

IBM Rational Developer for System z
Version 8.0.3

Host Configuration Quick Start Guide



IBM Rational Developer for System z
Version 8.0.3

Host Configuration Quick Start Guide



Note

Before using this document, read the general information under "Documentation notices for IBM Rational Developer for System z" on page 49.

Fifth edition (October 2011)

This edition applies to IBM Rational Developer for System z Version 8.0.3 (program number 5724-T07) and to all subsequent releases and modifications until otherwise indicated in new editions.

Order publications by phone or fax. IBM Software Manufacturing Solutions takes publication orders between 8:30 a.m. and 7:00 p.m. eastern standard time (EST). The phone number is (800) 879-2755. The fax number is (800) 445-9269. Faxes should be sent Attn: Publications, 3rd floor.

You can also order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

IBM welcomes your comments. You can send your comments by mail to the following address:

IBM Corporation
Attn: Information Development Department 53NA
Building 501 P.O. Box 12195
Research Triangle Park NC 27709-2195
USA

You can fax your comments to: 1-800-227-5088 (US and Canada)

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.

Note to U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

© Copyright IBM Corporation 2000, 2011.

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 book	ix
Who should read this book	ix

Chapter 1. Planning.	1
Requisites	1
Requisite products	1
Required resources	1
Client user ID	2
Servers	2
Configuration method	2
Deployment and upgrade	3

Chapter 2. Basic customization.	5
Basic customization	5
Customization setup	5
PARMLIB changes	5
Set z/OS UNIX limits in BPXPRMxx	6
Add started tasks to COMMNDxx	6
APF authorizations in PROGxx	6
Requisite LINKLIST and LPA definitions	7
PROCLIB changes	7
JES Job Monitor	7
RSE daemon	8
Lock daemon	8
ELAXF* remote build procedures	9
Security definitions	10
FEJJCNFG, JES Job Monitor configuration file	11
rsed.envvars, RSE configuration file	11
ISPF.conf, ISPF's TSO/ISPF Client Gateway configuration file	15

Chapter 3. Installation verification	17
Verify started tasks	17
JMON, JES, Job Monitor	17
LOCKD, Lock daemon	17
RSED, RSE daemon.	17
IVP operator commands	17
PassTicket reusability	17
RSE daemon connection	17
ISPF Client Gateway	17

Appendix A. Security definitions.	19
Activate security settings and classes	19
Define an OMVS segment for Developer for System z users	21
Define data set profiles	21
Define the Developer for System z started tasks	21
Define JES command security	22
Define RSE as a secure z/OS UNIX server	23
Define MVS program controlled libraries for RSE.	23

Define application protection for RSE.	23
Define PassTicket support for RSE.	24
Define z/OS UNIX program controlled files for RSE	24
Verify security settings	25

Appendix B. Migration considerations	27
Version 8.0.x migration notes	27
Migrate from version 7.6 to version 8.0.1	28
IBM Rational Developer for System z, FMID HHOP801	28

Appendix C. Operator commands	31
Modify (F).	31
JES Job Monitor	31
RSE daemon	32
Lock daemon	37

Appendix D. Optional customization	39
(Optional) Common Access Repository Manager (CARMA)	39
(Optional) SCLM Developer Toolkit	39
(Optional) Application Deployment Manager	40
(Optional) pushtoclient.properties, Host-based client control	40
(Optional) (deprecated) FMIEXT.properties, File Manager integration	41
(Optional) ssl.properties, RSE SSL encryption	41
(Optional) rsecomm.properties, RSE tracing.	41
(Optional) DB2 stored procedure	41
(Optional) CICS bidirectional language support	42
(Optional) Diagnostic IRZ error messages	42
(Optional) WORKAREA and /tmp cleanup.	42
(Optional) Using REXEC (or SSH)	43

Appendix E. Host Configuration Reference	45
Understanding Developer for System z	45
Security considerations	45
TCP/IP considerations.	45
WLM considerations	45
Tuning considerations	45
Performance considerations	45
Push-to-client considerations	46
CICSTS considerations.	46
Customizing the TSO environment	46
Running multiple instances	46
Troubleshooting configuration problems	46
Setting up SSL and X.509 authentication.	46
Setting up TCP/IP	47

Documentation notices for IBM Rational Developer for System z.	49
Copyright license	50
Trademark acknowledgments	50

Index	53
------------------------	-----------

Figures

1. JMON - JES Job Monitor started task	8	7. ISPF.conf - ISPF configuration file	15
2. RSED - RSE daemon started task.	8	8. MODIFY JMON operator command	31
3. LOCKD - Lock daemon started task.	9	9. MODIFY RSED operator command	32
4. FEJJCNFG - JES Job Monitor configuration file	11	10. MODIFY LOCKD operator command	37
5. rsed.envvars - RSE configuration file	12		
6. rsed.envvars - RSE configuration file (continued).	13		

Tables

1.	Required resources	1	4.	ELAXF* high-level qualifier checklist	10
2.	Administrators needed for required tasks	2	5.	Security setup variables	19
3.	Sample ELAXF* procedures	9	6.	Thread pool error status	33

About this book

This document discusses the configuration of the IBM Rational Developer for System z functions. It includes brief instructions on how to configure IBM Rational Developer for System z Version 8.0.3 on your z/OS[®] host system. For complete details on the configuration of this product, refer to *Rational Developer for System z Host Configuration Guide* (SC23-7658).

The following names are used in this manual:

- *IBM Rational Developer for System z* is called *Developer for System z*.
- *Common Access Repository Manager* is abbreviated to CARMA.
- *Software Configuration and Library Manager Developer Toolkit* is called *SCLM Developer Toolkit*, abbreviated to *SCLMDT*.
- *z/OS UNIX System Services* is called *z/OS UNIX*.
- *Customer Information Control System Transaction Server* is called *CICSTS*, abbreviated to *CICS*[®].

For earlier releases, including IBM WebSphere Developer for System z, IBM WebSphere Developer for zSeries, and IBM[®] WebSphere Studio Enterprise Developer, use the configuration information found in the *Host Configuration Guide* and *Program Directories* for those releases.

This document is part of a set of documents that describe Developer for System z host configuration. Each of these documents has a specific target audience. You do not have to read all documents to complete the Developer for System z configuration.

- *Rational[®] Developer for System z[®] Host Configuration Guide* (SC23-7658) describes in detail all planning tasks, configuration tasks, and options (including optional ones) and provides alternative scenarios.
- *Rational Developer for System z Host Configuration Reference* (SC14-7290) describes Developer for System z design and gives background information for various configuration tasks of Developer for System z, z/OS components, and other products (such as WLM and CICS) related to Developer for System z.
- *Rational Developer for System z Host Configuration Quick Start Guide* (GI11-9201) describes a minimal setup of Developer for System z.
- *Rational Developer for System z Host Configuration Utility* (SC14-7282) describes the Host Configuration Utility, an ISPF panel application that guides you through basic and common optional customization steps for Developer for System z.

The information in this document applies to all Rational Developer for System z Version 8.0.3 packages including IBM Rational Developer for zEnterprise[™].

Who should read this book

This document is intended for system programmers installing and configuring IBM Rational Developer for System z Version 8.0.3.

It lists and briefly documents the different steps needed to do a basic setup of the products. Refer to *Rational Developer for System z Host Configuration Guide* (SC23-7658) for more details on the listed actions and non-default settings.

To use this book, you need to be familiar with the z/OS UNIX System Services and MVS[™] host systems.

Chapter 1. Planning

This section summarizes the host component installation and configuration information in *Rational Developer for System z Host Configuration Guide* (SC23-7658). Refer to that publication for more details on these subjects:

- Migration considerations
- Planning considerations
- Preinstallation considerations
- Pre-configuration considerations
- Predeployment considerations
- Client checklist

Requisites

Requisite products

Developer for System z 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 Developer for System z. These requisites must be installed and operational at runtime for the corresponding feature to work as designed.

Refer to *Rational Developer for System z Prerequisites* (SC23-7659) to get a complete list of prerequisites and corequisites for your version of Developer for System z. The key requisites for a basic setup are the following:

- z/OS 1.8 or higher
- ISPF APAR OA35689 (TSO/ISPF Client Gateway)
- Java 5.0 or higher

Required resources

Basic setup for Developer for System z requires the allocation of the systems resources listed in Table 1.

Table 1. Required resources

Resource	Default value
APF authorized data set	FEK.SFEKAUTH
started task	JMON, RSED, and LOCKD
port for host-confined communication	6715 and 4036
port for client-host communication	4035
port range for client-host communication	any available port is used
z/OS UNIX server security definition	UPDATE permission to BPX.SERVER for RSED started task
PassTicket security definitions	no default

Table 2 on page 2 lists the administrators needed for basic customization tasks.

Table 2. Administrators needed for required tasks

Administrator	Task
System	Typical system programmer actions are required for all customization tasks
Security	<ul style="list-style-type: none">• Define OMVS segment for Developer for System z users• Define data set profiles• Define started tasks• Define operator command security• Define z/OS UNIX server profiles• Define application security• Define PassTicket support• Define program controlled data sets• Define program controlled z/OS UNIX files
TCP/IP	Define new TCP/IP ports
WLM	Assign started task goals to the servers and their child processes

Client user ID

The user ID of a Developer for System z user must have the following attributes:

- TSO access (with a normal region size)
- An OMVS segment defined in the security software (for both user ID and default group):
 - A valid z/OS UNIX user ID, UID 0 is not required
 - HOME directory with read, write, and execute access
 - PROGRAM must point to a valid z/OS UNIX shell, such as /bin/sh
 - ASSIZEMAX should be blank (use system defaults)
- Read and execute access to Developer for System z directories and files
- Read access to Developer for System z data sets

Servers

Developer for System z consists of the following permanently active servers, which can be started tasks or user jobs. These servers provide the requested services themselves, or start other servers (as z/OS UNIX threads or user jobs) to provide the service:

- JES Job Monitor (JMON) server provides all JES-related services.
- Lock Daemon (LOCKD) provides tracking services for data set locks.
- Remote Systems Explorer (RSE) server provides core services such as connecting the client to the host and starting other servers for specific services.

Configuration method

Developer for System z provides alternative methods to configure the host side of the product. This gives you a choice of the following methods:

- Using the ISPF panel application. This guides you through the required customization steps and selected optional customization steps. For more information, refer to the *Host Configuration Utility* (SC14-7282).
- Using the *Host Configuration Quick Start Guide*(GI11-9201). This guides you through the required customization steps. The scope of this guide is limited to a basic setup.
- Using the *Host Configuration Guide*(SC23-7658). This guides you through the required customization steps and all optional customization steps. All configurable options are covered in this guide, including some non-default scenarios

Deployment and upgrade

Developer for System z supports installing the products once, followed by copying a minimal set of necessary data sets and directories to other systems for deployment. Running multiple copies (same or different version) of the products on the same system is also supported.

During an upgrade, make backups of all configuration files BEFORE installing the product, as the install might overlay the files.

Chapter 2. Basic customization

This chapter summarizes the basic customization information in *Rational Developer for System z Host Configuration Guide* (SC23-7658). Refer to that publication for more details.

Basic customization

This chapter summarizes the basic customization information in *Rational Developer for System z Host Configuration Guide* (SC23-7658). Refer to that publication for more details.

Customization setup

Developer for System z comes with several sample configuration files and sample JCL. To avoid overwriting your customizations when applying maintenance, you should copy all these members and z/OS UNIX files to a different location and to customize the copy.

Some functions of Developer for System z also 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, FEKSETUP, is provided to create the copies and the required directories.

Note: The *Rational Developer for System z Host Configuration Utility Guide* (SC14-7282) describes the host configuration using the Host Configuration Utility. The FEKSETUP job and the utility do some of the same tasks, with no way of checking to see if those tasks have already been performed. Therefore it is possible to undo changes that have already been made. For this reason, you should not use both methods for a single installation.

Customize and submit sample member FEKSETUP in data set FEK.SFEKSAMP to create customizable copies of configuration files and configuration JCL, and to create required z/OS UNIX directories. The required customization steps are described within the member.

This job performs the following tasks:

- Create FEK.#CUST.PARMLIB and populate it with sample configuration files.
- Create FEK.#CUST.PROCLIB and populate it with sample SYS1.PROCLIB members.
- Create FEK.#CUST.JCL and populate it with sample configuration JCL.
- Create FEK.#CUST.CNTL and populate it with sample server startup scripts.
- Create FEK.#CUST.ASM and populate it with sample assembler source code.
- Create FEK.#CUST.COBOL and populate it with sample COBOL source code.
- Create /etc/rdz/* and populate it with sample configuration files.
- Create /var/rdz/* as work directories for various Developer for System z functions, and populate it with sample files.

PARMLIB changes

Note that the listed PARMLIB changes are for a basic setup only. More changes are needed if you select to use certain optional functions.

Set 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.

MAXTHREADS specifies the maximum number of active threads for a single process. Set MAXTHREADS in SYS1.PARMLIB(BPXPRMxx) to 1500 or higher.

MAXTHREADTASKS specifies the maximum number of active MVS tasks for a single process. Set MAXTHREADTASKS in SYS1.PARMLIB(BPXPRMxx) to 1500 or higher.

MAXPROCUSER specifies the maximum number of processes that a single z/OS UNIX user ID can have concurrently active. Set MAXPROCUSER in SYS1.PARMLIB(BPXPRMxx) to 50 or higher.

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
- SETOMVS MAXPROCUSER=50

Add started tasks to COMMNDxx

Add start commands for the Developer for System z RSE and JMON servers to SYS1.PARMLIB(COMMANDxx) to start them automatically at next system IPL.

Once the servers are defined and configured, they can be started dynamically with the following console commands:

- S RSED
- S LOCKD
- S JMON

APF authorizations in PROGxx

In order for JES Job Monitor to access JES spool files, module FEJJMON in the FEK.SFEKAUTH load library and the Language Environment® (LE) runtime libraries (CEE.SCEERUN*) must be APF authorized.

In order for ISPF to create the TSO/ISPF Client Gateway, module ISPZTS0 in SYS1.LINKLIB must be APF authorized.

APF authorizations are defined in SYS1.PARMLIB(PROGxx), if your site followed IBM recommendations.

APF authorizations can be set dynamically (until the next IPL) with the following console commands, where volser is the volume on which the data set resides if it is not SMS managed:

- SETPROG APF,ADD,DSN=FEK.SFEKAUTH,SMS
- SETPROG APF,ADD,DSN=CEE.SCEERUN,VOL=volser
- SETPROG APF,ADD,DSN=CEE.SCEERUN2,VOL=volser
- SETPROG APF,ADD,DSN=SYS1.LINKLIB,VOL=volser

Requisite LINKLIST and LPA definitions

The RSE server is a z/OS UNIX process that requires access to MVS load libraries. The following (prerequisite) libraries must be made available, either through STEPLIB or LINKLIST/LPALIB:

- System load library
 - SYS1.LINKLIB
- Language Environment's runtime
 - CEE.SCEERUN
 - CEE.SCEERUN2
- C++'s DLL class library
 - CBC.SCLBDLL
- ISPF's TSO/ISPF Client Gateway
 - ISP.SISPLoad
 - ISP.SISPLPA

LINKLIST data sets are defined in SYS1.PARMLIB(PROGxx), if your site followed IBM recommendations. LPA data sets are defined in SYS1.PARMLIB(LPALSTxx).

If you opt to use STEPLIB, you must define the libraries not available through LINKLIST/LPALIB in the STEPLIB directive of rsed.envvars, the RSE configuration file. Be aware, however, that:

- Using STEPLIB in z/OS UNIX has a negative performance impact.
- If one STEPLIB library is APF authorized, then all must be authorized. Libraries lose their APF authorization when they are mixed with non-authorized libraries in STEPLIB.
- Libraries added to the STEPLIB DD in a JCL are not propagated to the z/OS UNIX processes started by the JCL.

PROCLIB changes

The started task and remote build procedures listed below must reside in a system procedure library defined to your JES subsystem. In the instructions below, the IBM default procedure library, SYS1.PROCLIB, is used.

JES Job Monitor

Customize the sample started task member FEK.#CUST.PROCLIB(JMON), as described within the member, and copy it to SYS1.PROCLIB. As shown in the code sample below, you have to provide the following:

- The high-level qualifier of the load library, default FEK
- The JES Job Monitor configuration file, default FEK.#CUST.PARMLIB(FEJJCNFG)

```

//*
//* JES JOB MONITOR
//*
//JMON      PROC PRM=,                * PRM='-TV' TO START TRACING
//          LEPRM='RPTOPTS(ON)',
//          HLQ=FEK,
//          CFG=FEK.#CUST.PARMLIB(FEJJCNFG)
//*
//JMON      EXEC PGM=FEJJMON,REGION=0M,TIME=NOLIMIT,
//          PARM=('&LEPRM,ENVAR("_CEE_ENVFILE_S=DD:ENVIRON")/&PRM')
//STEPLIB   DD DISP=SHR,DSN=&HLQ..SF&KAUTH
//ENVIRON   DD DISP=SHR,DSN=&CFG
//SYSPRINT  DD SYSOUT=*
//SYSOUT    DD SYSOUT=*
//          PEND
//

```

Figure 1. JMON - JES Job Monitor started task

RSE daemon

Customize the sample started task member FEK.#CUST.PROCLIB(JMON), as described within the member, and copy it to SYS1.PROCLIB. As shown in the code sample below, you have to provide the following:

- The home directory where Developer for System z is installed, default /usr/lpp/rdz
- The location of the configuration files, default /etc/rdz

```

//*
//* RSE DAEMON
//*
//RSED      PROC TMPDIR=,
//          PORT=,
//          IVP=,                    * 'IVP' to do an IVP test
//          CNFG='/etc/rdz',
//          HOME='/usr/lpp/rdz'
//*
//RSED      EXEC PGM=BPXBATSL,REGION=0M,TIME=NOLIMIT,
//          PARM='PGM &HOME./bin/rsed.sh &IVP -C&CNFG -P&PORT -T&TMPDIR'
//STDOUT    DD SYSOUT=*
//STDERR    DD SYSOUT=*
//          PEND
//

```

Figure 2. RSED - RSE daemon started task

Lock daemon

Customize the sample started task member FEK.#CUST.PROCLIB(LOCKD), as described within the member, and copy it to SYS1.PROCLIB. As shown in the code sample below, you have to provide the following:

- The home directory where Developer for System z is installed, default /usr/lpp/rdz.
- The location of the configuration files, default /etc/rdz.

```

/**
/** LOCK DAEMON
/**
//LOCKD   PROC TMPDIR=,
//          LOG=,
//          CNFG='/etc/rdz',
//          HOME='/usr/lpp/rdz'
/**
//LOCKD   EXEC PGM=BPXBATSL,REGION=0M,TIME=NOLIMIT,
//          PARM='PGM &HOME./bin/lockd.sh -C&CNFG -L&LOG -T&TMPDIR'
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//          PEND
/**

```

Figure 3. LOCKD - Lock daemon started task

ELAXF* remote build procedures

Developer for System z provides sample JCL procedures that can be used for the JCL generation, remote project builds and remote syntax check features of CICS BMS maps, IMS™ MFS screens and COBOL, PL/I, Assembler and C/C++ programs. These procedures allow installations to apply their own standards, and ensure that developers use the same procedures with the same compiler options and compiler levels.

The sample procedures and their function are listed in Table 3.

Table 3. Sample ELAXF* procedures

Member	Purpose
ELAXFADT	Sample procedure for assembling and debugging High Level assembler programs.
ELAXFASM	Sample procedure for assembling High Level assembler programs.
ELAXFBMS	Sample procedure for creating CICS BMS object and corresponding copy, dsect, or include member.
ELAXFCOC	Sample procedure for doing COBOL Compiles, Integrated CICS translate and integrated DB2® translate.
ELAXFCOP	Sample procedure for doing DB2 preprocess of EXEC SQL statements embedded in COBOL programs.
ELAXFCOT	Sample procedure for doing CICS translation for EXEC CICS statements embedded in COBOL programs.
ELAXFCPC	Sample procedure for doing C compiles.
ELAXFCPP	Sample procedure for doing C++ compiles.
ELAXFCP1	Sample procedure for COBOL compiles with SCM preprocessor statements (-INC and ++INCLUDE).
ELAXFDCL	Sample procedure for running a program in TSO mode.
ELAXFGO	Sample procedure for the GO step.
ELAXFLNK	Sample procedure for linking C/C++, COBOL, PLI and High Level Assembler programs.
ELAXFMFS	Sample procedure for creating IMS MFS screens.
ELAXFPLP	Sample procedure for doing DB2 preprocess of EXEC SQL statements embedded in PLI programs.

Table 3. Sample ELAXF* procedures (continued)

Member	Purpose
ELAXFPLT	Sample procedure for doing CICS translation of EXEC CICS statements embedded in PLI programs.
ELAXFPL1	Sample procedure for doing PL/I compiles, integrated CICS translate and integrated DB2 translate.
ELAXFPP1	Sample procedure for PL/I compiles with SCM preprocessor statements (-INC and ++INCLUDE).
ELAXFTSO	Sample procedure for running/debugging generated DB2 code in TSO mode.
ELAXFUOP	Sample procedure for generating the UOPT step when building programs that run in CICS or IMS subsystems.

Customize the sample build procedure members, FEK.#CUST.PROCLIB(ELAXF*), as described within the members, and copy them to SYS1.PROCLIB. You have to provide the correct high-level qualifiers for different product libraries, as described in Table 4.

Table 4. ELAXF* high-level qualifier checklist

Product	Default HLQ	Value
Rational Developer for System z	FEK	
CICS	CICSTS32.CICS	
DB2	DSN910	
IMS	IMS	
COBOL	IGY.V4R1M0	
PL/I	IBMZ.V3R8M0	
C/C++	CBC	
LE	CEE	
system LINKLIB	SYS1	
system MACLIB	SYS1	

If the ELAXF* procedures cannot be copied into a system procedure library, ask the Developer for System z users to add a JCLLIB card (right after the JOB card) to the job properties on the client.

```
//MYJOB    JOB <job parameters>
//PROCS    JCLLIB ORDER=(FEK.#CUST.PROCLIB)
```

Security definitions

Customize and submit sample member FEKRACF in data set FEK.#CUST.JCL to create the security definitions for Developer for System z. The user submitting this job must have security administrator privileges, such as being RACF® SPECIAL. Refer to Appendix A, "Security definitions," on page 19 for more details.

Note:

- For those sites that use CA ACF2 TM for z/OS or CA Top Secret® for z/OS, please refer to your product page on the CA support site (<https://support.ca.com>) and check for the related Developer for System z Knowledge

Document. This Knowledge Document has details on the security commands necessary to properly configure Developer for System z.

- The sample FEKRACF job holds more than just RACF commands. The last step of the security definitions consists of making a z/OS UNIX file program controlled. Depending on the policies at your site, this might be a task for the system programmer and not the security administrator.

FEJJCENFG, JES Job Monitor configuration file

JES Job Monitor (JMON) provides all JES-related services. The behavior of JES Job Monitor can be controlled with the definitions in FEJJCENFG.

FEJJCENFG is located in FEK.#CUST.PARMLIB, unless you specified a different location when you customized and submitted job FEK.SFEKSAMP(FEKSETUP). See “Customization setup” on page 5 for more details.

Customize the sample JES Job Monitor configuration member FEJJCENFG, as shown in the following code sample:

```
SERV_PORT=6715
TZ=EST5EDT
# BPXK SETIBMOPT_TRANSPORT=TCPIP
#APPLID=FEKAPPL
#AUTHMETHOD=SAF
#CODEPAGE=UTF-8
#CONCHAR=$
#CONSOLE_NAME=JMON
#GEN_CONSOLE_NAME=OFF
#HOST_CODEPAGE=IBM-1047
#LIMIT_COMMANDS=NOLIMIT
#LIMIT_VIEW=USERID
#LISTEN_QUEUE_LENGTH=5
#MAX_DATASETS=32
#MAX_THREADS=200
#TIMEOUT=3600
#TIMEOUT_INTERVAL=1200
#SUBMIT_TIMEOUT=30
#SUBMITMETHOD=TSO
#TSO_TEMPLATE=FEK.#CUST.CNTL(FEJTSO)
```

Figure 4. FEJJCENFG - JES Job Monitor configuration file

SERV_PORT

The port number for JES Job Monitor host server. The default port is 6715. Can be changed if desired. This value must match the port number set for JES Job Monitor in the rsed.envvars configuration file. If these values differ, RSE cannot connect the client to JES Job Monitor.

TZ Time zone selector. The default is EST5EDT. The default time zone is UTC +5 hours (Eastern Standard Time (EST) Eastern Daylight Savings Time (EDT)). Change this to represent your time zone.

rsed.envvars, RSE configuration file

The RSE lock daemon and the RSE server processes (RSE daemon, RSE thread pool, and RSE server) use the definitions in rsed.envvars. Optional Developer for System z and third-party services can use this configuration file also to define environment variables for their use.

rsed.envvars is located in /etc/rdz/, unless you specified a different location when you customized and submitted job FEK.SFEKSAMP(FEKSETUP). See “Customization setup” on page 5 for more details. You can edit the file with the TSO OEDIT command.

See the following sample rsed.envvars file, which must be customized to match your system environment.

```
#=====
# (1) required definitions
JAVA_HOME=/usr/lpp/java/J5.0
RSE_HOME=/usr/lpp/rdz
_RSE_RSED_PORT=4035
_RSE_LOCKD_PORT=4036
_RSE_JMON_PORT=6715
_RSE_HOST_CODEPAGE=IBM-1047
TZ=EST5EDT
LANG=C
PATH=/bin:/usr/sbin
_CEE_DMPTARG=/tmp
STEPLIB=NONE
_RSE_JAVAOPTS=""
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Xms1m -Xmx256m"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Ddaemon.log=/var/rdz/logs"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Duser.log=/var/rdz/logs"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DDSTORE_LOG_DIRECTORY="
#=====
# (2) required definitions for TSO/ISPF Client Gateway
_CMDSERV_BASE_HOME=/usr/lpp/ispf
_CMDSERV_CONF_HOME=/etc/rdz
_CMDSERV_WORK_HOME=/var/rdz
_RSE_CMDSERV_OPTS=""
#=====
# (3) required definitions for SCLM Developer Toolkit
#=====
# (4) optional definitions
# _RSE_PORTRANGE=8108-8118
#=====
```

Figure 5. rsed.envvars - RSE configuration file


```

# (5) do not change unless directed by IBM support center
_RSE_SAF_CLASS=/usr/include/java_classes/IRRRacf.jar
_CEE_RUNOPTS="ALL31(ON) HEAP(32M,32K,ANYWHERE,KEEP,,) TRAP(ON)"
_BPX_SHAREAS=YES
_BPX_SPAWN_SCRIPT=YES
_EDC_ADD_ERRNO2=1
JAVA_PROPAGATE=NO
RSE_LIB=$RSE_HOME/lib
PATH=.:$JAVA_HOME/bin:$RSE_HOME/bin:$CMDSEV_BASE_HOME/bin:$PATH
LIBPATH=$JAVA_HOME/bin:$JAVA_HOME/bin/classic:$RSE_LIB:$RSE_LIB/icuc
LIBPATH=.:usr/lib:$LIBPATH
CLASSPATH=$RSE_LIB:$RSE_LIB/dstore_core.jar:$RSE_LIB/clientserver.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/dstore_extra_server.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/zosserver.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/dstore_miners.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/universalminers.jar:$RSE_LIB/mvsminers.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/carma.jar:$RSE_LIB/luceneminer.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/mvsluceneminer.jar:$RSE_LIB/cdzminer.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/mvscdzminer.jar:$RSE_LIB/jesminers.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/FAMiner.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/mvsutil.jar:$RSE_LIB/jesutils.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/lucene-core-2.3.2.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/cdtparser.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/wdzBidi.jar:$RSE_LIB/fmiExtensions.jar
CLASSPATH=$CLASSPATH:$RSE_SAF_CLASS
CLASSPATH=.:$CLASSPATH
_RSE_CMDSEV_OPTS="&SESSION=SPAWN$_RSE_CMDSEV_OPTS"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DISPF_OPTS='$_RSE_CMDSEV_OPTS'"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DA_PLUGIN_PATH=$RSE_LIB"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Xbootclasspath/p:$RSE_LIB/bidiTools.jar"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dfile.encoding=$_RSE_HOST_CODEPAGE"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dconsole.encoding=$_RSE_HOST_CODEPAGE"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDSTORE_SPIRIT_ON=true"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DSPIRIT_EXPIRY_TIME=6"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DSPIRIT_INTERVAL_TIME=6"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dcom.ibm.cacheLocalHost=true"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Duser.home=$HOME"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dclient.username=$_RSE_USER_ID"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dlow.heap.usage.ratio=15"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dmaximum.heap.usage.ratio=40"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDSTORE_KEEPALIVE_ENABLED=true"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDSTORE_KEEPALIVE_RESPONSE_TIMEOUT=60000"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDSTORE_IO_SOCKET_READ_TIMEOUT=180000"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DRSECOMM_LOGFILE_MAX=0"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Djob.monitor.port=$_RSE_JMON_PORT"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dlock.daemon.port=$_RSE_LOCKD_PORT"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dlock.daemon.cleanup.interval=1440"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -showversion"
_RSE_SERVER_CLASS=org.eclipse.dstore.core.server.Server
_RSE_DAEMON_CLASS=com.ibm.etools.zos.server.RseDaemon
_RSE_POOL_SERVER_CLASS=com.ibm.etools.zos.server.ThreadPoolProcess
_RSE_LOCKD_CLASS=com.ibm.ftt.rse.mvs.server.miners.MVSLockDaemon
_RSE_SERVER_TIMEOUT=120000
_SCLMDT_BASE_HOME=$RSE_HOME
_SCLMDT_WORK_HOME=$CMDSEV_WORK_HOME
CGI_DTWORK=$ _SCLMDT_WORK_HOME
CGI_ISPCONF=$ _CMDSEV_CONF_HOME
CGI_ISPWORK=$ _CMDSEV_WORK_HOME
=====
# (6) additional environment variables

```

Figure 6. rsed.envvars - RSE configuration file (continued)

JAVA_HOME

Java home directory. The default is /usr/lpp/java/J5.0. Change to match your Java installation.

RSE_HOME

RSE home directory. The default is /usr/lpp/rdz. Change to match your Developer for System z installation.

_RSE_RSED_PORT

RSE daemon port number. The default is 4035. Can be changed if desired.

_RSE_LOCKD_PORT

RSE lock daemon port number. The default is 4036. Can be changed if desired.

_RSE_JMON_PORT

JES Job Monitor port number. The default is 6715. Can be changed if desired. This value must match the port number set for JES Job Monitor in the FEJJCNG configuration file. If these values differ, RSE cannot connect the client to JES Job Monitor.

_RSE_HOST_CODEPAGE

The host codepage. The default is IBM-1047. Change to match your host codepage.

TZ Time zone selector. The default is EST5EDT. The default time zone is UTC +5 hours (Eastern Standard Time (EST) Eastern Daylight Savings Time (EDT)). Change to match your time zone.

LANG

Specifies the name of the default locale. The default is C. C specifies the POSIX locale and (for example) Ja_JP specifies the Japanese locale. Change to match your locale.

STEPLIB

Access MVS data sets not in LINKLIST/LPALIB. The default is NONE.

You can bypass the need of having (prerequisite) libraries in LINKLIST/LPALIB by uncommenting and customizing one or more of the following STEPLIB directives:

```
STEPLIB=$STEPLIB:CEE.SCEERUN:CEE.SCEERUN2:CBC.SCLBDLL
STEPLIB=$STEPLIB:ISP.SISPLoad:ISP.SISPLPA:SYS1.LINKLIB
STEPLIB=$STEPLIB:FEK.SFEKAUTH:FEK.SFEKLOAD
```

Note:

- Using STEPLIB in z/OS UNIX has a negative performance impact.
- If one STEPLIB library is APF authorized, then all must be authorized. Libraries lose their APF authorization when they are mixed with non-authorized libraries in STEPLIB.
- Libraries that are designed for LPA placement might require additional program control or APF authorizations if they are accessed through LINKLIST or STEPLIB.
- Coding a STEPLIB DD statement in the server JCL does not set the requested STEPLIB concatenation.

_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Ddaemon.log=/var/rdz/logs"

Directory holding the RSE daemon and server logging and RSE audit data. The default is /var/rdz/logs. Change to enforce the desired location. If this directive is commented out, the home directory of the user ID assigned to RSE daemon will be used. The home directory is defined in the OMVS security segment of the user ID.

_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Duser.log=/var/rdz/logs"

Directory leading to the user-specific logs. The default is /var/rdz/logs.

Change to enforce the desired location. If this directive is commented out, the home directory of the client user ID will be used. The directory path is defined in the OMVS security segment of the user ID.

Note: The complete path to the user logs is `userlog/dstorelog/$LOGNAME/`, where `userlog` is the value of the `user.log` directive, `dstorelog` is the value of the `DSTORE_LOG_DIRECTORY` directive and `$LOGNAME` is the client's user ID in uppercase.

_CMDSERV_BASE_HOME

Home directory for the ISPF code that provides the TSO/ISPF Client Gateway service. The default is `/usr/lpp/ispf`. Change to match your ISPF installation.

_CMDSERV_CONF_HOME

ISPF base configuration directory. The default is `/etc/rdz`. Change to match the location of `ISPF.conf`, the TSO/ISPF Client Gateway customization file.

_CMDSERV_WORK_HOME

ISPF base work directory. The default is `/var/rdz`. Change to match the location of the `WORKAREA` directory used by the TSO/ISPF Client Gateway.

_RSE_PORTRANGE

Specifies the port range that the RSE server can open for communication with a client. Any port can be used by default. This is an optional directive.

ISPF.conf, ISPF's TSO/ISPF Client Gateway configuration file

ISPF's TSO/ISPF Client Gateway uses the definitions in `ISPF.conf` to create a valid environment to execute batch TSO and ISPF commands. Developer for System z uses this environment to run some MVS based services.

`ISPF.conf` is located in `/etc/rdz/`, unless you specified a different location when you customized and submitted job `FEK.SFEKSAMP(FEKSETUP)`. See "Customization setup" on page 5 for more details. You can edit the file with the TSO **OEDIT** command.

Comment lines start with an asterisk (*). When concatenating data set names, add them on the same line and separate the names with a comma (,).

Besides providing the correct names for the ISPF data sets, you must also add the TSO Commands service data set name, `FEK.SFEKPROC`, to the `SYSPROC` or `SYSEXEC` statement, as shown in the following code sample.

```
* REQUIRED:
sysproc=ISP.SISPCLIB,FEK.SFEKPROC
isplib=ISP.SISPMENU
isptlib=ISP.SISPTENU
ispplib=ISP.SISPPENU
ispslib=ISP.SISPSLIB
ispllib=ISP.SISPLLOAD

* OPTIONAL:
*allocjob = ISP.SISPSAMP(ISPZISP2)
*ISPF_timeout = 900
```

Figure 7. ISPF.conf - ISPF configuration file

Note: You can add your own DD-like statements and data set concatenations to customize the TSO environment, thus mimicking a TSO logon procedure.

Chapter 3. Installation verification

This chapter summarizes the installation verification information in *Rational Developer for System z Host Configuration Guide* (SC23-7658). Refer to that publication for more details.

Verify started tasks

JMON, JES, Job Monitor

Start the JMON started task (or user job). The startup information in DD STDOUT should end with the following message:

JM200I Server initialization complete.

If the job ends with return code 66, then FEK.SFEKAUTH is not APF authorized.

LOCKD, Lock daemon

Start the LOCKD started task (or user job). The lock daemon issues the following console message upon successful startup:

FEK501I Lock daemon started, port=4036, cleanup interval=1440, log level=1

RSED, RSE daemon

Start the RSED started task (or user job) with the IVP=IVP parameter. With this parameter, the server will end after doing some installation verification tests. The output of these tests is available in DD STDOUT. In case of certain errors, data will also be available in DD STDERR.

Note: Start the RSE daemon, without the IVP parameter, before continuing with the other IVP tests. RSE daemon issues the following console message upon successful startup:

FEK002I RseDaemon started. (port=4035)

IVP operator commands

PassTicket reusability

Developer for System z requires that the PassTickets it generates are reusable, because PassTicket generation is limited to one per user per second. Verify PassTicket reusability by executing the following operator command. Replace userid with a valid user ID.

```
MODIFY RSED,APPL=IVP PASSTICKET,userid
```

RSE daemon connection

Verify the RSE daemon connection by executing the following command. Replace userid with a valid user ID.

```
MODIFY RSED,APPL=IVP DAEMON,userid
```

ISPF Client Gateway

Verify the ISPF Client Gateway connection by executing the following command. Replace userid with a valid user ID.

```
|          MODIFY RSED,APPL=IVP ISPF,userid
```

```
|
```

Appendix A. Security definitions

Customize and submit sample member FEKRACF in data set FEK.#CUST.JCL, which has sample RACF and z/OS UNIX commands to create the basic security definitions for Developer for System z.

Note: For those sites that use CA ACF2™ for z/OS or CA Top Secret® for z/OS, please refer to your product page on the CA support site (<https://support.ca.com>) and check for the related Developer for System z Knowledge Document. This Knowledge Document has details on the security commands necessary to properly configure Developer for System z.

To complete the security setup, the security administrator needs to know the values listed in Table 5. These values were defined during previous steps of the installation and customization of Developer for System z.

Table 5. Security setup variables

Description	<ul style="list-style-type: none">• Default value• Where to find the answer	Value
Developer for System z product high level qualifier	<ul style="list-style-type: none">• FEK• SMP/E installation	
Developer for System z customization high level qualifier	<ul style="list-style-type: none">• FEK.#CUST• FEK.SFEKSAMP(FEKSETUP), as described in “Customization setup” on page 5	
JES Job Monitor started task name	<ul style="list-style-type: none">• JMON• FEK.#CUST.PROCLIB(JMON), as described in “PROCLIB changes” on page 7	
RSE daemon started task name	<ul style="list-style-type: none">• RSED• FEK.#CUST.PROCLIB(RSED), as described in “PROCLIB changes” on page 7	
Lock daemon started task name	<ul style="list-style-type: none">• LOCKD• FEK.#CUST.PROCLIB(LOCKD), as described in “PROCLIB changes” on page 7	

Activate security settings and classes

Developer for System z utilizes a variety of security mechanisms to ensure a secure and controlled host environment for the client. In order to do so, several classes and security settings must be active, as shown with the following sample RACF commands:

- Display current settings

—

SETROPTS LIST

- Activate facility class for z/OS UNIX and digital certificate profiles
 -
 - SETROPTS GENERIC(FACILITY)
 -
 - SETROPTS CLASSACT(FACILITY) RACLIST(FACILITY)
- Activate started task definitions
 -
 - SETROPTS GENERIC(STARTED)
 -
 - RDEFINE STARTED ** STDATA(USER(=MEMBER) GROUP(STCGROUP) TRACE(YES))
 -
 - SETROPTS CLASSACT(STARTED) RACLIST(STARTED)
- Activate console security for JES Job Monitor
 -
 - SETROPTS GENERIC(CONSOLE)
 -
 - SETROPTS CLASSACT(CONSOLE) RACLIST(CONSOLE)
- Activate operator command protection for JES Job Monitor
 -
 - SETROPTS GENERIC(OPERCMDS)
 -
 - SETROPTS CLASSACT(OPERCMDS) RACLIST(OPERCMDS)
- Activate application protection for RSE
 -
 - SETROPTS GENERIC(APPL)
 -
 - SETROPTS CLASSACT(APPL) RACLIST(APPL)
- Activate secured signon using PassTickets for RSE
 -
 - SETROPTS GENERIC(PKTCDATA)
 -
 - SETROPTS CLASSACT(PKTCDATA) RACLIST(PKTCDATA)
- Activate program control to ensure that only trusted code can be loaded by RSE
 -
 - RDEFINE PROGRAM ** ADDMEM('SYS1.CMDLIB'//NOPADCHK) UACC(READ)
 -
 - SETROPTS WHEN(PROGRAM)

Note: Do not create the ** profile if you already have a * profile in the PROGRAM class. It obscures and complicates the search path used by your security software. In this case, you must merge the existing * and the new ** definitions. IBM recommends to use the ** profile, as documented in *Security Server RACF Security Administrator's Guide* (SA22-7683).

Attention: Some products, such as FTP, require being program controlled if "WHEN PROGRAM" is active. Test this before activating it on a production system.

Define an OMVS segment for Developer for System z users

A RACF OMVS segment (or equivalent) that specifies a valid non-zero uid, home directory, and shell command must be defined for each user of Developer for System z. Their default group also requires an OMVS segment with a group id.

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

Define data set profiles

READ access for users and ALTER for system programmers suffices for most Developer for System z data sets.

You should protect FEK.SFEKAUTH and FEK.SFEKLPA against updates because these data sets are APF authorized.

- ```
ADDGROUP (FEK) OWNER(IBMUSER) SUPGROUP(SYS1)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z - HLQ STUB')
```
- ```
ADDSD 'FEK.*.***' UACC(READ)  
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
```
- ```
PERMIT 'FEK.*.***' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
```
- ```
SETROPTS GENERIC(DATASET) REFRESH
```

Define the Developer for System z started tasks

The following sample RACF commands create the JMON, RSED, and LOCKD started tasks, with protected user IDs (STCJMON, STCRSE, and STCLOCK, respectively) and group STCGROUP assigned to them. Replace the #group-id and #user-id-* placeholders with valid OMVS IDs.

- ```
ADDGROUP STCGROUP OMVS(GID(#group-id))
DATA('GROUP WITH OMVS SEGMENT FOR STARTED TASKS')
```
- ```
ADDUSER STCJMON DFLTGROUP(STCGROUP) NOPASSWORD NAME('RDZ - JES JOBMONITOR')  
OMVS(UID(#user-id-jmon) HOME(/tmp) PROGRAM(/bin/sh) NOASSIZEMAX  
NOTHREADSMAX)  
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
```
- ```
ADDUSER STCRSE DFLTGROUP(STCGROUP) NOPASSWORD NAME('RDZ - RSE
DAEMON')
OMVS(UID(#user-id-rse) HOME(/tmp) PROGRAM(/bin/sh)
ASSIZEMAX(2147483647) NOTHREADSMAX)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
```
-

```

ADDUSER STCLOCK DFLTGROUP(STCGROUP) NOPASSWORD NAME('RDZ - LOCK
DAEMON')
OMVS(UID(#user-id-lock) HOME(/tmp) PROGRAM(/bin/sh) NOASSIZEMAX
NOTHEADSMAX)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
•
RDEFINE STARTED JMON.* DATA('RDZ - JES JOBMONITOR')
STDATA(USER(STCJMON) GROUP(STCGROUP) TRUSTED(NO))
•
RDEFINE STARTED RSED.* DATA('RDZ - RSE DAEMON')
STDATA(USER(STCRSE) GROUP(STCGROUP) TRUSTED(NO))
•
RDEFINE STARTED LOCKD.* DATA('RDZ - LOCK DAEMON')
STDATA(USER(STCLOCK) GROUP(STCGROUP) TRUSTED(NO))
•
SETROPTS RACLIST(STARTED) REFRESH

```

**Note:**

- Ensure that the started tasks user IDs are protected by specifying the NOPASSWORD keyword.
- Ensure that RSE server has a unique OMVS uid due to the z/OS UNIX related privileges granted to this uid.

---

## Define JES command security

JES Job Monitor issues all JES operator commands requested by a user through an extended MCS (EMCS) console, whose name is controlled with the `CONSOLE_NAME` directive, as documented in “FEJJCENFG, JES Job Monitor configuration file” on page 11.

The following sample RACF commands give Developer for System z users conditional access to a limited set of JES commands (Hold, Release, Cancel, and Purge). Users only have execution permission if they issue the commands through JES Job monitor. Replace the `#console` placeholder with the actual console name.

```

•
RDEFINE OPERCMDS MVS.MCSOPER.#console UACC(READ)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
•
RDEFINE OPERCMDS JES%.* UACC(NONE)
•
PERMIT JES%.* CLASS(OPERCMDS) ACCESS(UPDATE) WHEN(CONSOLE(JMON)) ID(*)
•
SETROPTS RACLIST(OPERCMDS) REFRESH

```

**Note:**

- Usage of the console is permitted if no `MVS.MCSOPER.#console` profile is defined
- The `CONSOLE` class must be active for `WHEN(CONSOLE(JMON))` to work, but there is no actual profile check in the `CONSOLE` class for EMCS consoles.
- Do not replace `JMON` with the actual console name in the `WHEN(CONSOLE(JMON))` clause. The `JMON` keyword represents the point-of-entry application, not the console name.

**Attention:** Defining JES commands with universal access NONE in your security software might impact other applications and operations. Test this before activating it on a production system.

---

## Define RSE as a secure z/OS UNIX server

RSE requires UPDATE access to the BPX.SERVER profile to create/delete the security environment for the client's thread. If this profile is not defined, UID(0) is required for RSE.

- RDEFINE FACILITY BPX.SERVER UACC(NONE)
- PERMIT BPX.SERVER CLASS(FACILITY) ACCESS(UPDATE) ID(STCRSE)
- SETROPTS RACLIST(FACILITY) REFRESH

**Attention:** Defining the BPX.SERVER profile makes z/OS UNIX as a whole switch from UNIX level security to z/OS UNIX level security, which is more secure. This might impact other z/OS UNIX applications and operations. Test this before activating it on a production system.

---

## Define MVS program controlled libraries for RSE

Servers with authority to BPX.SERVER must run in a clean, program-controlled environment. This implies that all programs called by the RSE server must also be program controlled. For MVS load libraries, program control is managed by your security software.

- RALTER PROGRAM \*\* UACC(READ) ADDMEM('SYS1.LINKLIB'//NOPADCHK)
- RALTER PROGRAM \*\* UACC(READ) ADDMEM('CEE.SCEERUN'//NOPADCHK)
- RALTER PROGRAM \*\* UACC(READ) ADDMEM('CEE.SCEERUN2'//NOPADCHK)
- RALTER PROGRAM \*\* UACC(READ) ADDMEM('ISP.SISPLoad'//NOPADCHK)
- SETROPTS WHEN(PROGRAM) REFRESH

**Note:** Do not use the \*\* profile if you already have a \* profile in the PROGRAM class. It obscures and complicates the search path used by your security software. In this case, you must merge the existing \* and the new \*\* definitions. IBM recommends using the \*\* profile, as documented in *Security Server RACF Security Administrator's Guide* (SA22-7683).

---

## Define application protection for RSE

During client logon, RSE daemon verifies that a user is allowed to use the application.

- RDEFINE APPL FEKAPPL UACC(READ)  
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
- SETROPTS RACLIST(APPL) REFRESH

**Note:** The client connection request will only fail if the application ID is defined and the user lacks READ access to the profile.

---

## Define PassTicket support for RSE

The client's password (or other means of identification, such as an X.509 certificate) is only used to verify his identity upon connection. Afterwards, PassTickets are used to maintain thread security. PassTickets are system generated passwords with a lifespan of about 10 minutes. The generated PassTickets are based upon a secret key. This key is a 64 bit number (16 hex characters). Replace in the sample RACF commands below the key16 placeholder with a user-supplied 16 character hex string (characters 0-9 and A-F).

- RDEFINE PTKTDATA FEKAPPL UACC(NONE) SSIGNON(KEYMASKED(key16))  
APPLDATA('NO REPLAY PROTECTION – DO NOT CHANGE') DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
- RDEFINE PTKTDATA IRRPTAUTH.FEKAPPL.\* UACC(NONE) DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
- PERMIT IRRPTAUTH.FEKAPPL.\* CLASS(PTKTDATA) ACCESS(UPDATE) ID(STCRSE)
- SETROPTS RACLIST(PTKTDATA) REFRESH

**Note:** If the PTKTDATA class is already defined, verify that it is defined as a generic class before creating the profiles listed above. The support for generic characters in the PTKTDATA class is new since z/OS release 1.7, with the introduction of a Java interface to PassTickets.

**Attention:** The client connection request will fail if PassTickets are not set up correctly.

---

## Define z/OS UNIX program controlled files for RSE

Servers with authority to BPX.SERVER must run in a clean, program-controlled environment. This implies that all programs called by RSE server must also be program controlled. For z/OS UNIX files, program control is managed by the **extattr** command. To execute this command, you need READ access to BPX.FILEATTR.PROGCTL in the FACILITY class, or be UID(0).

- extattr +p /usr/lib/libIRRRacf\*.so

**Note:**

- Since z/OS 1.9, /usr/lib/libIRRRacf\*.so is installed program controlled during SMP/E RACF install.
- Since z/OS 1.10, /usr/lib/libIRRRacf\*.so is part of SAF, which ships with base z/OS, so it is available also to non-RACF customers.
- The setup might be different if you use a security product other than RACF. Consult the documentation of your security product for more information.
- The SMP/E install of Developer for System z sets the program control bit for internal RSE server programs.
- Use the **ls -Eog** z/OS UNIX command to display the current status of the program control bit. (The file is program controlled if the letter p shows in the second string.)

```
$ ls -Eog /usr/lib/libIRRRacf*.so
-rwxr-xr-x aps- 2 69632 Oct 5 2007 /usr/lib/libIRRRacf.so
-rwxr-xr-x aps- 2 69632 Oct 5 2007 /usr/lib/libIRRRacf64.so
```

---

## Verify security settings

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

- Security settings and classes
  - SETROPTS LIST
- OMVS segment for users
  - LISTUSER #userid NORACF OMVS
  - LISTGRP #group-name NORACF OMVS
- Data set profiles
  - LISTGRP FEK ALL
  - LISTDSD PREFIX(FEK) ALL
- Started tasks
  - LISTGRP STCGROUP OMVS
  - LISTUSER STCJMON OMVS
  - LISTUSER STCRSE OMVS
  - LISTUSER STCLOCK OMVS
  - RLIST STARTED JMON.\* ALL STDATA
  - RLIST STARTED RSED.\* ALL STDATA
  - RLIST STARTED LOCKD.\* ALL STDATA
- JES command security
  - RLIST CONSOLE JMON ALL
  - RLIST OPERCMDS MVS.MCSOPER.JMON ALL
  - RLIST OPERCMDS JES%.\*\* ALL
- RSE as a secure z/OS UNIX server
  - RLIST FACILITY BPX.SERVER ALL
- MVS program controlled libraries for RSE
  - RLIST PROGRAM \*\* ALL
- Application protection for RSE
  - RLIST APPL FEKAPPL ALL
- PassTicket support for RSE
  - RLIST PTKTDATA FEKAPPL ALL SSIGNON
  - RLIST PTKTDATA IRRPTAUTH.FEKAPPL.\* ALL
- z/OS UNIX program controlled files for RSE
  - ls -E /usr/lib/libIRRRacf\*.so



---

## Appendix B. Migration considerations

This appendix summarizes the migration information in *Rational Developer for System z Host Configuration Guide* (SC23-7658). Refer to that publication for more details.

---

### Version 8.0.x migration notes

The following migration notes are version 8.0.x-specific. They are valid for migration from IBM Rational Developer for System z version 8.0.1 and version 8.0.2 to version 8.0.3, and are additions to the existing version 8.0.1 migration notes.

- CARMA - The CRAMSG VSAM must be updated (since version 8.0.3).
- CARMA - The CRADEF and CRASTRS VSAM files for the CA Endeavor® SCM RAM must be updated to use the new support for CA Endeavor® SCM batch-actions (since version 8.0.3) and CA Endeavor® SCM packages (since version 8.0.3).
- CARMA - New sample members have been added (since version 8.0.3):
  - CRABCFG - configuration file for CA Endeavor® SCM batch-actions.
  - CRABATCA - sample job for CA Endeavor® SCM batch-actions.
- CARMA - The following customizable members have changed (since version 8.0.3):
  - CRANDVRA
- CARMA - Additional DD statements are added for the CA Endeavor® SCM RAM (since version 8.0.3):
  - CRABCFG
  - CRABSKEL
  - PKGSCLS (allocated by CRANDVRA)
- File Manager Integration is deprecated.
- Host Configuration Utility - Added migration option (since version 8.0.2)
- PROCLIB - The following PROCLIB members have changed (since version 8.0.3):
  - ELAXFUOP
- RSE - New operator commands have been added to the LOCKD started task (since version 8.0.2):
  - MODIFY DISPLAY TABLE
- RSE - New operator commands have been added to the RSED started task (since version 8.0.2 and 8.0.3):
  - MODIFY IVP ISPF,userid
  - MODIFY IVP PASSTICKET,userid
- RSE - RSED started task operator commands have been enhanced (since version 8.0.2):
  - MODIFY DISPLAY CLIENT [{,LOGON | ,ID | ,USER}]
- RSE - The following console messages are new (since version 8.0.3):
  - FEK910I
- RSE - New non-customizable directives have been added to rsed.envvars (since version 8.0.3):

- (\_RSE\_JAVAOPTS) -Dldap.server.address
- (\_RSE\_JAVAOPTS) -Dldap.server.port
- (\_RSE\_JAVAOPTS) -Dldap.ptc.group.name.suffix
- \_RSE\_PTC
- RSE - New optional directives have been added to rsed.envvars (since version 8.0.3):
  - (\_RSE\_JAVAOPTS) -Daudit.action
  - (\_RSE\_JAVAOPTS) -Daudit.action.id
  - (\_RSE\_JAVAOPTS) -DCPP\_CLEANUP\_INTERVAL
  - (\_RSE\_JAVAOPTS) -DDSTORE\_TCP\_NO\_DELAY
  - \_RSE\_FEK\_SAF\_CLASS
  - \_RSE\_LDAP\_SERVER
  - \_RSE\_LDAP\_PORT
  - \_RSE\_LDAP\_PTC\_GROUP\_SUFFIX
- RSE - Interpretation of the following optional directives in rsed.envvars has changed (since version 8.0.3):
  - (\_RSE\_JAVAOPTS) -Dprocess.cleanup.interval
- RSE - New optional directives have been added to pushtoclient.properties (since version 8.0.3):
  - accept.product.license
- RSE - Interpretation of the following optional directives in pushtoclient.properties has changed (since version 8.0.3):
  - config.enabled
  - product.enabled
  - reject.config.updates
  - reject.product.updates
- RSE - New z/OS UNIX samples have been added (since version 8.0.3):
  - process\_audit.rex
- New publication, *Rational Developer for System z Messages and Codes* (SC14-7497).

---

## Migrate from version 7.6 to version 8.0.1

These notes are for a migration from a base version 7.6 to version 8.0.1. It includes changes that are already documented as part of version 7.6 maintenance. The changes that are part of the maintenance stream (and thus possibly already implemented) are marked with the release where they were introduced.

### IBM Rational Developer for System z, FMID HHOP801

- The default SMP/E install location for MVS and z/OS UNIX components did not change and thus remain FEK.\* and /usr/lpp/rdz/\*.
- Application Deployment Manager - Existing ADN\* modules in the CICS RPL concatenation must be updated (since version 7.6.1).
- Application Deployment Manager - The following sample members have been updated to add URIMAP support in the Administrative utility (since version 7.6.1):
  - ADNJSPAU
  - ADNVCRD



- Application Deployment Manager - An existing CRD repository VSAM must be replaced to enable URIMAP support (since version 7.6.1).
- CARMA - Added support for an universal unique RAM ID stored in the CARMA definition VSAM data set, CRADEF (since version 7.6.1). An existing CRADEF VSAM must be replaced to use it.
- CARMA - Added support for a variable-length layout for the CARMA custom information VSAM data set, CRASTRS (since version 7.6.1). An existing CRASTRS VSAM must be replaced to use it.
- CARMA - New sample members have been added (since version 7.6.1):
  - CRA#VS2 - migrate CRASTRS to variable-length format.
- CARMA - Existing sample members have been renamed:
  - CRA#VCAD -> CRA\$VCAD
  - CRA#VCAS -> CRA\$VCAS
- CARMA - The CRADEF and CRASTRS VSAMs for the CA Endeavor<sup>®</sup> SCM RAM must be updated.
- CARMA - Additional DD statements are added for the CA Endeavor<sup>®</sup> SCM RAM (since version 7.6.1):
  - EXT2ELM
  - SPCLLIST (allocated by CRANDVRA)
- CARMA – The following customizable members have changed (since version 8.0.1).
  - CRANDVRA
  - CRASHOW
- JES Job Monitor - Usage of \_CEE\_ENVFILE\_S in the started task JCL (since version 7.6.0.1).
- JES Job Monitor - The following FEJJCNG directives became optional (since version 7.6.1):
  - HOST\_CODEPAGE
- JES Job Monitor - New optional directives have been added to FEJJCNG (since version 8.0.1):
  - SUBMIT\_TIMEOUT
- JES Job Monitor – The following customizable members have changed (since version 8.0.1):
  - FEJTSO
- PROCLIB - New PROCLIB members have been added (since version 7.6.1):
  - ELAXFDCL
- PROCLIB - The following PROCLIB members have changed (since version 8.0.1):
  - LOCKD
  - RSED
- RSE - Usage of 64-bit Java is now supported (since version 7.6.1).
- RSE - Assume permission is granted when the application ID is not defined in the APPL security class (since version 8.0.1).
- RSE - The RSED and LOCKD started tasks now support TMPDIR (since version 8.0.1).
- RSE - The RSED started task now reads the RSED port number from rsed.envvars (since version 8.0.1).
- RSE - The LOCKD started task now reads the log level from rsecomm.properties (since version 8.0.1).

- RSE - New operator commands have been added (since version 7.6.1 and 8.0.1):
  - MODIFY DISPLAY PROCESS,DETAIL
  - MODIFY IVP DAEMON,userid
- RSE - The following non-customizable directives have changed or are new in `rsed.envvars` (since version 8.0.1):
  - (`_RSE_JAVAOPTS`) -DDSTORE\_KEEPALIVE\_RESPONSE\_TIMEOUT
  - (`_RSE_JAVAOPTS`) -DDSTORE\_IO\_SOCKET\_READ\_TIMEOUT
  - (`_RSE_JAVAOPTS`) -Djob.monitor.port
  - CGI\_ISPCONF
  - CGI\_ISPWORK
  - `_EDC_ADD_ERRNO2`
- RSE - New optional directives have been added to `rsed.envvars` (since version 7.6.0.1 and 8.0.1):
  - (`_RSE_JAVAOPTS`) -Daudit.log.mode
  - (`_RSE_JAVAOPTS`) -Ddeny.nozero.port
  - (`_RSE_JAVAOPTS`) -Dsingle.logon
  - (`_RSE_JAVAOPTS`) -Dprocess.cleanup.interval
  - (`_RSE_JAVAOPTS`) -DDISABLE\_DELETE\_IN\_SUBPROJECT
  - TMPDIR
- RSE - New required directives have been added to `rsed.envvars` (since version 8.0.1):
  - `RSE_RSED_PORT`
  - `RSE_JMON_PORT`
- RSE - The following configuration files are no longer used and should be removed from `/etc/rdz` (since version 8.0.1):
  - `projectcfg.properties`
  - `propertiescfg.properties`
  - `uchars.settings`
- RSE - The following configuration files are new (since version 8.0.1):
  - `pushtoclient.properties`
- RSE - The following directories have been renamed (since version 8.0.1):
  - `/var/rdz/properties` -> `/var/rdz/pushtoclient`
- RSE - The following directories have been moved (since version 8.0.1):
  - `/var/rdz/projects` -> `/var/rdz/pushtoclient/projects`
- RSE - The following console messages have changed or are new (since version 7.6.0.1, 7.6.1 and 8.0.1):
  - FEK001I
  - FEK012I
  - FEK210I
  - FEK900I
  - FEK901I
- RSE - documentation in the FEKSETUP job has exact details on what the job creates, and documents differences between Developer for System z versions (since version 8.0.1).
- RSE - FEKLOGS is enhanced to require less customization and collect more information (since version 8.0.1).

---

## Appendix C. Operator commands

This appendix summarizes the operator (or console) commands information in *Rational Developer for System z Host Configuration Guide* (SC23-7658). Refer to that publication for more details.

---

### Modify (F)

The **MODIFY** command allows you to dynamically query and change the characteristics of an active task. The abbreviated version of the command is the letter F.

### JES Job Monitor

```
>> [MODIFY | F] [procname] [, APPL=-TV | , APPL=-TN] ><
```

Figure 8. MODIFY JMON operator command

#### procname

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

- TV Enable verbose (trace) mode. Tracing will cause performance degradations and should only be done under the direction of the IBM support center.
- TN Disable verbose (trace) mode.

## RSE daemon

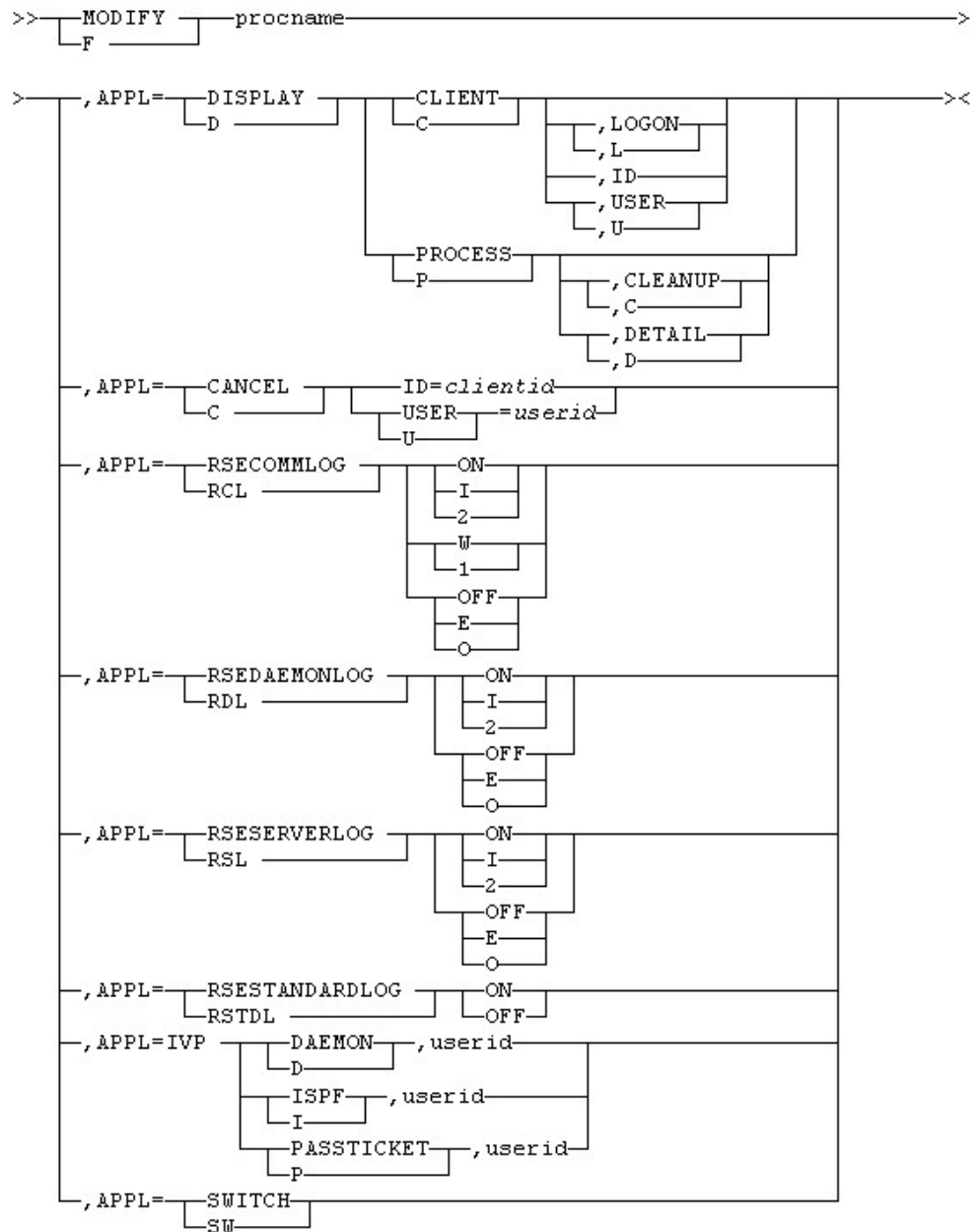


Figure 9. MODIFY RSED operator command

### procname

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

### DISPLAY CLIENT [{LOGON | ,ID | ,USER}]

Display the active clients in a single BPXM023I message. The result layout depends on which command option was used. You can change the sorting order with the optional command arguments.

- No command option: Clients are grouped by the thread pool process that serves them.

```

ProcessId(<processid>) ASId(<asid>) JobName(<jobname>)
Clients(<local>/<total>) Order(<startup order>)
<clientid><userid><connected since>

```

- LOGON command option: Clients are ordered by logon time.

```

LOGON TIME----- ID----- USERID--
<connected since> <clientid> <userid>

```

- ID command option: Clients are ordered by client ID.

```

ID----- USERID-- LOGON TIME-----
<clientid> <userid> <connected since>

```

- USER command option: Clients are ordered by user ID.

```

USERID-- ID----- LOGON TIME-----
<userid> <clientid> <connected since>

```

## DISPLAY PROCESS[,{CLEANUP | ,DETAIL}]

Display the RSE thread pool processes in a single BPXM023I message. There can be multiple processes, which are used for load balancing the connected users.

```

ProcessId(<processid>) Memory Usage(<java heap usage>%)
Clients(<number of clients>) Order(<startup order>) <error status>

```

### Note:

- <processid> can be used in process specific z/OS UNIX operator commands.
- Each process has its own Java heap, whose size can be set in `rsed.envvars`.
- <startup order> is a sequential number that indicates the order that the thread pools were started. The number corresponds to the number used in the filename of the `stderr.*.log` and `stdout.*.log` files.

In normal situations, <error status> is blank. Table 6 documents the possible non-blank values for <error status>.

Table 6. Thread pool error status

| Status           | Description                                                                                                                                                                                                                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| *severe error*   | The thread pool process encountered an unrecoverable error and halted operations. The other status fields show the last known values. Use the CLEANUP option of the <b>DISPLAY PROCESS</b> modify command to remove this entry from the table.                                                              |
| *killed process* | The thread pool process was killed by Java, z/OS UNIX or an operator command. The other status fields show the last known values. Use the CLEANUP option of the <b>DISPLAY PROCESS</b> modify command to remove this entry from the table.                                                                  |
| *timeout*        | The thread pool process did not respond in a timely manner to RSE daemon during a client connect request. The other status fields show the current values. The thread pool is excluded for future client connect requests. The *timeout* status is reset when a client served by this thread pool logs off. |

More information is provided when the DETAIL option of the **DISPLAY PROCESS** modify command is used:

```

ProcessId(33555087) ASId(002E) JobName(RSED8) Order(1)
PROCESS LIMITS: CURRENT HIGHWATER LIMIT
JAVA HEAP USAGE(%) 10 56 100
CLIENTS 0 25 60
MAXFILEPROC 83 103 64000
MAXPROCUSER 97 99 200
MAXTHREADS 9 14 1500
MAXTHREADTASKS 9 14 1500

```

The ASId field is the address space ID, in hexadecimal notation. The process limits table shows the current resource usage, the high-water mark for the resource usage, and the resource limit. Note that due to other limiting factors, the defined limit might never be reached.

#### **CANCEL ID=clientid**

Cancel a client connection based upon the client ID, which is shown in the **DISPLAY CLIENT** modify command.

#### **CANCEL USER=userid**

Cancel a client connection based upon the client's user ID, which is shown in the **DISPLAY CLIENT** modify command.

#### **RSECOMMLOG {ON | OFF | I | W | E | 2 | 1 | 0}**

Control the trace detail level for RSE server (rsecomm.log) and the MVS data set services (lock.log and ffs\*.log). The startup default is defined in rsecomm.properties. There are three detail levels available:

|               |                                                                                |
|---------------|--------------------------------------------------------------------------------|
| E or 0 or OFF | Error messages only.                                                           |
| W or 1        | Error and Warning messages. This is the default setting in rsecomm.properties. |
| I or 2 or ON  | Error, Warning and Informational messages.                                     |

Detailed tracing will cause performance degradations and should only be done under the direction of the IBM support center.

#### **RSEDAEMONLOG {ON | OFF | I | E | 2 | 0}**

Control the trace detail level for RSE daemon (rsedaemon.log). The startup default is defined in rsecomm.properties. There are two detail levels available:

|               |                                             |
|---------------|---------------------------------------------|
| E or 0 or OFF | Error messages only.                        |
| I or 2 or ON  | Error, Warning, and Informational messages. |

Detailed tracing will cause performance degradations and should only be done under the direction of the IBM support center.

#### **RSESERVERLOG {ON | OFF | I | E | 2 | 0}**

Control the trace detail level for RSE thread pools (rseserver.log). The startup default is defined in rsecomm.properties. There are two detail levels available:

|               |                                             |
|---------------|---------------------------------------------|
| E or 0 or OFF | Error messages only.                        |
| I or 2 or ON  | Error, Warning, and Informational messages. |

Detailed tracing will cause performance degradations and should only be done under the direction of the IBM support center.

## RSESTANDARDLOG {ON |, OFF}

Disable (OFF) or enable (ON) updating the log files holding the stdout and stderr streams of the thread pools (stdout.\*.log and stderr.\*.log). The startup default is defined by the enable.standard.log directive in rsed.envvars.

Detailed tracing will cause performance degradations and should only be done under the direction of the IBM support center.

## IVP DAEMON,userid

Log user ID userid on to RSE daemon to do a connection test. Results are shown with one ore more FEK900I console messages. The return code is shown with console message FEK901I.

```
+FEK900I DAEMON IVP: SSL is disabled
+FEK900I DAEMON IVP: connected
+FEK900I DAEMON IVP: 1977
+FEK900I DAEMON IVP: 6902918
+FEK900I DAEMON IVP: Success
+FEK901I DAEMON IVP Exit code = 0
```

### Note:

- The function is similar to what the fekfivpd IVP (Installation Verification Program) does.
- RSE daemon will generate a PassTicket which is used as password for the IVP, so there will be no WTOR (Write To Operator with Reply) requesting a password.

## IVP ISPF,userid

Invoke ISPF's Client Gateway as user ID userid. Results are shown with one ore more FEK900I console messages. The return code is shown with console message FEK901I.

```
+FEK900I ISPF IVP: executed on CDFMVS08 -- Tue Sep 13 22:29:28 EDT 2011
+FEK900I ISPF IVP: executed by uid=1(IBMUSER) gid=0(SYS1)
+FEK900I ISPF IVP: using /etc/rdz/rsed.envvars
+FEK900I ISPF IVP: current address space size limit is 2147483647
(2048.0 MB)
+FEK900I ISPF IVP: maximum address space size limit is 2147483647
(2048.0 MB)
+FEK900I ISPF IVP: -----

+FEK900I ISPF IVP: /etc/rdz/ISPF.conf content:
+FEK900I ISPF IVP: -----

+FEK900I ISPF IVP: ispllib=ISP.SISPLLOAD
+FEK900I ISPF IVP: isplib=ISP.SISPMENU
+FEK900I ISPF IVP: isptlib=ISP.SISPTEU
+FEK900I ISPF IVP: ispplib=ISP.SISPPENU
+FEK900I ISPF IVP: ispslib=ISP.SISPSLIB
+FEK900I ISPF IVP: sysproc=ISP.SISPCLIB,FEK.SFEKPROC
+FEK900I ISPF IVP: -----

+FEK900I ISPF IVP: Host install verification for RSE
+FEK900I ISPF IVP: Review IVP log messages from HOST below :
+FEK900I ISPF IVP: -----

+FEK900I ISPF IVP: Service level 22Feb2011
+FEK900I ISPF IVP: RSE connection and base TS0/ISPF session initializati
on check only
+FEK900I ISPF IVP: *** CHECK : ENVIRONMENT VARIABLES - key variables
displayed below :
+FEK900I ISPF IVP: Server PATH = ./usr/lpp/java/J5.0/bin:/usr/l
pp/rdz/bin:/usr/lpp/ispf/bin:/bin:/usr/sbin
+FEK900I ISPF IVP: STEPLIB = NONE
```

```

+FEK900I ISPF IVP: Temporary directory = /tmp
+FEK900I ISPF IVP: _CMDSERV_BASE_HOME = /usr/lpp/ispf
+FEK900I ISPF IVP: _CMDSERV_CONF_HOME = /etc/rdz
+FEK900I ISPF IVP: _CMDSERV_WORK_HOME = /var/rdz
+FEK900I ISPF IVP: -----
+FEK900I ISPF IVP: *** CHECK : USS MODULES
+FEK900I ISPF IVP: Checking ISPF Directory : /usr/lpp/ispf
+FEK900I ISPF IVP: Checking modules in /usr/lpp/ispf/bin directory
+FEK900I ISPF IVP: Checking for ISPF configuration file ISPF.conf
+FEK900I ISPF IVP: RC=0
+FEK900I ISPF IVP: MSG: SUCCESSFUL
+FEK900I ISPF IVP: -----
+FEK900I ISPF IVP: *** CHECK : TSO/ISPF INITIALIZATION
+FEK900I ISPF IVP: (TSO/ISPF session will be initialized)
+FEK900I ISPF IVP: RC=0
+FEK900I ISPF IVP: MSG: SUCCESSFUL
+FEK900I ISPF IVP: -----
+FEK900I ISPF IVP: *** CHECK: Shutting down TSO/ISPF IVP session
+FEK900I ISPF IVP: RC=0
+FEK900I ISPF IVP: MSG: SUCCESSFUL
+FEK900I ISPF IVP: -----
+FEK900I ISPF IVP: Host installation verification completed successfully
+FEK900I ISPF IVP: -----
+FEK901I ISPF IVP Exit code = 0

```

**Note:**

- The function is similar to what the fekfivpi IVP (Installation Verification Program) does.
- RSE daemon will generate a PassTicket which is used as password for the IVP, so there will be no WTOR (Write To Operator with Reply) requesting a password.

**IVP PASSTICKET,userid**

Test the reusability of a PassTicket generated for user ID userid. Results are shown with one or more FEK900I console messages. The return code is shown with console message FEK901I.

```

+FEK900I PASSTICKET IVP: the default applid=FEKAPPL
+FEK900I PASSTICKET IVP: Success, PassTicket IVP finished normally
+FEK901I PASSTICKET IVP Exit code = 0

```

**Note:**

- When using RACF as security product, reusable PassTickets require the "NO REPLAY PROTECTION" keyword in the security definitions.
- There is no equivalent IVP (Installation Verification Program) for this test. Starting RSE daemon with the IVP=IVP argument will invoke a PassTicket IVP that tests PassTicket generation, but it cannot test PassTicket reusability.
- RSE daemon will generate a PassTicket which is used as password for the IVP, so there will be no WTOR (Write To Operator with Reply) requesting a password.

**SWITCH**

Switch to a new audit log file.



## Lock daemon

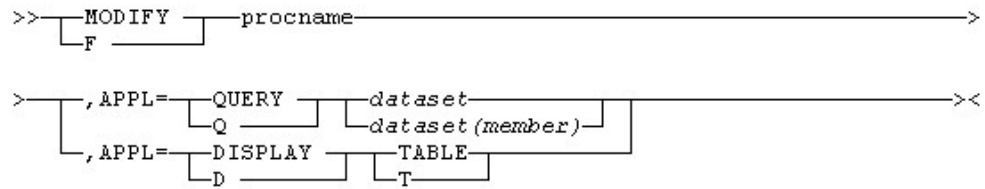


Figure 10. MODIFY LOCKD operator command

### procname

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

### QUERY dataset[(member)]

Query the lock status of the listed data set or member. The server will reply with one of the following messages:

```

BPXM023I (stclock) dataset[(member)] NOT LOCKED
BPXM023I (stclock) dataset[(member)] LOCKED BY userid

```

#### Note:

- The server will also report locks held by other products, such as ISPF.
- Locks held by Developer for System z clients who were unable to register with the lock daemon will result in the thread pool server address space (RSEDx) being reported as lock owner.

Console message FEK513W is generated when RSE server is unable to register the client with the lock daemon. The ASID and TCB values mentioned in this message can be compared against the output of the **D GRS,RES=(\*,dataset[(member)])** operator command in order to find the actual user holding the lock.

### DISPLAY TABLE

Display the lock daemon mapping table in a single BPXM023I message. The lock daemon uses this mapping table to determine which Developer for System z user holds a certain data set lock (GRS reports only the ASID/TCB pair).

```

PID----- ASID TCB----- USERID--
 350 001A 00123ABC IBMUSER

```



---

## Appendix D. Optional customization

This section summarizes the CARMA, Application Deployment Manager, SCLM Developer Toolkit, and other customization task information in *Rational Developer for System z Host Configuration Guide* (SC23-7658). Refer to that publication for more details.

---

### (Optional) Common Access Repository Manager (CARMA)

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

- TCP/IP port range for internal communication
- Security rule to allow developers update to CARMA VSAM files
- (Optional) Security rule to allow users to submit CRA\* jobs
- (Optional) LPA update

Common Access Repository Manager (CARMA) is a server platform for Repository Access Managers (RAMs). A RAM is an Application Programming Interface (API) for a z/OS based Software Configuration Manager (SCM). By wrapping the SCM functionality in a RAM, a single API is available for a client to access any supported SCM.

Developer for System z provides multiple pre-built RAMs, as well as source code examples for creating your own RAM.

The IBM® Rational® Developer for System z Interface for CA Endeavor® Software Configuration Manager gives Developer for System z clients direct access to CA Endeavor® SCM.

---

### (Optional) SCLM Developer Toolkit

You will need assistance of an SCLM administrator and optionally a security administrator to complete this customization task, which requires the following resources and/or special customization tasks:

- APF and LINKLIST updates
- Define SCLM language translators for JAVA/J2EE support
- Define SCLM types for JAVA/J2EE support
- (Optional) Security rule to allow users update to an SCLM VSAM
- (Optional) Installation of Ant

SCLM Developer Toolkit provides the tools needed to extend the capabilities of SCLM to the client. SCLM (Software Configuration and Library Manager) itself is a host-based source code manager that is shipped as part of ISPF.

The SCLM Developer Toolkit has an Eclipse-based plug-in that interfaces to SCLM and provides for access to all SCLM processes for legacy code development as well as support for full Java and J2EE development on the workstation with

synchronization to SCLM on the mainframe including building, assembling, and deployment of the J2EE code from the mainframe.

---

## **(Optional) Application Deployment Manager**

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

- TCP/IP port for external communication
- Update CICS region JCL
- Update CICS region CSD
- Define group to CICS region
- Security rule to allow administrators update to an Application Deployment Manager VSAM
- CICS TS security setup
- (Optional) Define CICS transaction names
- (Optional) Security rule to allow users update to an Application Deployment Manager VSAM

Developer for System z uses certain functions of Application Deployment Manager as a common deployment approach for various components. Optional customization enables more features of Application Deployment Manager and can add the following services to Developer for System z:

- IBM CICS Explorer provides an Eclipse-based infrastructure to view and manage CICS resources and enables greater integration between CICS tools.
- CICS Resource Definition (CRD) client and server provide the following functions:
  - CICS Resource Definition editor
  - Allow application developers to define CICS resources in a limited, controlled, and secure fashion.
  - Prevent CICS development access to unauthorized or incorrect VSAM data sets by providing the CICS administrator control over the physical data set name attribute in File definitions.
  - Miscellaneous CICS development aids
  - Miscellaneous CICS Web Service development aids

---

## **(Optional) pushtoclient.properties, Host-based client control**

This customization task does not require assistance, special resources, or special customization tasks.

Developer for System z clients version 8.0.1 and higher will pull client configuration files and upgrade information from the host when they connect, ensuring that all clients have common settings and that they are up-to-date.

z/OS Projects can be defined individually through the z/OS Projects perspective on the client or can be defined centrally on the host and propagated to the client on a per user basis. These "host-based projects" look and function exactly like projects defined on the client except that their structure, members, and properties cannot be modified by the client, and they are only accessible when connected to the host.

---

## **(Optional) (deprecated) FMEXT.properties, File Manager integration**

You will need the assistance of a security administrator to complete this customization task, which requires the following resources or special customization tasks:

- Security rule to add program controlled data sets

Developer for System z supports direct access from the client to a limited set of IBM File Manager for z/OS functions. IBM File Manager for z/OS provides comprehensive tools for working with MVS data sets, z/OS UNIX files, DB2, IMS and CICS data.

Note that the IBM File Manager for z/OS product must be ordered, installed and configured separately. Refer to *Rational Developer for System z Prerequisites Guide* (SC23-7659) to know which level of File Manager is required for your version of Developer for System z. The installation and customization of this product is not described in this manual.

---

## **(Optional) ssl.properties, RSE SSL encryption**

You will need assistance of a security administrator to complete this customization task, which requires the following resources or special customization tasks:

- LINKLIST update
- Security rule to add program controlled data sets
- (Optional) Security rule to add certificate for SSL

External (client-host) communication can be encrypted using SSL. This feature is disabled by default and is controlled by the settings in `ssl.properties`.

---

## **(Optional) rsecomm.properties, RSE tracing**

This customization task does not require assistance, special resources, or special customization tasks.

Developer for System z supports different levels of tracing the internal program flow for problem solving purposes. RSE, and some of the services called by RSE, use the settings in `rsecomm.properties` to know the desired detail level in the output logs.

---

## **(Optional) DB2 stored procedure**

You will need the assistance of a WLM administrator and a DB2 administrator to complete this customization task, which requires the following resources or special customization tasks:

- WLM update
- New PROCLIB member
- DB2 update

Developer for System z provides a sample DB2 stored procedure (PL/I and COBOL Stored Procedure Builder) for building COBOL and PL/I Stored Procedures from within the Developer for System z client.

---

## (Optional) CICS bidirectional language support

You will need the assistance of a CICS administrator to complete this customization task, which requires the following resources or special customization tasks:

- Update CICS region JCL
- Define a program to CICS

The Developer for System z Enterprise Service Tools (EST) component supports different formats of Arabic and Hebrew interface messages, as well as bidirectional data presentation and editing in all editors and views. In terminal applications, both left-to-right and right-to-left screens are supported, as well as numeric fields and fields with opposite-to-screen orientation.

Additional bidirectional features and functionality include the following:

- The EST service requestor dynamically specifies bidirectional attributes of interface messages.
- Bidirectional data processing in service flows is based on bidirectional attributes (text type, text orientation, numeric swapping, and symmetric swapping). These attributes can be specified in different stages of flow creation for both interface and terminal flows.
- EST-generated runtime code includes conversion of data between fields in messages that have different bidirectional attributes.

Additionally, EST-generated code can support bidi transformation in environments other than CICS SFR (for example, batch applications). You can make the EST generators to include calls to the bidirectional conversion routines by specifying the appropriate bidi transformation options in the EST generation wizards and linking the generated programs with the appropriate bidirectional conversion library, FEK.SFEKLOAD.

---

## (Optional) Diagnostic IRZ error messages

This customization task does not require assistance, but does require the following resources or special customization tasks:

- LINKLIST update
- Update CICS region JCL

The Developer for System z client has a code generation component called Enterprise Service Tools (EST). In order for code generated by EST to issue diagnostic error messages, all IRZ\* and IIRZ\* modules in the FEK.SFEKLOAD load library must be made available to the generated code.

---

## (Optional) WORKAREA and /tmp cleanup

This customization task does not require assistance, special resources, or special customization tasks.

ISPF's TSO/ISPF Client Gateway and the SCLM Developer Toolkit function use the WORKAREA and /tmp directories to store temporary work files, which are removed before the session is closed. However, temporary output is sometimes left behind,

for example, if there is a communication error while processing. For this reason, it is recommended that you clear out the WORKAREA and /tmp directories from time to time.

---

## **(Optional) Using REXEC (or SSH)**

This customization task does not require assistance, special resources, or special customization tasks.

REXEC (Remote Execution) is a TCP/IP service to let clients execute a command on the host. SSH (Secure Shell) is a similar service, but here all communication is encrypted using SSL (Secure Socket Layer). Developer for System z uses either service for doing remote (host-based) actions in z/OS UNIX subprojects.





---

## Appendix E. Host Configuration Reference

This section summarizes the information in *Rational Developer for System z Host Configuration Reference* (SC14-7290). Refer to that publication for more details.

---

### Understanding Developer for System z

The Developer for System z host consists of several components that interact to give the client access to the host services and data. Understanding the design of these components can help you make the correct configuration decisions.

---

### Security considerations

Developer for System z provides mainframe access to users on a non-mainframe workstation. Validating connection requests, providing secure communication between the host and the workstation, and authorizing and auditing activity are therefore important aspects of the product configuration.

---

### TCP/IP considerations

Developer for System z uses TCP/IP to provide mainframe access to users on a non-mainframe workstation. It also uses TCP/IP for communication between various components and other products.

---

### WLM considerations

Unlike traditional z/OS applications, Developer for System z is not a monolithic application that can be identified easily to Workload Manager (WLM). Developer for System z consists of several components that interact to give the client access to the host services and data. Some of these services are active in different address spaces, resulting in different WLM classifications.

---

### Tuning considerations

RSE (Remote Systems Explorer) is the core of Developer for System z. To manage the connections and workloads from the clients, RSE is composed of a daemon address space, which controls thread pooling address spaces. The daemon acts as a focal point for connection and management purposes, while the thread pools process the client workloads.

This makes RSE a prime target for tuning the Developer for System z setup. However, maintaining hundreds of users, each using 16 or more threads, a certain amount of storage, and possibly one or more address spaces requires proper configuration of both Developer for System z and z/OS.

---

### Performance considerations

z/OS is a highly customizable operating system, and (sometimes small) system changes can have a huge impact on the overall performance. This chapter highlights some of the changes that can be made to improve the performance of Developer for System z.

---

## **Push-to-client considerations**

Push-to-client, or host-based client control, supports central management of the following:

- Client configuration files
- Client product version
- Project definitions

---

## **CICSTS considerations**

This chapter contains information useful for a CICS Transaction Server administrator.

---

## **Customizing the TSO environment**

This chapter assists you with mimicking a TSO logon procedure by adding DD statements and data sets to the TSO environment in Developer for System z.

---

## **Running multiple instances**

There are times that you want multiple instances of Developer for System z active on the same system, for example, when testing an upgrade. However, some resources such as TCP/IP ports cannot be shared, so the defaults are not always applicable. Use the information in this chapter to plan the coexistence of the different instances of Developer for System z, after which you can use this configuration guide to customize them.

---

## **Troubleshooting configuration problems**

This chapter is provided to assist you with some common problems that you may encounter during your configuration of Developer for System z, and has the following sections:

- Log and setup analysis using FEKLOGS
- Log files
- Dump files
- Tracing
- z/OS UNIX permission bits
- Reserved TCP/IP ports
- Address Space size
- APPC transaction and TSO Commands service
- Miscellaneous information

---

## **Setting up SSL and X.509 authentication**

This appendix is provided to assist you with some common problems that you may encounter when setting up Secure Socket Layer (SSL), or during checking or modifying an existing setup. This appendix also provides a sample setup to support users authenticating themselves with an X.509 certificate.

---

## Setting up TCP/IP

This appendix is provided to assist you with some common problems that you may encounter when setting up TCP/IP, or during checking or modifying an existing setup.



---

## Documentation notices for IBM Rational Developer for System Z

© Copyright IBM Corporation - 2011

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

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

For license inquiries regarding double-byte (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.  
3-2-12, Roppongi, Minato-ku, Tokyo 106-8711 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 Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites 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

3039 Cornwallis Road, PO Box 12195  
Research Triangle Park, NC 27709  
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.

---

## Trademark acknowledgments

IBM, the IBM logo, and [ibm.com](http://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](http://www.ibm.com/legal/copytrade.shtml).

CA Endeavor is a registered trademark of CA Technologies.

Rational are trademarks of International Business Machines Corporation and Rational Software Corporation, in the United States, other countries, or both.

Intel and Pentium are trademarks of Intel Corporation in the United States, or other countries, or both.

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.





---

# Index

## Special characters

/tmp cleanup 42

## A

Activating IBM Common Access  
Repository Manager 39  
ADM, customization 40  
APF authorizations in PROGxx 6  
Application Deployment Manager,  
customization 40

## B

BPXPRMxx, Set z/OS UNIX limits in 6  
build procedures, ELAXF\* remote 9

## C

CARMA, activating 39  
Client Gateway configuration file,  
TSO/ISPF 15  
Client user ID 2  
command security, Define JES 22  
commands, JES Job Monitor Modify 31  
commands, Lock daemon Modify 37  
commands, RSE daemon Modify 32  
COMMNDxx, Add started tasks to 6  
Common Access Repository Manager,  
Activating 39  
configuration file, FEJJC�FG 11  
configuration, ssl.properties 41  
considerations, Security 19  
controlled libraries for RSE server, Define  
MVS 23  
customization, SCLM Developer  
Toolkit 39

## D

data set profiles, Define 21  
definitions, Security 10  
Deployment and upgrade 3  
Developer for System z started tasks,  
Define 21

## E

ELAXF\* procedures, Sample 9  
ELAXF\* remote build procedures 9

## F

FEJJC�FG, JES Job Monitor configuration  
file 11  
File Manager integration 41  
FMIEXT.properties 41

## H

Host based client control 40

## I

IBM Common Access Repository  
Manager, Activating 39  
ISPF Client Gateway 17  
ISPF.conf 15  
IVP operator commands 17

## J

JES command security, Define 22  
JES Job Monitor configuration file,  
FEJJC�FG 11  
JES Job Monitor server 7  
JES Job Monitor, Modify command 31

## L

libraries for RSE server, Define MVS 23  
LINKLIST and LPA definitions,  
Prerequisite 7  
Lock daemon, Modify command 37  
Logging configuration file,  
rsecomm.properties 41  
LPA definitions, Prerequisite 7

## M

migration 8.0.3 27  
MVS program controlled libraries for RSE  
server, Define 23

## O

OMVS segment, Define 21  
operator commands, IVP 17

## P

PARMLIB, changes 5  
PassTicket reusability 17  
PassTicket support for RSE server,  
Define 24  
Preparation 1  
Prerequisite LINKLIST and LPA  
definitions 7  
Prerequisite software 1  
PROCLIB changes 7  
profiles, Define data set 21  
PROGxx, APF authorizations in 6  
pushtoclient.properties 40

## R

remote build procedures, ELAXF\* 9  
Requisite software 1  
REXEC, using 43  
RSE daemon connection 17  
RSE daemon, Modify command 32  
RSE server 8  
RSE server, Define as a secure z/OS  
UNIX server 23  
RSE server, Define MVS program  
controlled libraries for 23  
RSE server, Define PassTicket support  
for 24  
RSE server, Define z/OS UNIX program  
controlled files for 24  
RSE server, Prerequisite LINKLIST and  
LPA definitions 7  
RSE SSL configuration, ssl.properties, 41  
RSE trace configuration,  
rsecomm.properties, 41  
rsecomm.properties, 41  
rsed.envvars 33

## S

SCLM Developer Toolkit,  
customization 39  
secure z/OS UNIX server, Define RSE  
server as a 23  
Security considerations 19  
Security definitions 10  
security settings and classes, Activate 19  
security settings, verify 25  
security, Define JES command 22  
segment, Define OMVS 21  
server, Define MVS program controlled  
libraries for RSE 23  
server, Define PassTicket support for  
RSE 24  
server, Define RSE server as a secure  
z/OS UNIX 23  
server, Define z/OS UNIX program  
controlled files for RSE 24  
server, JES Job Monitor 7  
server, RSE 8  
Servers 2  
settings and classes, Activate security 19  
SSH, using 43  
ssl.properties 41  
started tasks, Define Developer for  
System z 21  
support for RSE server, Define  
PassTicket 24

## T

tasks to COMMNDxx, Add 6  
tasks, Define Developer for System z 21  
trace configuration,  
rsecomm.properties 41

TSO/ISPF Client Gateway configuration  
file 15

## U

UNIX limits in BPXPRMxx, Set z/OS 6  
UNIX program controlled files for RSE  
server, Define 24  
UNIX server, Define RSE server as 23  
upgrade, Deployment and 3  
user ID, Client 2

## V

Verify security settings 25

## W

WLM considerations 45  
WORKAREA cleanup 42

## Z

z/OS UNIX limits in BPXPRMxx 6  
z/OS UNIX program controlled files for  
RSE server, Define 24  
z/OS UNIX server, Define RSE server  
as 23

---

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

IBM Rational Developer for System z  
Version 8.0.3  
Host Configuration Quick Start Guide

Publication No. GI11-9201-04

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.

Send your comments to the address on the reverse side of this form.

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

\_\_\_\_\_  
Name

\_\_\_\_\_  
Address

\_\_\_\_\_  
Company or Organization

\_\_\_\_\_  
Phone No.

\_\_\_\_\_  
Email address



Cut or Fold  
Along Line

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  
Corporation  
Building 501  
P.O Box 12195  
Research Triangle Park, NC  
USA 27709-2195



Fold and Tape

Please do not staple

Fold and Tape

Cut or Fold  
Along Line





Printed in USA

GI11-9201-04

