

IBM Remote System Explorer API

Host Configuration Guide



Note

Before using this information, be sure to read the general information under [“Notices” on page 23](#).

First edition (March 2020)

This edition applies to IBM® Remote System Explorer API Open Beta (program number 5655-EX1) and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 2020.**

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

Contents

Figures.....	v
Tables.....	vii
About this document.....	ix
Who should use this document.....	ix
Description of the document content.....	ix
Planning.....	ix
Basic customization.....	ix
Security definitions.....	ix
Operator commands.....	ix
Chapter 1. Planning.....	1
Planning considerations.....	1
Product overview.....	1
Skill requirements.....	1
Preinstallation considerations.....	1
Installation user ID.....	1
Requisite products.....	2
Required resources.....	2
Pre-configuration considerations.....	3
User ID considerations.....	3
Server considerations.....	4
Predeployment considerations.....	4
User checklist.....	5
Chapter 2. Basic customization.....	7
Requirements and checklist.....	7
Customization setup.....	7
PARMLIB updates.....	8
Set the z/OS UNIX limits in BPXPRMxx.....	8
Add the started tasks to COMMNDxx.....	8
PROCLIB changes.....	9
RSEAPI started task.....	9
JCL limitations for the PARM variable.....	10
TMPDIR processing.....	10
Security definitions.....	11
rseapi.env, the RSE API configuration file.....	11
Chapter 3. Security definitions.....	15
Requirements and checklist.....	15
Define an OMVS segment for RSE API users.....	15
Define the RSE API started tasks.....	16
Define RSE API as a secure z/OS UNIX server.....	17
Define the PassTicket support for RSE API.....	17
Define the data set profiles.....	17
Verify the security settings.....	18
Chapter 4. Operator commands.....	19

Start (S).....	19
RSEAPI	19
How to read a syntax diagram.....	20
Symbols.....	20
Operands.....	20
Syntax example.....	20
Nonalphanumeric characters and blank spaces.....	20
Selecting more than one operand.....	21
Longer than one line.....	21
Syntax fragments.....	21
Notices	23
Copyright license.....	25
Trademark acknowledgments.....	25

Figures

1. RSEAPI: started task..... 9

2. RSEAPI: alternate TMPDIR processing..... 10

3. rseapi.env: RSE API configuration file..... 12

4. START RSEAPI operator command..... 19

Tables

1. Required resources..... 2

2. Administrators needed for required tasks.....3

3. User checklist: Mandatory parts..... 5

4. Detail level for output logs..... 13

5. Security setup variables.....15

About this document

This document discusses the configuration of the IBM Remote System Explorer API functions. It includes instructions to configure IBM Remote System Explorer API on your z/OS® host system.

Who should use this document

This document is intended for system programmers who are installing and configuring IBM Remote System Explorer API.

This document lists in detail the steps that are needed to do a full setup of the product, including some non-default scenarios. To use this document, you must be familiar with the z/OS UNIX System Services and MVS™ host systems.

Description of the document content

This section summarizes the information that is given in this document.

Planning

Use the information in this chapter to plan the installation and deployment of RSE API.

Basic customization

The following customization steps are for a basic RSE API setup:

- [“Customization setup” on page 7](#)
- [“PARMLIB updates” on page 8](#)
- [“PROCLIB changes” on page 9](#)
- [“Security definitions” on page 11](#)
- [“rseapi.env, the RSE API configuration file” on page 11](#)

Security definitions

This section describes the required and optional security definitions with sample RACF® commands.

Operator commands

This section provides an overview of the available operator (or console) commands for RSE API.

Chapter 1. Planning

Use the information in this chapter and the IBM Remote System Explorer API (RSE API) software requirements to plan the installation and deployment of RSE API. The following subjects are described:

- [“Planning considerations” on page 1](#)
- [“Preinstallation considerations” on page 1](#)
- [“Pre-configuration considerations” on page 3](#)
- [“Predeployment considerations” on page 4](#)
- [“User checklist” on page 5](#)

For a complete listing of the RSE API hardware and software requirements including prerequisites and co-requisites, generate reports from [Software Product Compatibility Reports](#).

Planning considerations

Product overview

IBM Remote System Explorer API is a collection of REST APIs that allow a client to work with various components on the z/OS host system, including MVS data sets, z/OS UNIX files and commands, JES jobs, and more.

To learn more about the functionality that is offered by RSE API, see [Mainframe DEV](#), or your local IBM representative.

Skill requirements

SMP/E skills are needed for an RSE API host installation.

The configuration of RSE API requires more than the typical system programming permissions and expertise, so assistance from others might be needed. [Table 2 on page 3](#) lists the administrators who are needed for the required customization tasks.

Preinstallation considerations

For detailed instructions on the SMP/E installation of the product, see *Program Directory for IBM Remote System Explorer API* (GI13-5404).

The RSE API servers are single-system minded, and are not SYSPLEX aware. If you are using the servers in a SYSPLEX, you must ensure that the data requested by the users (data sets, job output, z/OS UNIX files) is available on the system RSE API is installed. See [“Predeployment considerations” on page 4](#) for cloning RSE API to other systems.

The file system (HFS or zFS) in which RSE API is installed must be mounted with the SETUID permission bit on (this is the system default). Mounting the file system with the NOSETUID parameter prevents RSE API from creating the user's security environment, and rejects the connection requests of the client. The same is true for the file systems hosting Java™ and z/OS UNIX binaries.

Installation user ID

The user ID that is used to install RSE API, or to install maintenance, must have at least the following attributes:

- TSO access (with a normal region size).

- An OMVS segment defined to the security system (for example, RACF), both for the user ID and its default group.
 - The HOME field must refer to a home directory that is allocated for the user, with READ, WRITE, and EXECUTE access.
 - The PROGRAM field in the OMVS segment should be /bin/sh or other valid z/OS UNIX shell, such as /bin/tcsh.
 - The user ID's default group requires a GID.
- UID=0 or READ authorization to the BPX.SUPERUSER profile in the FACILITY class.
- If the BPX.FILEATTR.APF or BPX.FILEATTR.PROGCTL profiles are defined in the FACILITY class, READ access to these profiles.
- READ, WRITE, and EXECUTE access to the /tmp directory (or a directory referenced in the TMPDIR environment variable).

Requisite products

RSE API has a list of prerequisite software that must be installed and operational before the product will work. There is also a list of corequisite software to support specific features of RSE API. These requisites must be installed and operational at runtime for the corresponding features to work as designed.

For a complete listing of the RSE API software requirements including prerequisites and co-requisites, generate reports from [Software Product Compatibility Reports](#).

Plan ahead to have these requisite products available, as it might take some time, depending on the policies at your site. The key requisites for a basic setup are:

- z/OS 2.2 or higher
- Latest service release of Java 8.0 or higher (31 or 64 bit)
- Latest service release of z/OS Explorer 3.1 or higher (FMID HALGxxx)

Required resources

RSE API requires the allocation of the systems resources listed in [Table 1 on page 2](#). Plan to have these resources available because, depending on the policies at your site, it might take some time to get the software.

<i>Table 1. Required resources</i>		
Resource	Default value	Information
started task	RSEAPI	“PROCLIB changes” on page 9
port for API communication (RSEAPI)	6800, 60800	“rseapi.env, the RSE API configuration file” on page 11
z/OS UNIX server security definition	UPDATE permission to BPX.SERVER for RSE API started task	“Define RSE API as a secure z/OS UNIX server” on page 17
PassTicket security definitions	No default	“Define the PassTicket support for RSE API” on page 17

The configuration of RSE API requires more than the typical system programming permissions and expertise; therefore, assistance from others might be needed. [Table 2 on page 3](#) lists the administrators who are needed for the required customization tasks.

Table 2. Administrators needed for required tasks		
Administrator	Task	Information
System	Typical system programmer actions are required for all customization tasks	N/A
Security	<ul style="list-style-type: none"> • Define OMVS segment for RSE API users • Define data set profiles • Define started tasks • Define z/OS UNIX server profiles • Define the PassTicket support 	Security definitions
TCP/IP	Define new TCP/IP ports	N/A
WLM	Assign the started task goals to the servers and their child processes	N/A

Pre-configuration considerations

User ID considerations

The user ID of an RSE API user must have at least the following attributes:

- An OMVS segment defined to the security system (for example, RACF), both for the user ID and its default group.
 - The HOME field must refer to a home directory allocated for the user (with READ, WRITE and EXECUTE access).
 - The PROGRAM field in the OMVS segment should be /bin/sh or other valid z/OS UNIX shell, such as /bin/tcsh.
 - The ASSIZEMAX field should not be set, so that system defaults are used.
 - The user ID does not require UID 0.

Example (command **LISTUSER userid NORACF OMVS**):

```
USER=userid
OMVS INFORMATION
-----
UID= 0000003200
HOME= /u/userid
PROGRAM= /bin/sh
CPUTIMEMAX= NONE
ASSIZEMAX= NONE
FILEPROCMA= NONE
PROCUSERMA= NONE
THREADSMA= NONE
MMAPAREAMA= NONE
```

- The user ID's default group requires a GID.

Example (command **LISTGRP group NORACF OMVS**):

```
GROUP group
OMVS INFORMATION
```

GID= 0000003243

- READ and EXECUTE access to the RSE API installation and configuration directories and files, default `/usr/lpp/IBM/rseapi/*`, `/etc/zexpl/*`, and `/var/zexpl/*`.
- READ, WRITE, and EXECUTE access to the RSE API WORK directory, default `/var/zexpl/rseapi/rseapi_WORK`, and user log directory, default `/var/zexpl/logs`.
- READ access to the RSE API installation data sets, default `HUH.SHUH*`.
- READ, WRITE, and EXECUTE access to the `/tmp` directory or a directory referenced in the `TMPDIR` environment variable.

Server considerations

RSE API can be a started task or user job. The only requirement is that the server is up and running before the first user tries to connect.

The security mechanisms used by RSE API server and services rely on the data sets and file systems they reside in being secure. This implies that only trusted system administrators should be able to update the program libraries and configuration files.

The server ports must be available for the users to connect to, and must be defined to your firewall protecting the host system if the users are not on your z/OS system. Listed below are the ports that are needed for external communication:

- RSE API requests (using TCP), default port 6800.

Predeployment considerations

RSE API supports the cloning of an installation to a different system, thus avoiding the need for a SMP/E installation on each system.

The following data sets, directories, and files are mandatory for deployment to other systems. If you copied a file to a different location, this file must replace its counterpart in the following lists.

Note: The following list does not cover the deployment needs of the prerequisite and co-requisite software.

- `HUH.#CUST.PROCLIB(*)`
- `/usr/lpp/IBM/rseapi/*`
- `/etc/zexpl/*`
- `/var/zexpl/rseapi/*` (directory structure only)

Note:

- `HUH` and `/usr/lpp/IBM/rseapi` are the high-level qualifier and path used during the installation of RSE API. `HUH.#CUST`, `/etc/zexpl` and `/var/zexpl/rseapi` are the default locations used during the customization of the product (see [“Customization setup”](#) on page 7 for more information).
- You should install RSE API in a private file system (HFS or zFS) to ease the deploying of the z/OS UNIX parts of the product. If you cannot use a private file system, use an archiving tool such as the z/OS UNIX `tar` command to transport the z/OS UNIX directories from one system to another. This method is for preserving the attributes (such as program control) for the RSE API files and directories.

For more information about the following sample commands to archive and restore the RSE API installation directory, see *UNIX System Services Command Reference* (SA22-7802).

- Archive: `cd /SYS1/usr/lpp/IBM/rseapi; tar -cSf /u/userid/rseapi.tar`
- Restore: `cd /SYS2/usr/lpp/IBM/rseapi; tar -xSpf /u/userid/rseapi.tar`

User checklist

Users of RSE API must know the result of certain host system customizations, such as TCP/IP port numbers, for the server to work properly. Use these checklists to gather the information needed.

The checklist in [Table 3 on page 5](#) lists the required results of mandatory customization steps.

<i>Table 3. User checklist: Mandatory parts</i>	
Customization	Value
RSE API TCP/IP port number. The default is 6800. See “rseapi.env, the RSE API configuration file” on page 11 .	

Chapter 2. Basic customization

The following customization steps are for a basic RSE API setup. See the chapters about the optional components for their customization requirements.

Requirements and checklist

You need the assistance of a security administrator and a TCP/IP administrator to complete this customization task, which requires the following resources and special customization tasks:

- Various PARMLIB updates
- Various security software updates
- Various TCP/IP ports for internal and client-host communication

To verify the installation and to start using RSE API at your site, do the following tasks. Unless otherwise indicated, all tasks are mandatory.

1. Create customizable copies of samples and create the work environment for RSE API. For details, see [“Customization setup” on page 7](#).
2. Update z/OS UNIX system limits and start started tasks. For details, see [“PARMLIB updates” on page 8](#).
3. Create started task procedures. For details, see [“PROCLIB changes” on page 9](#).
4. Update security definitions. For details, see [“Security definitions” on page 11](#).
5. Customize RSE API configuration files. For details, see [“rseapi.env, the RSE API configuration file” on page 11](#).

Customization setup

RSE API contains several sample configuration files and sample JCL. To avoid overwriting your customizations when applying maintenance, copy all of these members and z/OS UNIX files to a different location, and customize the copy.

Some functions of RSE API require the existence of certain directories in z/OS UNIX, which must be created during the customization of the product. To ease the installation effort, a sample job, HUHSETUP, is provided to create the copies and the required directories.

To create customizable copies of configuration files and configuration JCL, and to create required z/OS UNIX directories, customize and submit the sample HUHSETUP member in the HUH.SHUHSAMP data set. The required customization steps are described within the member.

This job performs the following tasks:

- Create HUH.#CUST.PROCLIB and populate it with sample SYS1.PROCLIB members.
- Create HUH.#CUST.JCL and populate it with sample configuration JCL.
- Populate /etc/zexpl/* with sample configuration files.
- Create /var/zexpl/rseapi/* directory structure.

Note:

- The configuration steps in this publication use the member and file locations created by the HUHSETUP job, unless noted otherwise. The original samples, which should not be updated, are in HUH.SHUHSAMP and /usr/lpp/IBM/rseapi/tomcat.base/samples and /usr/lpp/IBM/rseapi/tomcat.base/lib.

- For more details on which sample members are copied to which data set, and for more details on which directories are created, their permission bitmask, and where the various sample files are copied to, see the comments in HUH . SHUHSAMP (HUHSETUP).
- To aid in migrating an existing setup, the comments in HUH . SHUHSAMP (HUHSETUP) also document the changes between different versions of RSE API.

PARMLIB updates

The following PARMLIB changes are documented in this section:

- [“Set the z/OS UNIX limits in BPXPRMxx” on page 8](#)
- [“Add the started tasks to COMMNDxx” on page 8](#)

Set the z/OS UNIX limits in BPXPRMxx

MAXASSIZE specifies the maximum address space (process) region size. Set MAXASSIZE in SYS1 . PARMLIB (BPXPRMxx) to 2G. This is the maximum value allowed. This is a system-wide limit, and thus active for all z/OS UNIX address spaces. If this is not what you want, you can set the limit only for RSE API in your security software, as described in [“Define the RSE API started tasks” on page 16](#).

MAXTHREADS specifies the maximum number of active threads for a single process. Set MAXTHREADS in SYS1 . PARMLIB (BPXPRMxx) to 1500 or higher. This is a system-wide limit, and thus active for all z/OS UNIX address spaces. If this is not what you want, you can set the limit only for RSE API in your security software, as described in [“Define the RSE API started tasks” on page 16](#).

MAXTHREADTASKS specifies the maximum number of active MVS tasks for a single process. Set MAXTHREADTASKS in SYS1 . PARMLIB (BPXPRMxx) to 1500 or higher. This is a system-wide limit, and thus active for all z/OS UNIX address spaces. If this is not what you want, you can set the limit only for RSE API in your security software, as described in [“Define the RSE API started tasks” on page 16](#).

These values can be checked and set dynamically (until the next IPL) with the following console commands:

- DISPLAY OMVS,0
- SETOMVS MAXASSIZE=2G
- SETOMVS MAXTHREADS=1500
- SETOMVS MAXTHREADTASKS=1500

Note:

- Ensure that other BPXPRMxx values, such as those for MAXPROCSYS and MAXUIDS, are sufficient to handle the expected amount of concurrently active RSE API users.
- During the SMP/E install of RSE API, you were advised to place the code in a separate file system (zFS or HFS) and update BPXPRMxx to mount this file system during system IPL. Included is a repeat of the sample mount command in case this update still must be done:

```
MOUNT FILESYSTEM('#dsn')
MOUNTPOINT('-PathPrefix-usr/lpp/IBM/rseapi')
MODE(RDWR) /* can be MODE(READ) */
TYPE(ZFS) PARM('AGGRGROW') /* zFS, with extents */
```

Add the started tasks to COMMNDxx

Add start commands for the RSE API servers to SYS1 . PARMLIB (COMMANDxx) to start them automatically at next system IPL. Define CMD=xx in the IEASYSxx parmlib member to specify which COMMNDxx parmlib member should be used during IPL.

After the servers are defined and configured, they can be started dynamically (until the next IPL) with the following console command:

- S RSEAPI

PROCLIB changes

The following PROCLIB change is documented in this section:

- “RSEAPI started task” on page 9

Additional information is available in the following sub-sections:

- “JCL limitations for the PARM variable” on page 10
- “TMPDIR processing” on page 10

The started task listed in the following sections must reside in a system procedure library defined to your JES subsystem. In the instructions in the following sections, the IBM default procedure library, SYS1.PROCLIB, is used.

RSEAPI started task

Customize the HUH.#CUST.PROCLIB(RSEAPI) sample started task member, as described within the member, and copy it to SYS1.PROCLIB. As shown in the following code sample, provide this information:

- The home directory where RSE API is installed, default /usr/lpp/IBM/rseapi.
- The location of the configuration files, default /etc/zexpl

```

/*
/* RSE API
/*
//RSEAPI PROC HOME='/usr/lpp/IBM/rseapi',
//          APICFG='etc/zexpl',
//          RSECFG='',
//          SRVNUM='1',
//          SECURE='false'
/*
//          EXPORT SYMLIST=*
//          SET QUOTE='''
//          SET HOME=&QUOTE.&HOME.&QUOTE.
//          SET RSECFG=&QUOTE.&RSECFG.&QUOTE.
//          SET APICFG=&QUOTE.&APICFG.&QUOTE.
//          SET SRVNUM=&SRVNUM.
//          SET SECURE=&QUOTE.&SECURE.&QUOTE.
/*
//RSEAPI EXEC PGM=BPXBATCH,REGION=0M,TIME=NOLIMIT,
//          PARM='PGM &HOME./tomcat.base/start.sh'
//STDOUT DD SYSOUT=*
//STDERR DD SYSOUT=*
//STDENV DD *,SYMBOLS=JCLONLY
        BPXK_AUTOCVT=ON
CATALINA_RUN=run
RSEAPI_CFG=&APICFG
RSE_CFG=&RSECFG
SERVER_NUM=&SRVNUM
SECURE_SERVER=&SECURE
//          PEND
/*

```

Figure 1. RSEAPI: started task

Note:

- For more information about the startup parameters, see [Chapter 4, “Operator commands,”](#) on page 19.
- The sample JCL is initially named HUH.SHUHSAMP(HUHSTC) and is renamed to HUH.#CUST.PROCLIB(RSEAPI) in [“Customization setup”](#) on page 7.
- Limit the length of the job name to 7 characters or fewer. If an 8-character name is used, the **modify** and **stop** operator commands fail with message “IEE342I MODIFY REJECTED-TASK BUSY”. This behavior is caused by the z/OS UNIX design for child processes.

JCL limitations for the PARM variable

The maximum length for the PARM variable is 100 characters, which might cause problems if you use custom directory names. To bypass this problem, use one of these options:

- Use default values.

The start.sh startup script can be started without arguments, in which case the default argument values are used if no environment variables have been predefined.

- Use symbolic links.

Symbolic links can be used as shorthand for a long directory name. The following sample z/OS UNIX command defines a symbolic link (/usr/lpp/IBM/rseapi) to another directory (/long/directory/name/usr/lpp/IBM/rseapi).

```
ln -s /long/directory/name/usr/lpp/IBM/rseapi /usr/lpp/IBM/rseapi
```

TMPDIR processing

z/OS UNIX needs write access to /tmp, or another directory that is referenced by the TMPDIR variable, to be able to process certain commands during started task startup. RSE API uses the following logic to set TMPDIR during started task startup.

During started task startup, RSE API checks whether TMPDIR is already set (DD STDENV). If so, the started task uses that value. If TMPDIR is not set, the started task will test whether it can use /tmp. If not, the started task will test whether it can use the home directory that is assigned to the started task user ID. If this directory cannot be used either, startup fails.

If you cannot use the home directory, which is the default backup for /tmp, then you have to predefine TMPDIR using DD STDENV, as in the following sample:

```
//*  
/* RSE API  
/*  
/*RSEAPI PROC HOME='/usr/lpp/IBM/rseapi',  
/* APICFG='etc/zexpl',  
/* RSECFG='',  
/* SRVNUM='1',  
/* SECURE='false'  
/*  
/* EXPORT SYMLIST=*  
/* SET QUOTE=''''  
/* SET HOME=&QUOTE.&HOME.&QUOTE.  
/* SET RSECFG=&QUOTE.&RSECFG.&QUOTE.  
/* SET APICFG=&QUOTE.&APICFG.&QUOTE.  
/* SET SRVNUM=&SRVNUM.  
/* SET SECURE=&QUOTE.&SECURE.&QUOTE.  
/*  
/*RSEAPI EXEC PGM=BPXBATCH,REGION=0M,TIME=NOLIMIT,  
/* PARM='PGM &HOME./tomcat.base/start.sh'  
/*STDOUT DD SYSOUT=*  
/*STDERR DD SYSOUT=*  
/*STDENV DD *,SYMBOLS=JCLONLY  
_BPXK_AUTOCVT=ON  
CATALINA_RUN=run  
RSEAPI_CFG=&APICFG  
RSE_CFG=&RSECFG  
SERVER_NUM=&SRVNUM  
SECURE_SERVER=&SECURE  
TMPDIR=/my/tempdir  
/* PEND  
/*
```

Figure 2. RSEAPI: alternate TMPDIR processing

Make sure the directory TMPDIR points at, for example, /my/tempdir, is available.

Security definitions

To create the security definitions for RSE API, customize and submit the sample HUHRACF member. The user submitting this job must have security administrator privileges, such as being RACF SPECIAL.

HUHRACF is located in HUH.#CUST.JCL, unless you specified a different location when you customized and submitted the HUH.SHUHSAMP (HUHSETUP) job. For more details, see [“Customization setup” on page 7](#).

The following list of security-related definitions for RSE API are discussed in detail in [Chapter 3, “Security definitions,” on page 15](#).

- Define an OMVS segment for RSE API users
- Define the RSE API started tasks
- Define RSE API as a secure z/OS UNIX server
- Define PassTicket support for RSE API
- Define data set profiles

Note: The sample commands assume that the setup for requisite product IBM Explorer for z/OS (FMID HALGxxx) is already completed, and that the same group name is used.

rseapi.env, the RSE API configuration file

The RSE API server processes use the definitions in `rseapi.env`.

`rseapi.env` is located in `/etc/zexpl/`, unless you specified a different location when you customized and submitted the HUH.SHUHSAMP (HUHSETUP) job. For more details, see [“Customization setup” on page 7](#). You can edit the file with the TSO **EDIT** command.

See the following sample `rseapi.env` file, which can be customized to match your system environment. Default values are provided for all variables that are not explicitly specified. The syntax of the file follows standard z/OS UNIX shell syntax rules. For example, comments start with a number sign (#) when using a US code page, and spaces around the equal sign (=) are not supported.

Note:

- For your changes to take effect, the RSEAPI started task must be restarted.
- The `rse.env` configuration file of requisite product IBM Explorer for z/OS is processed before `rseapi.env` is processed. Therefore, in `rseapi.env`, you can use variables that are defined in `rse.env`.
- Variable `RSEAPI_CFG` is set by the caller and can be used inside `rseapi.env`.

```

## home
#RSEAPI_HOME=/usr/lpp/IBM/rseapi

## ports
#RSEAPI_PORT_HTTP_1=6800
#RSEAPI_PORT_SHUTDOWN_1=60800

## security
#RSEAPI_KEYSTORE_FILE=$RSEAPI_CFG/keystore.dat
#RSEAPI_KEYSTORE_PASS=tomcat

## logging
#RSEAPI_LOG_LEVEL=SEVERE
#RSEAPI_LOGS=$RSE_LOGS

## data
#RSEAPI_DATA=/var/zexpl/rseapi
#RSEAPI_WORKDIR=$RSEAPI_DATA/rseapi_WORK

## tomcat
#CATALINA_TMPDIR=$TMPDIR

## RSE
#RSE_CFG=

## umask
#RSEAPI_UMASK=RWX.RX.RX

```

Figure 3. *rseapi.env*: RSE API configuration file

The following definitions are optional. If omitted, default values are used.

RSEAPI_HOME

RSE API home directory. The default is the directory specified in the HOME variable of the RSEAPI started task (default /usr/lpp/IBM/rseapi). Uncomment and change to match your RSE API installation.

Note: RSE API startup will fail if RSEAPI_HOME is not equal to the HOME variable of the RSEAPI started task.

RSEAPI_PORT_HTTP_1

RSE API port for server 1. The default is 6800. Uncomment and change to match your setup.

Note:

- Before selecting a port, verify that the port is available on your system by using the TSO commands NETSTAT and NETSTAT PORTL.
- This port is used for client-host communication.
- The number in this variable (RSEAPI_PORT_HTTP_<num>) corresponds to the value of the SRVNUM variable in the started task. For example, RSEAPI_PORT_HTTP_3 is used when launching a server with SRVNUM=3. If you are starting RSE API via the z/OS UNIX command line, it corresponds to the -P or -S startup arguments.

RSEAPI_PORT_SHUTDOWN_1

RSE API shutdown port for server 1. The default is 60800. Uncomment and change to match your setup.

Note:

- Before selecting a port, verify that the port is available on your system by using the TSO commands NETSTAT and NETSTAT PORTL.
- This port is used to shut down the RSE API server via the z/OS Unix command line.
- The number in this variable (RSEAPI_PORT_SHUTDOWN_<num>) corresponds to the value of the SRVNUM variable in the started task. For example, RSEAPI_PORT_SHUTDOWN_3 is used when launching a server with SRVNUM=3. If you are starting RSE API via the z/OS UNIX command line, it corresponds to the -P or -S startup arguments.

RSEAPI_KEYSTORE_FILE

The KeyStore file that holds the server certificate and private key for HTTPS encrypted communication. Uncomment and change to use the specified file.

RSEAPI_KEYSTORE_PASS

The password needed to access \$RSEAPI_KEYSTORE_FILE. Uncomment and change to match your setup.

RSEAPI_LOG_LEVEL

Detail level for RSE API server logs. The default is SEVERE, which logs error messages only.

Table 4. Detail level for output logs	
Value	Detail level
SEVERE	Log error messages only
INFO	Log error, warning, and informational messages
ALL	Log error, warning, informational, and debug messages

RSEAPI_LOGS

The root for the RSE API log directory. The default is \$RSE_LOGS which is the root for the log directory used by requisite product IBM Explorer for z/OS. Uncomment and change to use the specified path.

RSEAPI_DATA

RSE API data location. The default is /var/zexpl/rseapi. Uncomment and change to use the specified path.

Note: If you did not use the SHUHSAMP (HUHSETUP) sample job to build the customizable environment, verify that the last directory in the path specified in RSEAPI_DATA has read, write, and execute permission for owner, group, and other (permission bitmask 777).

RSEAPI_WORKDIR

RSE API work directory. The default is \$RSEAPI_DATA/rseapi_WORK. Uncomment and change to use the specified path.

Note: If you did not use the SHUHSAMP (HUHSETUP) sample job to build the customizable environment, verify that the last directory in the path specified in RSEAPI_DATA has read, write, and execute permission for owner, group, and other (permission bitmask 777).

CATALINA_TMPDIR

Specifies the path used to store temporary files. The default is \$TMPDIR. Uncomment and change to use the specified path.

RSE_CFG

Directory holding rse.env, the configuration file of requisite product IBM Explorer for z/OS. The default is a null string, which results in using the value of variable RSEAPI_CFG. RSEAPI_CFG is set in the RSE API started task. Uncomment and change to match your z/OS Explorer setup.

Note: RSE API startup will fail if RSE_CFG is not equal to the RSECFG variable of the RSE API started task.

RSEAPI_UMASK

Specifies the access permission mask for z/OS UNIX files and directories that are created by users. The default is RWX.RX.RX, which grants the owner read, write, and execute/search access. The owner's default group and everyone else have read and execute access. To set the required access permissions, uncomment and customize this variable.

UNIX standards dictate that permissions can be set for three types of users: owner, group, and other. The fields in this variable match this order, and the fields are separated by a period (.). Each field can be empty (which equals **N**), or have **N**, or any combination of **R**, **W**, and **X** as values, where **N** = none, **R** = read, **W** = write, and **X** = execute/search.

Chapter 3. Security definitions

Customize and submit the sample HUHRACF member, which has sample RACF commands to create the basic security definitions for RSE API.

HUHRACF is located in HUH.#CUST.JCL, unless you specified a different location when you customized and submitted the HUH.SHUHSAMP (HUHSETUP) job. For more details, see [“Customization setup”](#) on page 7.

See the *RACF Command Language Reference* (SA22–7687), for more information about RACF commands.

The following sections describe the required steps, optional configuration, and possible alternatives.

Requirements and checklist

To complete the security setup, the security administrator must know the values that are listed in [Table 5](#) on page 15. These values were defined during previous steps of the installation and customization of RSE API.

Table 5. Security setup variables		
Description	<ul style="list-style-type: none">• Default value• Where to find the answer	Value
RSE API product high-level qualifier	<ul style="list-style-type: none">• HUH• SMP/E installation	
RSE API started task name	<ul style="list-style-type: none">• RSEAPI• HUH.#CUST.PROCLIB(RSEAPI), as described in “PROCLIB changes” on page 9.	

The following list is an overview of the actions that are required to complete the basic security setup of RSE API. As documented in the following sections, different methods can be used to fulfill these requirements, depending on the required security level.

- [“Define an OMVS segment for RSE API users”](#) on page 15
- [“Define the RSE API started tasks”](#) on page 16
- [“Define RSE API as a secure z/OS UNIX server”](#) on page 17
- [“Define the PassTicket support for RSE API”](#) on page 17
- [“Define the data set profiles”](#) on page 17
- [“Verify the security settings”](#) on page 18

Define an OMVS segment for RSE API users

A RACF OMVS segment or equivalent that specifies a valid nonzero z/OS UNIX user ID (UID), home directory, and shell command must be defined for each user of RSE API. Their default group also requires an OMVS segment with a group ID.

In the following sample RACF commands, replace the `#userid`, `#user-identifier`, `#group-name`, and `#group-identifier` placeholders with actual values:

- `ALTUSER #userid
OMVS(UID(#user-identifier) HOME(/u/#userid) PROGRAM(/bin/sh) NOASSIZEMAX)`
- `ALTGROUP #group-name OMVS(GID(#group-identifier))`

Define the RSE API started tasks

The following sample RACF commands create the RSEAPI started task, with a protected user ID (STCAPI) and the STCGROUP group assigned to it.

- `ADDGROUP STCGROUP OMVS(AUTOUID)
DATA('GROUP WITH OMVS SEGMENT FOR STARTED TASKS')`
- `ADDUSER STCAPI DFLTGRP(STCGROUP) NOPASSWORD NAME('RSE API')
OMVS(AUTOUID HOME(/tmp) PROGRAM(/bin/sh) ASSIZEMAX(2147483647))
DATA('IBM Remote System Explorer API (RSE API)')`
- `RDEFINE STARTED RSEAPI.* DATA('ZEXPL - RSE API')
STDATA(USER(STCAPI) GROUP(STCGROUP) TRUSTED(NO))`
- `SETROPTS RACLIST(STARTED) REFRESH`

Note:

- The sample commands assume that the setup for requisite product IBM Explorer for z/OS (FMID HALGxxx) is already completed.
- Ensure that the started tasks user IDs are protected by specifying the NOPASSWORD keyword.
- Ensure that RSE API has a unique OMVS uid due to the z/OS UNIX related privileges granted to this uid.
- RSE API requires a large address space size (2GB) for proper operation. Set this value in the ASSIZEMAX variable of the OMVS segment for user ID STCAPI. Setting this value ensures that RSE API gets the required region size, regardless of changes to MAXASSIZE in SYS1.PARMLIB(BPXPRMxx).
- RSE API also requires a large number of threads for proper operation. You can set the limit in the THREADSMAX variable of the OMVS segment for user ID STCAPI. Setting the limit ensures that RSE API gets the required thread limit, regardless of changes to MAXTHREADS or MAXTHREADTASKS in SYS1.PARMLIB(BPXPRMxx).

Consider making the STCAPI user ID restricted. Users with the RESTRICTED attribute cannot access protected (MVS) resources that they are not specifically authorized to access.

```
ALTUSER STCAPI RESTRICTED
```

To ensure that restricted users do not gain access to z/OS UNIX file system resources through the “other” permission bits, define the RESTRICTED.FILESYS.ACCESS profile in the UNIXPRIV class with UACC(NONE). For more information about restricting user IDs, see *Security Server RACF Security Administrator's Guide* (SA22-7683).



Attention: If you use restricted user IDs, explicitly add the permission to access a resource by using the TSO **PERMIT** or the z/OS UNIX **setfacl** commands. The resources include those resources where the RSE API documentation uses UACC(READ), or where it relies on common z/OS UNIX conventions, such as everyone having read and execute permission for Java libraries. Test the access before activating it on a production system.

Define RSE API as a secure z/OS UNIX server

RSE requires UPDATE access to the BPX.SERVER profile to create or delete the security environment for the client's thread. Note that using UID(0) to bypass this requirement is not supported. This step is required for clients to be able to connect.

- PERMIT BPX.SERVER CLASS(FACILITY) ACCESS(UPDATE) ID(STCAPI)
- SETROPTS RACLIST(FACILITY) REFRESH

Note: The sample commands assume that the setup for requisite product IBM Explorer for z/OS (FMID HALGxxx) is already completed.

Define the PassTicket support for RSE API

The client's password is used only to verify the identity upon connection. Afterward, PassTickets are used to maintain thread security. This step is required for clients to be able to connect.

Replace FEKAPPL with the application ID used by RSED started task, as defined in /etc/zexpl/rse.env.

- PERMIT IRRPTAUTH.FEKAPPL.* CLASS(PTKTDATA) ACCESS(UPDATE) ID(STCAPI)
- SETROPTS RACLIST(PTKTDATA) REFRESH

Note:

- The sample commands assume that the setup for requisite product IBM Explorer for z/OS (FMID HALGxxx) is already completed.



Attention: The client connection request fails if PassTickets are not set up correctly.

Define the data set profiles

READ access for users and ALTER for system programmers is sufficient for RSE API data sets. Replace the #sysprog placeholder with valid user IDs or RACF group names. Also, ask the system programmer who installed and configured the product for the correct data set names. HUH is the default high-level qualifier used during installation.

- ADDGROUP (HUH) OWNER(IBMUSER) SUPGROUP(SYS1)
DATA('IBM Remote System Explorer API - HLQ STUB')
- ADDSD 'HUH.*.**' UACC(NONE)
DATA('IBM Remote System Explorer API (RSE API)')
- PERMIT 'HUH.*.**' CLASS(DATASET) ACCESS(READ) ID(*)
- PERMIT 'HUH.*.**' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
- SETROPTS GENERIC(DATASET) REFRESH

Note:

- The sample commands in this publication and in the HUHRACF job assume that Enhanced Generic Naming (EGN) is active. When EGN is active, the ** qualifier can be used to represent any number of qualifiers in the DATASET class. Substitute *.* with * if EGN is not active on your system. For more information about EGN, see *Security Server RACF Security Administrator's Guide (SA22-7683)*.

Verify the security settings

Use the following sample commands to display the results of your security-related customizations.

- OMVS segment for users
 - `LISTUSER #userid NORACF OMVS`
 - `LISTGRP #group-name NORACF OMVS`
- Started tasks
 - `LISTGRP STCGROUP OMVS`
 - `LISTUSER STCAPI OMVS`
 - `RLIST STARTED RSEAPI.* ALL STDATA`
- RSE API as a secure z/OS UNIX server
 - `RLIST FACILITY BPX.SERVER ALL`
- PassTicket support for RSE API
 - `RLIST PTKTDATA IRRPTAUTH.FEKAPPL.* ALL`
- Data set profiles
 - `LISTGRP HUH`
 - `LISTDSD PREFIX(HUH) ALL`

Chapter 4. Operator commands

This chapter provides an overview of the available operator (or console) commands for RSE API. If you are unfamiliar with the syntax diagrams used to explain the command format, see [How to read a syntax diagram](#).

Start (S)

Use the **START** command to dynamically start a started task (STC). The abbreviated version of the command is the letter S.

RSEAPI

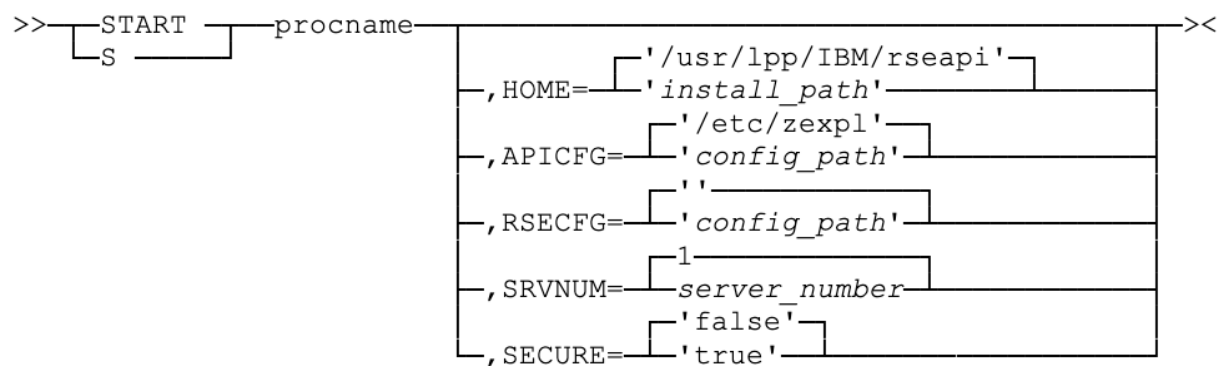


Figure 4. START RSEAPI operator command

procname

The name of the member in a procedure library that is used to start the server. The default name used during the host system configuration is RSEAPI.

HOME='install_path'

Path prefix and the mandatory `usr/lpp/IBM/rseapi` used to install RSE API. The default is `' /usr/lpp/IBM/rseapi '`. The z/OS UNIX path is case-sensitive and must be enclosed in single quotation marks (') to preserve lowercase characters.

APICFG='config_path'

Absolute location of the RSE API configuration files stored in z/OS UNIX. The default is `' /etc/zexpl '`. The z/OS UNIX path is case-sensitive and must be enclosed in single quotation marks (') to preserve lowercase characters.

RSECFG='config_path'

Absolute location of the `rse.env` configuration file. The default is `' '`, which results in using the same value as specified for variable `APICFG`. The z/OS UNIX path is case-sensitive and must be enclosed in single quotation marks (') to preserve lowercase characters. `rse.env` is a configuration file that belongs to requisite product IBM Explorer for z/OS.

SRVNUM=server_number

The value corresponds to the number in the variable `RSEAPI_PORT_HTTP_<num>` in `rseapi.env`. The default is 1. For example, `RSEAPI_PORT_HTTP_3` is used when launching `SRVNUM=3`.

SECURE='false'

The option to start RSE API via HTTPS encrypted communication. The default is `' false '`. The value of this variable can only be `' true '` or `' false '`.

How to read a syntax diagram

The syntax diagram shows you how to specify a command so that the operating system can correctly interpret what you type. Read the syntax diagram from left to right and from top to bottom, following the horizontal line, which is the main path.

Symbols

The following symbols are used in syntax diagrams:

Symbol	Description
>>	Marks the beginning of the syntax diagram.
>	Indicates that the syntax diagram is continued.
	Marks the beginning and end of a fragment or part of the syntax diagram.
><	Marks the end of the syntax diagram.

Operands

The following types of operands are used in syntax diagrams:

- Required operands are displayed on the main path line:

```
>>—REQUIRED_OPERAND—><
```

- Optional operands are displayed below the main path line:

```
>>└──OPTIONAL_OPERAND──┘><
```

- Default operands are displayed above the main path line:

```
>>┐──DEFAULT_OPERAND──┘><
```

Operands are classified as keywords or variables:

- Keywords are constants that must be provided. If the keyword appears in the syntax diagram in both uppercase and lowercase, the uppercase portion is the abbreviation for the keyword; for example, KEYword. Keywords are not case-sensitive.
- Variables are italicized, appear in lowercase letters, and represent names or values you supply. For example, a data set name is a variable. Variables can be case-sensitive.

Syntax example

In the following example, the USER command is a keyword. The required variable parameter is `user_id`, and the optional variable parameter is `password`. Replace the variable parameters with your own values:

```
>>—USER—user_id└──password──┘><
```

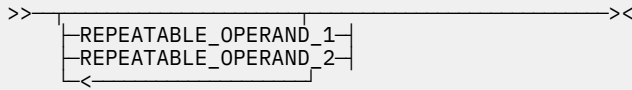
Nonalphanumeric characters and blank spaces

If a diagram shows a character that is not alphanumeric, such as parentheses, periods, commas, equal signs, and blank spaces, you must code the character as part of the syntax. In this example, you must code `OPERAND=(001 0.001)`:

```
>>—OPERAND—==(001 0.001)—><
```

Selecting more than one operand

An arrow returning to the left in a group of operands means that more than one can be selected, or that a single one can be repeated:



Longer than one line

If a diagram is longer than one line, the first line ends with a single arrowhead and the second line begins with a single arrowhead:

```
>>—| The first line of a syntax diagram that is longer than one line |—>  
>—| The continuation of the subcommands, parameters, or both |————><
```

Syntax fragments

Some diagrams might contain syntax fragments, which serve to break up diagrams that are too long, too complex, or too repetitious. Syntax fragment names are in mixed case and are shown in the diagram and in the heading of the fragment. The fragment is placed below the main diagram:

```
>>—| Syntax fragment |————><  
  
Syntax fragment:  
|—1ST_OPERAND—,—2ND_OPERAND—,—3RD_OPERAND—|
```


Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

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

IBM Director of Licensing IBM Corporation North Castle Drive Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing Legal and Intellectual Property Law IBM Japan, Ltd. 19-21, Nihonbashi-Hakozakicho, Chuo-ku Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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

Intellectual Property Dept. for Rational Software IBM Corporation Silicon Valley Lab 555 Bailey Avenue San Jose, CA 95141-1003 U.S.A.

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

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated

through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Copyright license

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 1992, 2017.

If you are viewing this information in softcopy, the photographs and color illustrations may not appear.

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Terms and conditions for product documentation

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Copyright license

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademark acknowledgments

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at www.ibm.com/legal/copytrade.shtml.

Adobe and PostScript are trademarks of Adobe Systems Incorporated.

Cell Broadband Engine - Sony Computer Entertainment Inc.

Rational® is a trademark of International Business Machines Corporation and Rational Software Corporation, in the United States, other countries, or both.

Intel, Intel Centrino, Intel SpeedStep, Intel Xeon, Celeron, Itanium, and Pentium are trademarks of Intel Corporation in the United States, or other countries, or both.

IT Infrastructure Library is a trademark of Central Computer and Telecommunications Agency

ITIL is a trademark of The Minister for the Cabinet Office

Linear Tape-Open, LTO, and Ultrium are trademarks of HP, IBM Corp., and Quantum

Linux is a trademark of Linus Torvalds

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

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

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



Part Number:

SC27-9905-00



(1P) P/N: