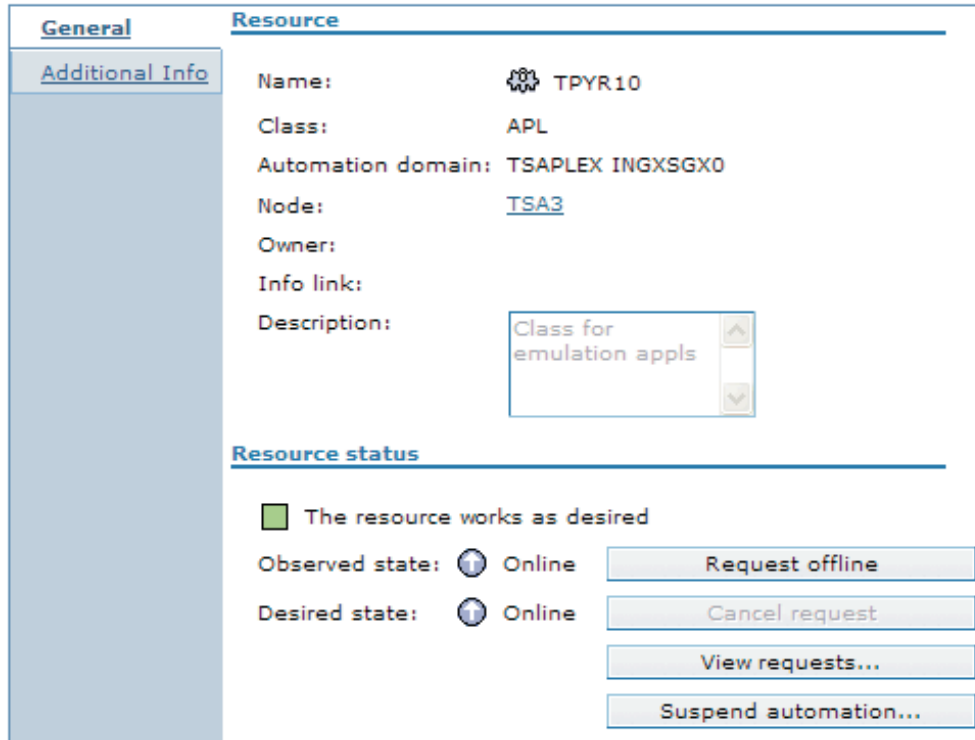


Figure 20. General information for a resource object

This page provides the following information:

Name The resource name.

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 AUTOMATION INFO policy item of the automation policy.

Info link

The Info link as specified in the AUTOMATION 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 21.

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 72.

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 72.

Request offline/online

The automation adapter converts these functions to SA z/OS specific commands. For details concerning these requests see “Requests issued by the operations console and the end-to-end automation manager” on page 68.

Cancel request

The automation adapter converts these functions to SA z/OS specific commands. For details concerning these requests see “Requests issued by the operations console and the end-to-end automation manager” on page 68.

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.

Figure 21 shows additional information for a resource object.

Information area

Additional properties for MONALISA/APL/KEY4	
Name ^	Value ^
SA compound status	Degraded
SA observed status	Available
SA desired status	Available
SA health status	Minor
Page 1 of 1	Total: 4 Filtered: 4 Displayed: 4

Figure 21. Additional information for a resource object

This page provides 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

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

Figure 22 and Figure 23 on page 67 show relationships information for a resource object.

Information area

<p>General</p> <p>Relationships</p> <p>Additional Info</p>	<p>Forward relationships</p> <p>Resources on which ADAPTER/APL/KEY4 depends on</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 5%;">Select</th> <th style="width: 70%;">Resource</th> <th style="width: 25%;">Condition</th> </tr> </thead> <tbody> <tr> <td><input type="radio"/></td> <td>▼ ADAPTER/APL/KEY4</td> <td></td> </tr> <tr> <td><input type="radio"/></td> <td>▼ MakeAvailable</td> <td></td> </tr> <tr> <td><input type="radio"/></td> <td>▶ E2EMOVE/APG</td> <td>passive/WhenObservedDown</td> </tr> <tr> <td><input type="radio"/></td> <td>▼ ForceDown</td> <td></td> </tr> <tr> <td><input type="radio"/></td> <td>▶ EAS/APL/KEY4</td> <td>WhenObservedHardDown</td> </tr> <tr> <td><input type="radio"/></td> <td>▼ HasParent</td> <td></td> </tr> <tr> <td><input type="radio"/></td> <td>▶ EAS/APL/KEY4</td> <td></td> </tr> </tbody> </table> <p>Page 1 of 1 Total: 7 Displayed: 7 Selected: 0</p> <p>Backward relationships</p> <p>Resources that depend on ADAPTER/APL/KEY4</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Source</th> <th style="width: 25%;">Relationship</th> <th style="width: 25%;">Target</th> <th style="width: 25%;">Condition</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> </tr> <tr> <td colspan="4" style="text-align: center;">Total: 0 Filtered: 0 Displayed: 0</td> </tr> </tbody> </table>	Select	Resource	Condition	<input type="radio"/>	▼ ADAPTER/APL/KEY4		<input type="radio"/>	▼ MakeAvailable		<input type="radio"/>	▶ E2EMOVE/APG	passive/WhenObservedDown	<input type="radio"/>	▼ ForceDown		<input type="radio"/>	▶ EAS/APL/KEY4	WhenObservedHardDown	<input type="radio"/>	▼ HasParent		<input type="radio"/>	▶ EAS/APL/KEY4		Source	Relationship	Target	Condition					Total: 0 Filtered: 0 Displayed: 0			
Select	Resource	Condition																																			
<input type="radio"/>	▼ ADAPTER/APL/KEY4																																				
<input type="radio"/>	▼ MakeAvailable																																				
<input type="radio"/>	▶ E2EMOVE/APG	passive/WhenObservedDown																																			
<input type="radio"/>	▼ ForceDown																																				
<input type="radio"/>	▶ EAS/APL/KEY4	WhenObservedHardDown																																			
<input type="radio"/>	▼ HasParent																																				
<input type="radio"/>	▶ EAS/APL/KEY4																																				
Source	Relationship	Target	Condition																																		
Total: 0 Filtered: 0 Displayed: 0																																					

Figure 22. Forward relationships information for a resource object

Information area

Figure 23. Backward relationships information for a resource object

This page shows both forward and backward relationships for the selected resource where the Target is the supporting resource that the selected resource depends on.

Request issued to resource objects

Figure 24 shows requests for a resource object.

Figure 24. request list for a resource object

This page provides the following information for requests:

Requested action

Online for MakeAvailable requests, Offline 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.

Node

Figure 25 shows general information for a node.

The screenshot shows a web interface titled "Information area" with a blue header. Below the header is a tabbed interface with two tabs: "General" and "Node". The "Node" tab is active, displaying the following information:

- Name:** TSA1 (with a small icon)
- Class:** SYS
- Automation domain:** TSAPLEX INGXSGX0
- Description:** A text input field with a vertical scrollbar.

Below the "Node" tab is a section titled "Node status" with a blue header. It contains:

- Observed state:** Online (with a blue circle icon)
- Is included in automation:** Yes (with a radio button) and an "Exclude node..." button.

Figure 25. General information for a node

This page provides the following information:

Name The name of the system.

Class The resource type of the system.

Description

The description as specified as the short description in the DESCRIPTION policy item of the automation policy.

Observed state

Offline if the observed state of this resource as detected by SA z/OS is SYSGONE; otherwise Online.

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 operations console and the end-to-end automation manager."

Requests issued by the operations console and the end-to-end automation manager

The end-to-end automation GUI or manager 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 7.

Table 7. Mapping of end-to-end automation requests to SA z/OS commands

End-to-end automation request name	SA z/OS command ¹
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/SYG/system</i> REQUEST=MAKEUNAVAILABLE SOURCE= <i>source</i> WAIT= <i>time</i>
ExcludeNode	INGREQ <i>system/SYG/system</i> REQ=STOP SOURCE= <i>source</i> USERID= <i>requestor</i> WAIT= <i>time</i> CMT= <i>comment</i>
ResetFromNRE	SETSTATE <i>resource</i> AUTODOWN TARGET= <i>system</i>
SuspendResource	INGSET SET <i>resource</i> AUTOFLAG=NO WAIT= <i>time</i>
ResumeResource	INGSET SET <i>resource</i> AUTOFLAG=YES WAIT= <i>time</i>
1. These commands also take the parameters OUTMODE=LINE and VERIFY=NO.	

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 72. 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 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 GUI or the end-to-end automation manager
2. The timeout in seconds defined in the end-to-end automation 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 7 on page 69 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 Tivoli 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 online, the end-to-end automation manager issues a start request for it after the referenced SA z/OS resource is started (after an automation agent restart), even though the NOSTART option is specified for the resource.

Definition of references for SA z/OS resources in an end-to-end automation policy

The end-to-end automation manager uses an end-to-end automation policy that is written in an XML file. This policy contains definitions of resources that are to be managed, their desired state, and their relationships. (See "Creating and modifying policies" in *System Automation for Multiplatforms End-to-End Automation Management* for more details.) Figure 26 shows an example of a resource reference that is based on the operations console figures in this appendix.

```
<ResourceReference name="movegroup1">
  <DesiredState>Online</DesiredState>
  <Description>MOVE group for E2E automation adapter</Description>
  <Owner>PeterPan</Owner>
  <ExternalLink>InfoLink<www.goodcompany.com/e2e/InfoWeb/movegroup.html</ExternalLink>
  <ReferencedResource>
    <AutomationDomain>KEY1PLEX INGXSG34</AutomationDomain>
    <Name>E2EMOVE</Name>
    <Class>APG</Class>
    <Node>KEY4</Node>
  </ReferencedResource>
</ResourceReference>
```

Figure 26. Example end-to-end automation resource reference

INGE2E

Purpose

The INGE2E command provides the operational API for the end-to-end automation adapter.

Format



Parameters

RESET

Specifies cleaning up end-to-end automation-specific settings in the primary automation agent.

RECYCLE

Specifies recycling a MOVE group. The MOVE group is part of the end-to-end automation high availability policy as described in “Operating the end-to-end automation adapter via SA z/OS” in *System Automation for z/OS End-to-end automation adapter*. The MOVE group includes the end-to-end automation adapters on the different systems within a subplex.

The INGE2E command is specified in the command entry of the entry/type-pair E2E_ADPT/ACORESTART. It recycles the MOVE group only in cases when ACORESTART is caused by recycling the primary automation agent or an ACF COLD of the configuration file.

move-group

The name of the MOVE group to be recycled.

RESYNC

Specifies resynchronizing the end-to-end automation manager with the first-level automation. For this purpose the automation manager is first notified that the automation domain is offline and then it is set to online again. This causes the end-to-end automation manager to request initial status information for all resources known by it in that automation domain.

Example

To recycle the MOVE group E2E_ADPT_X/APG, specify:

```
INGE2E RECYCLE,E2E_ADPT_X/APG
```

Status mapping

This section gives details of the mapping of statuses between the SA z/OS automation adapter and end-to-end automation server.

Introduction

The SA z/OS automation manager 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

Note that not all values that are available are used. For a full list of status values, refer to *System Automation for Multiplatforms End-to-End Automation Management*.

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

The mapping is carried out in three steps:

Step 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.”

Note: There is a direct one-to-one mapping of the SA z/OS desired state to the end-to-end automation desired state.

Step 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 74.

Step 3. The compound state of the end-to-end resource data model is derived from the operational state. See “Step 3” on page 75.

Step 1

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 74).
2. The SA z/OS desired status values Available and Unavailable correspond to the values Online and Offline of the end-to-end automation desired state. Other values of the end-to-end automation desired state are ignored.

SA z/OS desired status		End-to-end automation desired state
Available	⇒	Online
Unavailable	⇒	Offline

3. An automation status of Unknown is mapped to the operational status Unknown, regardless of the other statuses.

SA z/OS automation state		End-to-end automation operational status
Unknown	⇒	Unknown

4. Table 8 shows the SA z/OS automation manager status mapping. Note that the SA z/OS desired state is mapped directly to the end-to-end automation desired state (which is not given in the table).

Table 8. SA z/OS automation manager status mapping

SA z/OS automation manager status				End-to-end automation resource data model status	
Observed state	Desired state	Automation state	⇒	Observed state	Operational state
Available	*	Problem	⇒	Online	Error
Available	*	*	⇒	Online	Ok

Table 8. SA z/OS automation manager status mapping (continued)

SA z/OS automation manager status				End-to-end automation resource data model status	
Observed state	Desired state	Automation state	⇒	Observed state	Operational state
WasAvail	*	*	⇒	Online	LostCommunication
SoftDown	*	Problem	⇒	Offline	Error
SoftDown	*	*	⇒	Offline	Ok
HardDown	*	*	⇒	Offline	For APLs: NonRecoverableError For other resource types: Error
Standby	*	Problem	⇒	Offline	Error
Standby	*	*	⇒	Offline	Dormant
Starting	*	Problem	⇒	Starting	Error
Starting	*	*	⇒	Starting	InProgress
Degraded	*	*	⇒	Online	Degraded
Stopping	*	Problem	⇒	Stopping	Error
Stopping	*	*	⇒	Stopping	InProgress
Problem	Available	*	⇒	Starting	Error
Problem	Unavailable	*	⇒	Stopping	Error
SysGone	*	*	⇒	Offline	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 9.

Table 9. Status mapping: second step if the SA z/OS compound status reports a problem

SA z/OS compound status	Desired status	Observed status		Operational state
Degraded	*	*	⇒	Degraded
Awaiting	Online	Offline	⇒	StartReqPending
Awaiting	Offline	Online	⇒	StopReqPending
Denied	Online	Offline	⇒	StartInhibited
Denied	Offline	Online	⇒	StopInhibited
Inhibited	Online	Offline Stopping	⇒	StartInhibited
Inhibited	Offline	Online Starting	⇒	StopInhibited
Inhibited	Online	Online	⇒	StopInhibited
Inhibited	Offline	Offline	⇒	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:

Operational state		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 C. Messages

This chapter 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*

Explanation: A syntax error has occurred in the configuration file, for example, a leading = on a line.

System action: The automation adapter stops.

Operator response: Analyze the configuration file for invalid syntax.

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.

Operator response: Analyze the configuration file for invalid key/value pairs.

EEZA0003E The key "*the key that was not found*" was not found and no default value was given

Explanation: The system wanted to retrieve from the configuration file a value that did not exist and no default value was given.

System action: The automation adapter stops.

Operator response: Supply a value for the key in the configuration file.

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*

Explanation: The system expected an integer value between the given bounds (inclusive) for the given key, but found a value outside these bounds.

System action: The automation adapter stops.

Operator response: Supply a value within the given bounds for the key.

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:

EEZA0010E • EEZA0013E

- 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 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 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.

- 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. Contact IBM Support if it is an internal error.

EEZA0010E Request expires 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 or contact IBM Support.

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 or contact IBM Support.

EEZA0013E Authentication for user ID *user name* failed

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.

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: Contact IBM Support.

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: None.

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.

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 configuration file.

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.

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.

EEZA0027E Cannot load configuration file: *configuration file name*

Explanation: The specified configuration file cannot be loaded.

System action: The automation adapter stops.

Operator response: Correct the configuration file.

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: Correct the configuration file.

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: Correct the configuration file.

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: Correct the configuration file.

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.

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 that were sent previously.

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.sapplugin.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: None.

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: Contact IBM Support. Check the trace file for the place that threw the internal exception.

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: None.

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: None.

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: None.

EEZA0064I The termination command for the adapter was successful

Explanation: The automation adapter was successfully stopped.

System action: The automation adapter stops.

Operator response: None.

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 valid host name.

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 valid domain name in the plug-in configuration file.

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: None.

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: None.

EEZA0102I The adapter is ready

Explanation: The automation adapter startup sequence is complete.

System action: None.

Operator response: None.

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: None.

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: None.

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: None.

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: None.

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: None.

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: None.

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.

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: Contact IBM Support.

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.

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.

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 DISPPI.

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.

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.

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.

ING236W • ING238E

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" on page 88.
- 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.

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** *'time'* **was too short –** *missing_time*.

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 task execution requests. If necessary, increase it as described in "Requests issued by the operations console and the end-to-end automation manager" on page 68.

If the timeout occurs during the INIT_DOMAIN request, increase the TIMEOUT value in `ing.adapter.plugin.properties`.

ING250E **Execution of a service timed out. The expiration time** *time* **was reduced to** *number* **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** *filter1* **and** *filter2*. **Values** *value1* **and** *value2* **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** *system*.

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 *receiver* 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 *domain* 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 end-to-end automation server 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 *service* 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: *name*

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: *type*

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: *parameters*

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 *resource* is CTLDOWN.

Explanation: A reset-from-non-recoverable-error request was received for a resource that is in agent status CTLDOWN. Such a request is not supported for resources in agent status CTLDOWN.

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 *resource* 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 *resource* 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 *resource* 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 *request 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:** *wait*

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 *user* is not authorized to execute command:** *command*

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 8, "Operating the end-to-end automation adapter," on page 33.

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.

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 8, "Operating the end-to-end automation adapter," on page 33.

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 information about this topic see adapter manual.

INGX9710I Could not retrieve installed codepage. Trying with default codepage:

Explanation: The script was not able to determine the installed codepage because environment variable E2A 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 E2A 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.

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 operations console and the end-to-end automation manager" on page 68.

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)
- 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 *NetView Application 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.
- 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 from path *path*.

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 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 Tivoli 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 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.

Index

A

- accessibility xiii
- ACORESTART and CHECK_HEALTH
 - function failure 53
- AdapterJoinEvent 55
- additional information, automation
 - domain 61
- advanced automation option
 - AOF_E2E_EAS_PPI 16, 52
 - AOF_E2E_EVT_RETRY 16
 - AOF_E2E_EXREQ_NETLOG 17
 - AOF_E2E_TKOVV_TIMEOUT 17
 - customizing 16
 - INGREQ_PRI.E2EMGR 17, 71
 - setting request priority 71
 - setting the PPI receiver ID 33, 52
- AOF_E2E_EAS_PPI advanced automation option 16
- AOF_E2E_EVT_RETRY advanced automation option 16
- AOF_E2E_EXREQ_NETLOG advanced automation option 17
- AOF_E2E_TKOVV_TIMEOUT advanced automation option 17
- AOF233E message, diagnosing domain initialization failure 49
- AOFEXE2E sample exit and overwriting commands 70
- application definition
 - automation adapter 41
 - event/automation service 40
- authentication 24
- authentication checking 29
- authentication, operator 20
- authorization checking 29
- authorization exit 29
- authorization user exit, sample 30
- Auto Operators entry type 15
- automated applications defined to SA z/OS 39
- automated function 22, 57
- automated operator functions 15
 - and concurrent working threads 16
 - defining 15
 - EVTOPER, checking status 49
- automation adapter
 - application definition 41
 - asynchronous communication 4
 - communication 3
 - communication flow 21
 - configuration files 19
 - configuring 19
 - customizing 19
 - event/automation service 4
 - failover to a backup system 38
 - grouping considerations 39
 - high availability 37
 - host name 20, 56
 - host name port 56
 - ingadapter.sh 23
 - initial contact timeout interval 55

- automation adapter (*continued*)
 - initial contact with end-to-end automation server 55
 - installation steps 9
 - installing 9
 - introduction 1, 2
 - IP address 20, 56
 - IP address port 56
 - Java Virtual Machine access 21
 - master configuration file 19
 - messages 77
 - MOVE group definition 47
 - non-disruptive recovery from errors 37
 - operating 33
 - plug-in domain name 23, 58
 - plug-in, messages 85
 - preparing to install 9
 - prerequisites 5
 - problem determination 49
 - program-to-program interface 3
 - purpose and rules 37
 - recycling after stopping NetView 53
 - recycling after stopping SSI subsystem 53
 - recycling in the same system 38
 - remote contact activity interval 56
 - restart 38
 - resynchronization of 38
 - security 29
 - start-stop script 23
 - start-stop script, and SSL configuration file 23
 - start-stop script, using 33
 - stop delay 56
 - synchronous communication 3
 - TCP/IP port relationship 21
 - two in same sysplex 4
- automation agent, primary 4
- automation domain
 - additional information 61
 - general information 59
 - policy information 60
 - remaining offline after error recovery 53
- automation manager status 72
 - mapping to resource data model status 73
- automation manager, problem with automation domain after restarting 53
- automation operator tasks 16
- automation policy, defining 40
- AUTOPFN parameter, description 57

B

- backup system, failover of the automation adapter to 38
- bound socket, dropping with NETSTAT 52

- buffer queue limit
 - description 22, 57
 - setting 15

C

- CEE3501S error message, and link list errors 52
- certificate alias name 58
- certificates 23
- CHECK_HEALTH function failure 53
- code page, Java 57
- CODEPAGE parameter, description 57
- commands, overwriting for resource objects 70
- common global variable
 - customizing 16
 - INGREQ_PRI.E2EMGR 71
 - setting request priority 71
 - setting the PPI receiver ID 33
- communication
 - automation adapter, asynchronous 4
 - automation adapter, introduction 3
 - automation adapter, synchronous 3
 - event/automation service 4
 - program-to-program interface 3
- communication flow, automation adapter 21
- concurrent working threads 55
 - and automated operator functions 16
- configuration file
 - example plug-in 22
 - JAAS 24
 - Java logger 25
 - master 19
 - modifying for user HFS 10
 - plug-in 21
 - SSL 23
- configuration files 19
- configuring
 - automation adapter 19
 - global initialization file 7
 - IHSAINIT for event/automation service 7
 - IHSAMCFG for event/automation service 8
 - NetView message adapter service 8
- creating
 - hierarchical file system 9
- customizing
 - advanced automation options 16
 - automation adapter 19
 - DSIPARM data set 15
 - ingadapter.sh shell script 11
 - INGPXDST communication task 15
 - INGXINIT initialization member 15
 - NetView 15
 - NetView tasks 15
 - UNIX System Services 25

D

- default hierarchical file system, creating 9
- defining
 - application for event/automation service 40
 - application for the automation adapter 41
 - automated operator functions 15
 - automation policy 40
 - E2OPER automated function 16
 - MOVE group for the automation adapter 47
- delaying the stopping of the automation adapter 56
- disability xiii
- DISPAOPS command response 49
- DISPPI command, using for problems with program-to-program interface 50
- domain initialization failure, diagnosing 49
- domain object 59
- domain state 60
- dropping bound socket with NETSTAT 52
- DSIPARM data set
 - customizing 15
- DSIPARM data set and automation operator tasks 16

E

- E2OPER automated function, defining 16
- eez-data-directory parameter, description 55
- eez-initial-contact parameter, description 55
- eez-initial-contact-retry-interval-minutes parameter, description 55
- eez-max-connections parameter, description 55
- eez-operator-authentication parameter 29
- eez-operator-authentication parameter, description 55
- eez-plugin-classpath parameter, description 56
- eez-remote-contact-activity-interval-seconds parameter, description 56
- eez-remote-contact-hostname parameter, description 56
- eez-remote-contact-over-ssl parameter, description 56
- eez-remote-contact-port parameter, description 56
- eez-ssl-keystore parameter, description 58
- eez-ssl-keystorealias parameter, description 58
- eez-ssl-keystorepassword parameter, description 58
- eez-ssl-truststore parameter, description 58
- eez-stop-delay parameter, description 56
- EEZA0061E message 52

- EEZAdapterDefaultLogin 29
- EEZU002I message and Tivoli Common Directory 25
- EIF cache directory 20, 55
- EIF caching, switching on and off 56
- EIF receive host name 56
- EIF receive port 20, 56
- EIF send host name 20, 57
- EIF send port 57
- EIF socket reconnect interval 56
- eif-cache parameter, description 56
- eif-cache-size parameter, description 56
- eif-receive-from-hostname parameter, description 56
- eif-receive-from-port parameter, description 56
- eif-retry-interval-seconds parameter, description 56
- eif-send-to-hostname parameter, description 57
- eif-send-to-port parameter, description 57
- enabling
 - event/automation service 7
 - NetView message adapter service 7
- end-to-end automation
 - limitations 48
 - login module 24
 - operations console 59
 - resource data model 59
 - resource reference 71
 - terminology 1
- end-to-end automation management, introduction 1
- end-to-end automation manager
 - requests, overruling hold flag for 71
- end-to-end automation manager, requests issued by 69
- end-to-end automation server, initial contact by automation adapter 55
- error recovery, automation domain remaining offline after 53
- errors
 - link list 52
 - non-disruptive recovery from 37
- event log file 25
- event/automation service
 - application definition 40
 - communication with automation adapter 4
 - enabling 7
 - IHSAINIT 7
 - IHSAMCFG 7
 - initialization files 7
 - problems, using MODIFY command for 50
 - setting the PPI receiver ID 33
 - starting 7
 - starting with IHSAEVNT sample 7
 - updating IHSAEVNT sample 7
- EVTOPER automated operator function, checking status 49
- example plug-in configuration file 22
- examples of RACF profiles 31

F

- failover of the automation adapter to a backup system 38
- first failure data capture files, common directory structure 25
- format file, NetView message adapter 8
- functions, automated operator 15

G

- generateSampleKeys parameter, of ingadapter.sh start script 34
- global initialization file, configuring 7
- graphical user interface 25
- group ID in INGXINIT 22, 57
- grouping considerations for the automation adapter 39
- GRPID parameter
 - description 57
 - setting in INGXINIT 15

H

- HFS
 - See hierarchical file system
- hierarchical file system
 - creating 9
 - default 9
 - defining 12
 - shared 13
 - shared, and virtual IP 13
 - user-defined 10
- high availability, automation adapter 37
- hold flag, disregarding 71
- host name
 - automation adapter 20, 56
 - EIF receive 56
 - EIF send 20, 57

I

- IBMSupport parameter, of ingadapter.sh start script 34
- IHSAEVNT sample
 - starting event/automation service with 7
 - updating 7
- IHSAINIT
 - configuring for event/automation service 7
 - global initialization file for event/automation service 7
- IHSAMCFG
 - message adapter configuration file for event/automation service 7, 8
- ing.adapter.properties file 19
- ING008I message and automation domain timeout 54
- ING235W message and automation domain timeout 54
- ING276E message and automation domain timeout 54
- ingadapter.sh shell script
 - and nohup command 34
 - and SSL configuration file 23

ingadapter.sh shell script (*continued*)

- customizing 11
- parameters 34
- prerequisites 33
- sample 11
- syntax 34
- using 33

INGAMS command, using to check primary NetView agent 49

INGE2E command 71

INGE2E RESYNC command and automation domain timeout 54

INGEVOBS command, using to check end-to-end automation registration consumer registration, checking with INGEVOBS command 51

INGEVOBS LIST command response 51

INGPX DST communication task customizing 15
problem with automation domain after restarting 53

INGREQ_PRI.E2EMGR advanced automation option 17

INGREQ_WAIT common global variable, setting timeout with 54

INGX9820 message and recycling automation adapter 53

INGX9905 message and recycling automation adapter 53

INGXADPT 34

INGXINIT initialization member, customizing 15

INGXINIT, group ID 22, 57

initial contact, automation adapter timeout interval 55
with end-to-end automation server 55

initialization files for event/automation service 7

installation steps, automation adapter 9

installing the automation adapter 9

introducing the automation adapter 1

IP address automation adapter 20, 56
master configuration file 19

J

JAAS configuration file end-to-end automation login module 24
overview 24
sample 24
z/OS login module 24

JAAS login module 29

Java

- classpath and security 56
- code page 57
- JIT compiler, and USS problems 53
- loading DLLs, problems with 53
- logger configuration file 25
- system properties, and SSL configuration file 23

Java Virtual Machine 21

JNI exception 15

K

keyboard xiii

keystore 23

keystore file, password 58

keystore filename 58

L

limit, buffer queue, setting 15

limitations to end-to-end automation 48

log file

- common directory structure 25
- event 25
- message 25
- trace 25

logger configuration file, Java 25

login module

- end-to-end automation 24

JAAS 24

z/OS 24

LookAt message retrieval tool xii

M

master configuration file

- introduction 19
- IP addresses 19
- parameters 55
- sample 20

message adapter service, NetView

- configuring 8
- enabling 7

message log file 25

message retrieval tool, LookAt xii

messages

- AOF233E and domain initialization failure 49
- automation adapter 77
- automation adapter plug-in 85
- CEE3501S and link list errors 52
- EEZU002I and Tivoli Common Directory 25
- ING008I and automation domain timeout 54
- ING235W and automation domain timeout 54
- ING276E and automation domain timeout 54
- INGX9820 and recycling automation adapter 53
- INGX9905 and recycling automation adapter 53

MODIFY command, for problems with event/automation service 50

modifying

- configuration files, for user HFS 10
- ingadapter.sh start script, for user HFS 10

MOVE group definition for the automation adapter 47

MVS MODIFY command, for problems with event/automation service 50

N

NETSTAT command, using to drop bound socket 52

NetView

checking primary agent with INGAMS 49

customizing 15

DSIPARM data set, customizing 15

event/automation service, problems with 50

message adapter format file 8

message adapter service, configuring 8

message adapter service, enabling 7

operator task name 22, 57

PIPE default safe 29

planned outages 48

stopping, recycling automation

adapter after 53

tasks, customizing 15

node, general information 68

nohup command, and ingadapter.sh start script 34

non-disruptive recovery from errors 37

O

offline, automation domain remaining

after error recovery 53

online, switching automation domain to after error recovery 53

operating the automation adapter

as a SA z/OS defined resource 35

from USS console 33

with customizable procedure, INGXADPT 34

operations console

requests issued by 69

resource data displayed by 59

operator authentication 20

operator task name, NetView 22, 57

overruling hold flag for end-to-end

automation manager requests 71

overwriting commands, for resource objects 70

P

parameters

AUTOPFN 57

CODEPAGE 57

eez-data-directory 55

eez-initial-contact 55

eez-initial-contact-retry-interval-minutes 55

eez-max-connections 55

eez-operator-authentication 29, 55

eez-plugin-classpath 56

eez-remote-contact-activity-interval-seconds 56

eez-remote-contact-hostname 56

eez-remote-contact-over-ssl 56

eez-remote-contact-port 56

eez-ssl-keystore 58

eez-ssl-keystorealias 58

eez-ssl-keystorepassword 58

- parameters (*continued*)
 - eez-ssl-truststore 58
 - eez-stop-delay 56
 - eif-cache 56
 - eif-cache-size 56
 - eif-receive-from-hostname 56
 - eif-receive-from-port 56
 - eif-retry-interval-seconds 56
 - eif-send-to-hostname 57
 - eif-send-to-port 57
 - GRPID 57
 - master configuration file 55
 - plug-in configuration file 57
 - plugin-configfile-sa4zos 22, 57
 - plugin-domain-name 58
 - PPIBQL 57
 - SSL configuration file 58
 - TIMEOUT 57
 - TIMEOUT_SA_NOTAVAIL 57
- password 29
- plug-in configuration file
 - example 22
 - introduction 21
 - name, setting 57
 - parameters 57
- plug-in domain name 23, 58
- plugin-configfile-sa4zos parameter,
 - description 57
- plugin-configfile-sa4zos parameter,
 - dsscription 22
- plugin-domain-name parameter,
 - description 58
- policy information, automation
 - domain 60
- port
 - automation adapter host name 56
 - EIF receive 20, 56
 - EIF send 57
- PPI buffer queue limit 22, 57
- PPI queue, number of elements 22, 57
- PPI receiver ID
 - setting for EAS 33
 - specifying in IHSAINIT 8
- PPIBQL parameter
 - description 57
 - setting buffer queue limit in
 - INGXINIT with 15
- prerequisites for UNIX System
 - Services 33
- prerequisites, automation adapter 5
- primary automation agent 4
- primary NetView agent, problems
 - selecting 49
- problem determination
 - automation domain remaining offline
 - after error recovery 53
 - consumer registration 51
 - domain initialization failure 49
 - introduction 49
 - link list errors 52
 - NetView event/automation
 - service 50
 - program-to-program interface 50
 - selecting the primary agent 49
 - TCP/IP bind socket 52
 - timeout of requests 53

- profile
 - RACF 30
 - RACF, examples of 31
- program-to-program interface, problems
 - with 50

R

- RACF 24, 29
 - profile 30
 - profile, examples 31
- receiver ID, PPI 8
- reconnect interval, EIF socket 56
- recycling the automation adapter in the
 - same system 38
- relations to resource objects 66
- relationship, TCP/IP port and automation
 - adapter 21
- remote contact activity interval 56
- request mapping to SA z/OS commands
 - for resource objects 69
- request priority, for resource objects 71
- requests
 - issued by end-to-end automation
 - manager 69
 - issued by operations console 69
 - issued to resource objects 67
 - overruling hold flag for 71
 - overwriting commands from 70
 - timeout problems with 53
- resource data displayed by operations
 - console 59
- resource data model
 - automation domain, additional
 - information 61
 - automation domain, policy
 - information 60
 - compound status 75
 - domain object 59
 - domain state 60
 - node 68
 - node, general information 68
 - operational state 75
 - relations to resource objects 66
 - request mapping to SA z/OS
 - commands 69
 - resource object 63
 - resource object, additional
 - information 65
 - resource object, general
 - information 63
 - resource object, overwriting
 - commands for 70
 - resource object, relationships
 - information 66
 - resource object, request priority 71
 - resource object, requests issued to 67
 - status 72
 - status mapping 72
 - status mapping rules 73
 - status mapping steps 73
 - top level resource 65
 - WAIT parameter 69
- resource object
 - additional information 65
 - general information 63
 - overwriting commands for 70

- resource object (*continued*)
 - relations to 66
 - relationships information 66
 - request mapping to SA z/OS
 - commands 69
 - request priority 71
 - requests issued to 67
- resource reference, end-to-end
 - automation 71
- restart of the automation adapter 38
- resynchronization of the automation
 - adapter 38

S

- SA z/OS Agent 15
- sample authorization user exit 30
- sample ingadapter.sh shell script 11
- sample master configuration file 20
- security
 - authorization exit 29
 - automation adapter 29
 - JAAS configuration file 24
 - Java classpath 56
 - NetView PIPE default safe 29
 - password 29
 - RACF profiles 30
 - SSL configuration file 23
 - user ID 29
- setting
 - GRPID parameter in INGXINIT 15
 - PPI receiver ID for EAS 33
 - PPIBQL buffer queue limit in
 - INGXINIT 15
 - request priority for resource
 - objects 71
 - timeout with INGREQ_WAIT common
 - global variable 54
- shared hierarchical file system
 - overview 13
 - virtual IP 13
- shell script, ingadapter.sh, sample 11
- shortcut keys xiii
- socket connection 56
- socket, binding problem 52
- SSI subsystem
 - stopping, recycling automation
 - adapter after 53
- SSL configuration file
 - certificate alias name 58
 - example 23
 - Java system properties 23
 - keystore file, password 58
 - keystore filename 58
 - missing parameters 23
 - parameters 58
 - truststore filename 58
- SSL socket connection 56
- start parameter, of ingadapter.sh start
 - script 34
- start-stop script
 - JAAS option 24
 - modifying for user HFS 10
 - using 33
- starting the event/automation service 7

- status mapping
 - for problem SA z/OS compound
 - status 74
 - introduction 72
 - rules 73
 - steps 73
- stop parameter, of ingadapter.sh start script 34
- subplex, XCF group ID 22, 57
- switching on and off EIF caching 56

T

- tasks, automation operator 16
- TCP/IP bind socket problem 52
- TIMEOUT parameter, description 57
- TIMEOUT_SA_NOTAVAIL parameter, description 57
- Tivoli Common Directory 25
- top level resource 65
- trace log file 25
- traceMIN parameter, of ingadapter.sh start script 34
- traceOFF parameter, of ingadapter.sh start script 34
- traceON parameter, of ingadapter.sh start script 34
- truststore file 23
- truststore filename 58

U

- UNIX System Services
 - console 33, 34
 - customizing 25
 - prerequisites 33
 - problems loading DLLs 53
 - shell script, ingadapter.sh 11
- user ID 29
- user-defined hierarchical file system 10, 12
- USS
 - See* UNIX System Services

V

- virtual IP and shared hierarchical file system 13

W

- WAIT parameter, for SA z/OS
 - commands 69
- WebSphere authentication and end-to-end automation login module 24

X

- XCF group ID 22, 57

Z

- z/OS login module 24

Readers' Comments — We'd Like to Hear from You

System Automation for z/OS
End-to-End Automation Adapter
Version 3 Release 1

Publication No. SC33-8271-03

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.

Submit your comments using one of these channels:

- Send your comments to the address on the reverse side of this form.
- Send a fax to the following number: FAX (Germany): 07031+16-3456
FAX (Other Countries): (+49)+7031-16-3456
- Send your comments via e-mail to: s390id@de.ibm.com

If you would like a response from IBM, please fill in the following information:

Name

Address

Company or Organization

Phone No.

E-mail address



Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Deutschland Entwicklung GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany



Fold and Tape

Please do not staple

Fold and Tape



Program Number: 5698-SA3

Printed in USA

SC33-8271-03

