

IBM Rational Developer for System z  
Version 9.0.1

*Configuration Guide*





IBM Rational Developer for System z  
Version 9.0.1

*Configuration Guide*



**Note**

Before using this information, be sure to read the general information under “Documentation notices for IBM Rational Developer for System z” on page 201.

**Eleventh edition (December 2013)**

This edition applies to IBM Rational Developer for System z Version 9.0.1 (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, 2013.

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

# Contents

<b>Figures</b> . . . . .	<b>vii</b>
--------------------------	------------

<b>Tables</b> . . . . .	<b>ix</b>
-------------------------	-----------

## About this document . . . . . xi

Who should use this document . . . . .	xi
Summary of changes . . . . .	xii
Description of the document content. . . . .	xiv
Planning . . . . .	xiv
Basic customization . . . . .	xiv
(Optional) Common Access Repository Manager (CARMA) . . . . .	xv
(Optional) SCLM Developer Toolkit . . . . .	xv
(Optional) Application Deployment Manager (deprecated) . . . . .	xv
(Optional) Host-based code analysis . . . . .	xv
(Optional) Other customization tasks . . . . .	xvi
Installation verification . . . . .	xvi
Security definitions . . . . .	xvi
Migration guide . . . . .	xvi
Operator commands . . . . .	xvi
Host configuration reference . . . . .	xvii

## IBM Rational Developer for System z Host Configuration Guide . . . . . 1

### Chapter 1. Planning. . . . . 3

Migration considerations . . . . .	3
Planning considerations . . . . .	3
Product overview. . . . .	3
Skill requirements . . . . .	4
Time requirements . . . . .	4
Preinstallation considerations. . . . .	4
Installation user ID . . . . .	4
Requisite products . . . . .	5
Required resources . . . . .	5
Pre-configuration considerations. . . . .	9
Workload management . . . . .	9
Resource usage and system limits . . . . .	9
Required configuration of requisite products. . . . .	9
User ID considerations . . . . .	9
Server considerations . . . . .	10
Configuration method . . . . .	11
Predeployment considerations . . . . .	11
Client checklist . . . . .	12

### Chapter 2. Basic customization . . . . . 15

Requirements and checklist . . . . .	15
Customization setup . . . . .	15
PARMLIB changes . . . . .	16
Set the z/OS UNIX limits in BPXPRMxx. . . . .	17
Add the started tasks to COMMNDxx . . . . .	18
SVC definitions in IEASVCxx . . . . .	18
LPA definitions in LPALSTxx . . . . .	19

APF authorizations in PROGxx. . . . .	19
LINKLIST definitions in PROGxx . . . . .	20
Requisite LINKLIST and LPA definitions . . . . .	22
LINKLIST definitions for other products. . . . .	23
PROCLIB changes . . . . .	23
JMON, JES Job Monitor started task . . . . .	23
DBGMGR, Debug manager started task . . . . .	24
RSED, RSE daemon started task . . . . .	25
JCL limitations for the PARM variable . . . . .	26
TMPDIR processing . . . . .	27
ELAXF* remote build procedures . . . . .	27
Security definitions . . . . .	29
FEJCNFG, the JES Job Monitor configuration file. . . . .	30
rсед.envvars, the RSE configuration file . . . . .	35
Defining the PORTRANGE available for RSE server . . . . .	44
Defining extra Java startup parameters with _RSE_JAVAOPTS . . . . .	45
Defining the extra Java startup parameters with _RSE_ISPF_OPTS . . . . .	52
ISPF.conf, the ISPF's TSO/ISPF Client Gateway configuration file . . . . .	52
Optional components . . . . .	54
Installation verification . . . . .	54

### Chapter 3. (Optional) Common Access Repository Manager (CARMA). . . . . 57

Requirements and checklist . . . . .	57
Select the server startup method and active RAM . . . . .	58
CARMA server startup . . . . .	58
CRASTART . . . . .	58
Batch submit . . . . .	58
(Deprecated) TSO/ISPF Client Gateway . . . . .	58
Production RAMs . . . . .	59
CA Endeavor SCM RAM . . . . .	59
CA Endeavor SCM packages RAM . . . . .	59
Sample RAMs . . . . .	59
PDS RAM . . . . .	59
Skeleton RAM . . . . .	59
SCLM RAM . . . . .	59
Preconfigured RAM and server startup combinations . . . . .	59
CRASTART with CA Endeavor SCM RAM . . . . .	59
Create the CARMA VSAM data sets . . . . .	60
Customize CRASRV.properties . . . . .	60
Customize crastart.endevor.conf . . . . .	60
(Optional) Additional CA Endeavor SCM RAM customization . . . . .	61
CRASTART with sample RAMs . . . . .	62
Create the CARMA VSAM data sets . . . . .	62
CARMA . . . . .	62
Sample RAMs . . . . .	62
Customize CRASRV.properties . . . . .	62
Customize crastart.conf . . . . .	63
(Optional) Additional custom RAM customization . . . . .	63

Batch submit with CA Endeavor SCM RAM . . . . .	64
Create the CARMA VSAM data sets . . . . .	64
Customize CRASRV.properties . . . . .	64
Customize CRASUBCA . . . . .	64
(Optional) Additional CA Endeavor SCM RAM customization . . . . .	65
Batch submit with sample RAMs . . . . .	66
Create the VSAM data sets . . . . .	66
CARMA . . . . .	66
Sample RAMs . . . . .	66
Customize CRASRV.properties . . . . .	66
Customize CRASUBMT . . . . .	67
(Optional) Additional custom RAM customization . . . . .	67
CARMA configuration details . . . . .	68
CRASRV.properties, the RSE interface to CARMA	68
crastart*.conf, the CRASTART server startup	71
Collecting the CRASTART log files . . . . .	73
CRASUB*, the batch submit server startup . . . . .	74
CARMA VSAM data sets . . . . .	75
CRADEF, the configuration data set . . . . .	76
CRAMSG, the message data set . . . . .	76
CRASTRS, the custom string data set . . . . .	76
CARMA VSAM migration notes . . . . .	76
CARMA Repository Access Managers (RAMs) . . . . .	77
CA Endeavor SCM RAM . . . . .	77
CA Endeavor SCM packages RAM . . . . .	78
PDS RAM . . . . .	78
Skeleton RAM . . . . .	78
SCLM RAM . . . . .	78
CRACFG, CRASHOW and CRATMAP, the CA Endeavor SCM RAM configuration files . . . . .	79
CRACFG, CA Endeavor SCM RAM interaction with the SCM . . . . .	79
CRASHOW, CA Endeavor SCM RAM default filters . . . . .	79
CRATMAP, the CA Endeavor SCM RAM file extension mappings . . . . .	80
CRANDVRA, the CA Endeavor SCM RAM allocation exec . . . . .	80
CA Endeavor SCM RAM batch actions . . . . .	81
CRABCFG, the CA Endeavor SCM RAM batch-action configuration . . . . .	81
CRABATCA, the CA Endeavor SCM RAM batch action JCL . . . . .	82
CRABJOB, the CA Endeavor SCM RAM batch action JOB card . . . . .	83
CRAALLOC, the custom RAM allocation exec . . . . .	83
CARMA return codes . . . . .	84
(Optional) Supporting multiple RAMs . . . . .	84
Example . . . . .	85
(Optional) Custom allocation exec . . . . .	86
(Optional) CARMA user exit . . . . .	86
(Optional) IRXJCL versus CRAXJCL . . . . .	87
Create CRAXJCL . . . . .	88

**Chapter 4. (Optional) SCLM Developer Toolkit . . . . . 89**

Requirements and checklist . . . . .	89
Prerequisites . . . . .	89
ISPF.conf updates for SCLMDT . . . . .	90

rased.envvars updates for SCLMDT . . . . .	91
(Optional) Long/short name translation . . . . .	91
Create LSTRANS.FILE, the long/short name translation VSAM . . . . .	92
rased.envvars updates for long/short name translation . . . . .	94
(Optional) Install and customize Ant . . . . .	94
SCLM updates for SCLMDT . . . . .	95
Remove old files from WORKAREA and /tmp . . . . .	96

**Chapter 5. (Optional) Application Deployment Manager (deprecated). . . 97**

Requirements and checklist . . . . .	97
CRD repository . . . . .	98
CICS administrative utility . . . . .	98
RESTful versus Web Service . . . . .	99
CRD server that uses the RESTful interface . . . . .	99
CICS primary connection region . . . . .	99
CICS non-primary connection regions (Optional) Customize the CRD server transaction IDs . . . . .	100
CRD server that uses the Web Service interface . . . . .	100
Pipeline message handler . . . . .	101
CICS primary connection region . . . . .	101
CICS non-primary connection regions (Optional) Manifest repository . . . . .	102

**Chapter 6. (Optional) Host-based code analysis. . . . . 105**

Requirements and checklist . . . . .	105
Code review . . . . .	105
Modify code review processing . . . . .	106
Code coverage . . . . .	106
Code coverage output . . . . .	106

**Chapter 7. (Optional) Other customization tasks . . . . . 107**

(Optional) pushtoclient.properties, the host-based client control . . . . .	107
(Optional) ssl.properties, the RSE SSL encryption . . . . .	110
(Optional) rsecomm.properties, the RSE tracing . . . . .	112
(Optional) include.conf, Forced includes for C/C++ content assist . . . . .	113
(Optional) z/OS UNIX subprojects . . . . .	115
REXEC or SSH setup . . . . .	115
(Optional) Include preprocessor support . . . . .	115
(Optional) xUnit support for Enterprise COBOL and PL/I . . . . .	116
(Optional) Enterprise Service Tools support . . . . .	117
(Optional) CICS bidirectional language support . . . . .	118
(Optional) Diagnostic IRZ messages for the generated code . . . . .	118
(Optional) Integrated debugger . . . . .	120
Integrated debugger configuration parameters . . . . .	121
Integrated debugger parmlib updates . . . . .	121
Integrated debugger TCP/IP updates . . . . .	121
Integrated debugger security updates . . . . .	122
Integrated debugger CICS updates . . . . .	122
(Optional) Problem Determination Tools support . . . . .	122
(Optional) DB2 and IMS debug support . . . . .	123

(Optional) File Manager support . . . . .	123	Configuration files. . . . .	168
(Optional) WORKAREA and /tmp cleanup . . . . .	124	Version 8.5 migration notes. . . . .	169
<b>Chapter 8. Installation verification . . . . .</b>	<b>127</b>	Migrate from version 8.0.1 to version 8.5 . . . . .	170
Verify the started tasks . . . . .	127	IBM Rational Developer for System z, FMID	
JMON, the JES Job Monitor. . . . .	127	HHOP850 . . . . .	170
RSED, the RSE daemon . . . . .	127	Configurable files . . . . .	173
DBGMGR, the debug manager . . . . .	131	<b>Chapter 11. Operator commands . . . . .</b>	<b>177</b>
IVP operator commands. . . . .	132	Start (S) . . . . .	177
PassTicket reusability. . . . .	132	Integrated debugger . . . . .	177
RSE daemon connection. . . . .	132	JES Job Monitor . . . . .	178
ISPF Client Gateway . . . . .	132	RSE daemon. . . . .	178
Verify the services. . . . .	133	Modify (F) . . . . .	179
IVP initialization . . . . .	134	Integrated debugger . . . . .	179
Port availability . . . . .	135	JES Job Monitor . . . . .	180
TCP/IP setup . . . . .	135	RSE daemon. . . . .	182
RSE daemon connection. . . . .	136	Stop (P) . . . . .	190
JES Job Monitor connection. . . . .	137	How to read a syntax diagram . . . . .	190
ISPF's TSO/ISPF Client Gateway connection . . . . .	137	Symbols . . . . .	190
(Optional) CARMA connection . . . . .	139	Operands. . . . .	190
(Optional) SCLMDT connection . . . . .	140	Syntax example . . . . .	191
(Optional) Integrated Debugger connection . . . . .	141	Nonalphanumeric characters and blank spaces . . . . .	191
		Selecting more than one operand. . . . .	191
		Longer than one line . . . . .	191
		Syntax fragments . . . . .	191
<b>Chapter 9. Security definitions . . . . .</b>	<b>143</b>	<b>Chapter 12. Host Configuration</b>	
Requirements and checklist. . . . .	143	<b>Reference . . . . .</b>	<b>193</b>
Activate the security settings and classes . . . . .	144	Understanding Developer for System z. . . . .	193
Define an OMVS segment for Developer for		Security considerations . . . . .	193
System z users . . . . .	145	TCP/IP considerations . . . . .	193
Define the Developer for System z started tasks	146	WLM considerations . . . . .	193
Define RSE as a secure z/OS UNIX server. . . . .	147	Tuning considerations . . . . .	193
Define the MVS program controlled libraries for		Performance considerations. . . . .	193
RSE . . . . .	147	Push-to-client considerations . . . . .	194
Define the PassTicket support for RSE . . . . .	148	CICSTS considerations . . . . .	194
Define the application protection for RSE . . . . .	149	User exit considerations . . . . .	194
Define the JES command security . . . . .	150	Customizing the TSO environment . . . . .	194
Define the data set profiles . . . . .	151	Running multiple instances. . . . .	194
Define the z/OS UNIX program controlled files for		Troubleshooting configuration problems . . . . .	194
RSE . . . . .	155	Setting up SSL and X.509 authentication . . . . .	195
Verify the security settings . . . . .	156	Setting up TCP/IP . . . . .	195
<b>Chapter 10. Migration guide . . . . .</b>	<b>159</b>	<b>Bibliography. . . . .</b>	<b>197</b>
Migration considerations . . . . .	159	Referenced publications . . . . .	197
Backing up the previously configured files . . . . .	159	Informational publications . . . . .	200
Version 9.0 migration notes. . . . .	160	<b>Documentation notices for IBM</b>	
IBM Rational Developer for System z, FMID		<b>Rational Developer for System z . . . . .</b>	<b>201</b>
HHOP900 . . . . .	160	Copyright license . . . . .	204
IBM Rational Developer for System z Host		Trademark acknowledgments . . . . .	204
Utilities, FMID HAKG900 . . . . .	161		
Migrate from version 8.5 to version 9.0. . . . .	161		
IBM Rational Developer for System z, FMID			
HHOP900 . . . . .	161		
Configurable files . . . . .	163		
IBM Rational Developer for System z Host			
Utilities, FMID HAKG900 . . . . .	168		



# Figures

1.	JMON: JES Job Monitor started task . . . . .	24	23.	CRASUB*: CARMA startup using batch submit	75
2.	DBGMGR: Debug Manager started task . . . . .	25	24.	CRACFG - CA Endeavor SCM RAM interaction with the SCM. . . . .	79
3.	RSED: RSE daemon started task. . . . .	25	25.	CRASHOW - CA Endeavor SCM RAM default filters . . . . .	79
4.	RSED: Alternate RSE daemon startup . . . . .	26	26.	CRATMAP: CA Endeavor SCM RAM default filters . . . . .	80
5.	rсед.stdin.sh: Alternate RSE daemon startup	26	27.	CRABCFG: CA Endeavor SCM RAM batch-action configuration. . . . .	82
6.	RSED: alternate TMPDIR processing . . . . .	27	28.	CRABATCA: CA Endeavor SCM RAM batch-action JCL . . . . .	83
7.	rсед.stdenv: alternate TMPDIR processing	27	29.	CRABJOB: CA Endeavor SCM RAM batch-action JOB card . . . . .	83
8.	FEJCNFG, JES Job Monitor configuration file	31	30.	ISPF.conf updates for SCLMDT . . . . .	91
9.	rсед.envvars: RSE configuration file . . . . .	36	31.	rсед.envvars updates for SCLMDT. . . . .	91
10.	rсед.envvars: RSE configuration file (continued). . . . .	37	32.	FLM02LST: Long/short name translation setup JCL . . . . .	93
11.	rсед.envvars: RSE configuration file (continued). . . . .	38	33.	pushtoclient.properties: Host-based client control configuration file. . . . .	108
12.	ISPF.conf: ISPF configuration file . . . . .	53	34.	ssl.properties – SSL configuration file	111
13.	CRASRV.properties: CRASTART with CA Endeavor SCM RAM. . . . .	60	35.	rsecomm.properties – Logging configuration file . . . . .	113
14.	crastart.endeavor.conf: CRASTART with CA Endeavor SCM RAM. . . . .	61	36.	include.conf - Forced includes for C/C++ content assist. . . . .	114
15.	CRASRV.properties: CRASTART with sample RAMs . . . . .	63	37.	START DBGMGR operator command	177
16.	crastart.conf: CRASTART with sample RAMs	63	38.	START JMON operator command. . . . .	178
17.	CRASRV.properties: Batch submit with CA Endeavor SCM RAM. . . . .	64	39.	START RSED operator command . . . . .	178
18.	CRASUBCA: Batch submit with CA Endeavor SCM RAM. . . . .	65	40.	MODIFY DBGMGR operator command	179
19.	CRASRV.properties: Batch submit with sample RAMs . . . . .	67	41.	MODIFY JMON operator command . . . . .	180
20.	CRASUBMT: Batch submit with sample RAMs	67	42.	MODIFY RSED operator command . . . . .	182
21.	CRASRV.properties – CARMA configuration file . . . . .	68	43.	STOP operator command . . . . .	190
22.	crastart*.conf: CARMA server startup using CRASTART . . . . .	73			



---

## Tables

1. Required resources . . . . .	5	15. Default CRD server transaction IDs . . . . .	101
2. Optional resources . . . . .	6	16. Push-to-client group support . . . . .	109
3. Administrators needed for required tasks	7	17. SSL certificate storage mechanisms . . . . .	110
4. Administrators needed for optional tasks	7	18. Valid keystore types . . . . .	112
5. Client checklist: Mandatory parts . . . . .	12	19. IVPs for services . . . . .	133
6. Client checklist: Optional parts . . . . .	12	20. Security setup variables . . . . .	143
7. Match load modules to functions . . . . .	20	21. JES2 Job Monitor operator commands	150
8. Sample ELAXF* procedures . . . . .	28	22. JES3 Job Monitor operator commands	151
9. ELAXF* high-level qualifier checklist . . . . .	29	23. Version 9.0 customizations . . . . .	163
10. ELAXF* . . . . .	29	24. Host Utilities version 9.0 customizations	168
11. LIMIT_COMMANDS command permission matrix . . . . .	33	25. Version 8.5 customizations . . . . .	173
12. CARMA return codes . . . . .	84	26. Thread pool error status . . . . .	184
13. SCLM administrator checklist . . . . .	95	27. Referenced publications . . . . .	197
14. Default CRD server transaction IDs . . . . .	100	28. Referenced Web sites . . . . .	199
		29. Informational publications . . . . .	200



---

## About this document

This document discusses the configuration of the IBM® Rational® Developer for System z® functions. It includes instructions to configure IBM Rational Developer for System z Version 9.0.1 on your z/OS® host system.

From here on, the following names are used in this manual:

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

This document is part of a set of documents that describe Developer for System z host system configuration. Each of these documents has a specific target audience. To complete the Developer for System z configuration, you are not required to read all documents.

- *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 IBM Rational Developer for System z Version 9.0 packages.

---

## Who should use this document

This document is intended for system programmers who are installing and configuring IBM Rational Developer for System z Version 9.0.1.

This document lists in detail the steps that are needed to do a full setup of the product, including some non-default scenarios. Background information that can help you to plan and execute the configuration can be found in the *IBM Rational Developer for System z Host Configuration Reference* (SC14-7290). To use this document, you must be familiar with the z/OS UNIX System Services and MVS™ host systems.

---

## Summary of changes

This section summarizes the changes for *IBM Rational Developer for System z Version 9.0.1 Host Configuration Guide*, SC23-7658-10 (updated December 2013).

Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

New information:

- Version 9.0.1 migration information. See “Version 9.0 migration notes” on page 160.
- New optional PARMLIB updates. See “PARMLIB changes” on page 16.
- New optional started tasks. See “PROCLIB changes” on page 23
- New operator commands. See Chapter 11, “Operator commands,” on page 177
- New and changed directives in `rzed.envvars`. See “`rzed.envvars`, the RSE configuration file” on page 35.
- New directives in `CRASRV.properties`. See “`CRASRV.properties`, the RSE interface to CARMA” on page 68.
- New configuration file for CARMA CA Endeavor<sup>®</sup> SCM RAM. See “`CRACFG`, `CRASHOW` and `CRATMAP`, the CA Endeavor<sup>®</sup> SCM RAM configuration files” on page 79.
- Support for a user exit invoked during CARMA startup. See “(Optional) CARMA user exit” on page 86.

This document contains information that was previously given in *IBM Rational Developer for System z Version 9.0 Host Configuration Guide*, SC23-7658-09.

New information:

- Version 9.0 migration information. See “Migrate from version 8.5 to version 9.0” on page 161.
- New and changed directives in `rzed.envvars`. See “`rzed.envvars`, the RSE configuration file” on page 35.
- New and removed directives in `FEJJCNFG`. See “`FEJJCNFG`, the JES Job Monitor configuration file” on page 30.
- New ELAXF\* PROCLIB members. See “ELAXF\* remote build procedures” on page 27.
- New JMON and RSED operator commands. See “Modify (F)” on page 179.
- Added information about host based code analysis. See Chapter 6, “(Optional) Host-based code analysis,” on page 105.

Removed information:

- The LOCKD started task is no longer used, so all information about the lock daemon is removed.
- The sample DB2<sup>®</sup> stored procedure has been replaced by new ELAXF\* build procedures, so all information about the DB2 stored procedure is removed.
- Migration information for unsupported releases is removed.

This document contains information that was previously given in *IBM Rational Developer for System z Version 8.5.1 Host Configuration Guide*, SC23-7658-08.

New information:

- Version 8.5.1 migration information. See “Version 8.5 migration notes” on page 169.
- New and changed directives in `rsed.envvars`. See “`rsed.envvars`, the RSE configuration file” on page 35.
- Enhanced support for CA Endeavor® SCM RAM batch actions. See “CA Endeavor® SCM RAM batch actions” on page 81.
- Installation user ID requirements. See “Installation user ID” on page 4.

This document contains information that was previously given in *IBM Rational Developer for System z Version 8.5 Host Configuration Guide*, SC23-7658-07.

New information:

- New optional directives in `FEJJCNFG`. See “`FEJJCNFG`, the JES Job Monitor configuration file” on page 30.
- New optional directives in `rsed.envvars`. See “`rsed.envvars`, the RSE configuration file” on page 35.
- New optional configuration file. See “(Optional) `include.conf`, Forced includes for C/C++ content assist” on page 113.
- New optional component. See “(Optional) Include preprocessor support” on page 115.
- New optional component. See “(Optional) xUnit support for Enterprise COBOL and PL/I” on page 116.
- New optional component. See “(Optional) DB2 and IMS debug support” on page 123.
- New and enhanced operator commands. See Chapter 11, “Operator commands,” on page 177.
- Version 8.5 migration information. See “Migrate from version 8.0.1 to version 8.5” on page 170.

Removed information:

- Support for File Manager Integration has changed, resulting in the removal of most of the information in the “(Optional) File Manager support” on page 123 section.

This document contains information that was previously given in *Rational Developer for System z Version 8.0.3 Host Configuration Guide*, SC23-7658-06.

New information:

- New directives in `rsed.envvars`. See “`rsed.envvars`, the RSE configuration file” on page 35.
- Added support for CA Endeavor® SCM background actions. See “CA Endeavor® SCM RAM batch actions” on page 81.
- Added support for CA Endeavor® SCM packages. See “CARMA Repository Access Managers (RAMs)” on page 77.
- New directives in `pushtoclient.properties`. See “(Optional) `pushtoclient.properties`, the host-based client control” on page 107.
- File Manager Integration is deprecated. See “(Optional) File Manager support” on page 123.
- New and enhanced operator commands. See Chapter 11, “Operator commands,” on page 177.

- New publication, *IBM Rational Developer for System z Messages and Codes* (SC14-7497).

This document contains information that was previously given in *Rational Developer for System z Version 8.0.1 Host Configuration Guide*, SC23-7658-05.

New information:

- New directives in FEJJCENFG. See “FEJJCENFG, the JES Job Monitor configuration file” on page 30.
- New directives in rsed.envvars. See “rsed.envvars, the RSE configuration file” on page 35.
- The CARMA chapter has been reorganized and some new information has been added. See Chapter 3, “(Optional) Common Access Repository Manager (CARMA),” on page 57.
- New configuration file pushtoclient.properties. See “(Optional) pushtoclient.properties, the host-based client control” on page 107.

Removed information:

- The information that was previously given in *Rational Developer for System z Version 7.6.1 Host Configuration Guide* (SC23-7658-04) is now split into two documents: *Rational Developer for System z Host Configuration Guide* (SC23-7658) and *Rational Developer for System z Host Configuration Reference* (SC14-7290).
- Information regarding APPC setup has moved to a white paper titled *Using APPC to provide TSO command services* (SC14-7291).
- Information regarding CARMA using ISPF Client Gateway has moved to a white paper titled *Using ISPF Client Gateway to provide CARMA services* (SC14-7292).
- “(Optional) Host-based property groups” section in “(Optional) Other customization tasks” (described propertiescfg.properties).
- “(Optional) Host-based projects” section in “(Optional) Other customization tasks” (described projectcfg.properties).
- “(Optional) Uneditable characters” section in “(Optional) Other customization tasks” (described uchars.settings).
- “Version 7.6.1 migration notes” section in “Migration guide”.

---

## Description of the document content

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

### Planning

Use the information in this chapter to plan the installation and deployment of Developer for System z.

### Basic customization

The following customization steps are for a basic Developer for System z setup:

- “Customization setup” on page 15
- “PARMLIB changes” on page 16
- “PROCLIB changes” on page 23
- “Security definitions” on page 29
- “FEJJCENFG, the JES Job Monitor configuration file” on page 30
- “rsed.envvars, the RSE configuration file” on page 35

- “ISPF.conf, the ISPF's TSO/ISPF Client Gateway configuration file” on page 52

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

Common Access Repository Manager (CARMA) is a server platform for Repository Access Managers (RAMs). A RAM is an Application Programming Interface (API) for a Software Configuration Manager (SCM) that is based on a z/OS system. 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 and source code examples for creating your own RAM.

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

## **(Optional) SCLM Developer Toolkit**

SCLM Developer Toolkit provides the tools that are needed to extend the capabilities of SCLM to the client. SCLM itself is a host-based source code manager that is included in ISPF.

The SCLM Developer Toolkit has an Eclipse-based plug-in that interfaces to SCLM and provides access to all SCLM processes for heritage code development and support for full Java<sup>™</sup> and J2EE development on the workstation with synchronization to SCLM on the mainframe. The synchronization activities include building, assembling, and deployment of the J2EE code from the mainframe.

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

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<sup>®</sup> 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
  - CICS resources defined by application developers in a limited, controlled, and secure fashion
  - CICS development access not available 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) Host-based code analysis**

Similar to the Developer for System z client, the Developer for System z host supports running code analysis tools, which are provided as a separate product, Rational Developer for System z Host Utilities. A benefit of doing code analysis on the host is that it can be integrated in your daily batch processing.

The following code analysis tools are available on the host:

- Code review: Using rules with different severity levels, code review scans source code and reports rule violations.
- Code coverage: Analyze a running program and generate a report of lines that are executed, compared to the total number of executable lines.

## (Optional) Other customization tasks

This section combines a variety of optional customization tasks. To configure the required service, follow the instructions in the appropriate section.

Customizations to Developer for System z configuration files:

- pushtoclient.properties, Host-based client control
- ssl.properties, RSE SSL encryption
- rsecomm.properties, RSE tracing
- include.conf, Forced includes for C/C++ content assist

Developer for System z related customizations to or for other products:

- DB2 stored procedure
- z/OS UNIX subprojects
- Include preprocessor support
- xUnit support for Enterprise COBOL and PL/I
- Enterprise Service Tools support
- CICS bidirectional language support
- Diagnostic IRZ messages for generated code
- Integrated Debugger
- Problem Determination Tools support
- DB2 and IMS™ debug support
- File Manager support
- WORKAREA and /tmp cleanup

## Installation verification

After completing the product customization, you can verify the successful setup of key product components by using the Installation Verification Programs (IVPs) described in this chapter.

## Security definitions

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

## Migration guide

This section highlights the installation and configuration changes compared to the previous releases of the product. It also gives some general guidelines to migrate to this release.

## Operator commands

This section provides an overview of the available operator (or console) commands for Developer for System z.

## Host configuration reference

This section summarizes the information in *IBM Rational Developer for System z Host Configuration Reference* (SC14-7290).



---

# IBM Rational Developer for System z Host Configuration Guide



---

## Chapter 1. Planning

Use the information in this chapter, *IBM Rational Developer for System z Prerequisites* (SC23-7659), to plan the installation and deployment of Developer for System z. The following subjects are described:

- “Migration considerations”
- “Planning considerations”
- “Preinstallation considerations” on page 4
- “Pre-configuration considerations” on page 9
- “Predeployment considerations” on page 11
- “Client checklist” on page 12

---

### Migration considerations

Chapter 10, “Migration guide,” on page 159 describes the installation and configuration changes compared to previous releases of the product. Use this information to plan your migration to the current release of Developer for System z.

**Note:**

- If you are a previous user of IBM Rational Developer for System z, IBM WebSphere Developer for System z, IBM WebSphere Developer for zSeries or IBM WebSphere Studio Enterprise Developer, save the related customized files before installing IBM Rational Developer for System z Version 9.0. For an overview of files that required customization, see Chapter 10, “Migration guide,” on page 159.
- If you plan to run multiple instances of Developer for System z, see “Running multiple instances” in the *Host Configuration Reference* (SC14-7290).

---

### Planning considerations

#### Product overview

Developer for System z consists of a client, installed on the user's personal computer, and a server, installed on one or more host systems. This documentation contains information for a z/OS host system. However, other operating systems, such as AIX<sup>®</sup> and Linux on System z, are also supported.

The client provides developers with an Eclipse-based development environment that facilitates a uniform graphical interface to the host, and that, among other things, can offload work from the host to the client, saving resources on the host.

The host portion consists of several permanently active tasks and tasks that are started ad hoc. These tasks allow the client to work with the various components of your z/OS host system, such as MVS data sets, TSO commands, z/OS UNIX files and commands, job submit, and job output.

Developer for System z can also interact with subsystems and other application software on the host system, such as CICS, IBM File Manager, and Software Configuration Managers (SCMs), if Developer for System z is configured to do so, and if these co-requisite products are available.

For a basic understanding of the Developer for System z design, see "Understanding Developer for System z" in the *Host Configuration Reference* (SC14-7290).

To learn more about the functionality that is offered by Developer for System z, see the Developer for System z website, <http://www-03.ibm.com/software/products/us/en/developerforsystemz/>, or your local IBM representative.

## Skill requirements

SMP/E skills are needed for a Developer for System z host installation.

The configuration of Developer for System z requires more than the typical system programming permissions and expertise, so assistance from others might be needed. Table 3 on page 7 and Table 4 on page 7 list the administrators who are needed for the required and optional customization tasks.

## Time requirements

The amount of time that is required to install and configure the Developer for System z host system components depends on various factors such as these:

- The current z/OS UNIX and TCP/IP configuration
- The availability of prerequisite software and maintenance
- Whether OMVS segments are defined for Developer for System z users
- The availability of a user, who has successfully installed the client, to test the installation and report any problems that might occur

Experience has shown that the installation and configuration process of the Developer for System z host system requires from one to four days to complete. This time requirement is for a clean installation performed by an experienced system programmer. If problems are encountered, or if the required skills are not available, the setup will take longer.

---

## Preinstallation considerations

For detailed instructions on the SMP/E installation of the product, see *Program Directory for IBM Rational Developer for System z* (GI11-8298).

To run multiple instances of Developer for System z, see "Running multiple instances" in the *Host Configuration Reference* (SC14-7290).

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

## Installation user ID

The user ID that is used to install Developer for System z, or to install maintenance, must have at least the following attributes:

- TSO access (with a normal region size).
- An OMVS segment defined to the security system (for example, RACF), both for the user ID and its default group.
  - The HOME field must refer to a home directory that is allocated for the user, with READ, WRITE, and EXECUTE access.

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

## Requisite products

*IBM Rational Developer for System z Prerequisites (SC23-7659)* has a list of prerequisite software that must be installed and operational before Developer for System z 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.

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

- z/OS 1.8 or higher
- ISPF APAR OA43014 (TSO/ISPF Client Gateway)
- Java 6.0 or higher (31 or 64 bit)

## Required resources

Developer for System z requires the allocation of the systems resources listed in Table 1. The resources listed in Table 2 on page 6 are required for optional services. Plan to have these resources available because, depending on the policies at your site, it might take some time to get the software.

*Table 1. Required resources*

Resource	Default value	Information
APF-authorized data set	FEK.SFEKAUTH	"APF authorizations in PROGxx" on page 19
started task	JMON and RSED	"PROCLIB changes" on page 23
port for host-confined use (JMON)	6715	"FEJJCNFG, the JES Job Monitor configuration file" on page 30
port for client-host communication (RSED)	4035	"rsed.envvars, the RSE configuration file" on page 35
port range for client-host communication (RSED)	any available port is used	"Defining the PORTRANGE available for RSE server" on page 44
z/OS UNIX server security definition	UPDATE permission to BPX.SERVER for RSED started task	"Define RSE as a secure z/OS UNIX server" on page 147
PassTicket security definitions	no default	"Define the PassTicket support for RSE" on page 148

Table 1. Required resources (continued)

Resource	Default value	Information
MVS build procedures	ELAXF*	“PROCLIB changes” on page 23

Table 2. Optional resources

Resource	Default value	Information
IPL with CLPA	not applicable	“(Optional) Integrated debugger” on page 120
started task	DBGMGR	“(Optional) Integrated debugger” on page 120
LINKLIST data set	FEK.SFEKAUTH and FEK.SFEKLOAD	<ul style="list-style-type: none"> <li>Chapter 4, “(Optional) SCLM Developer Toolkit,” on page 89</li> <li>“(Optional) Integrated debugger” on page 120</li> </ul>
LPA data set	FEK.SFEKLPA	Chapter 3, “(Optional) Common Access Repository Manager (CARMA),” on page 57
port range for host-confined use	any available port is used	<ul style="list-style-type: none"> <li>Chapter 3, “(Optional) Common Access Repository Manager (CARMA),” on page 57</li> <li>Chapter 6, “(Optional) Host-based code analysis,” on page 105</li> </ul>
port range for host-confined use	5336	“(Optional) Integrated debugger” on page 120
port for client-host communication	<ul style="list-style-type: none"> <li>5129 for Web Service or 5130 for RESTful services</li> <li>5335 for Integrated Debugger</li> </ul>	<ul style="list-style-type: none"> <li>Chapter 5, “(Optional) Application Deployment Manager (deprecated),” on page 97</li> <li>“(Optional) Integrated debugger” on page 120</li> </ul>
CICS CSD update	multiple values	<ul style="list-style-type: none"> <li>Chapter 5, “(Optional) Application Deployment Manager (deprecated),” on page 97</li> <li>“(Optional) Integrated debugger” on page 120</li> </ul>
CICS JCL update	<ul style="list-style-type: none"> <li>FEK.SFEKLOAD</li> <li>FEK.SFEKAUTH</li> </ul>	<ul style="list-style-type: none"> <li>Chapter 5, “(Optional) Application Deployment Manager (deprecated),” on page 97</li> <li>“(Optional) CICS bidirectional language support” on page 118</li> <li>“(Optional) Integrated debugger” on page 120</li> </ul>

The configuration of Developer for System z requires more than the typical system programming permissions and expertise; therefore, assistance from others might be needed. Table 3 and Table 4 list the administrators who are needed for the required and optional customization tasks.

*Table 3. Administrators needed for required tasks*

<b>Administrator</b>	<b>Task</b>	<b>Information</b>
System	Typical system programmer actions are required for all customization tasks	N/A
Security	<ul style="list-style-type: none"> <li>• Define OMVS segment for Developer for System z users</li> <li>• Define data set profiles</li> <li>• Define started tasks</li> <li>• Define operator command security</li> <li>• Define z/OS UNIX server profiles</li> <li>• Define application security</li> <li>• Define PassTicket support</li> <li>• Define program controlled data sets</li> <li>• Define program controlled z/OS UNIX files</li> </ul>	"Security considerations" in <i>Host Configuration Reference</i> (SC14-7290)
TCP/IP	Define new TCP/IP ports	"TCP/IP considerations" in <i>Host Configuration Reference</i> (SC14-7290)
WLM	Assign the started task goals to the servers and their child processes	"WLM considerations" in the <i>Host Configuration Reference</i> (SC14-7290).

*Table 4. Administrators needed for optional tasks*

<b>Administrator</b>	<b>Task</b>	<b>Information</b>
System	Typical system programmer actions are required for all customization tasks	N/A

Table 4. Administrators needed for optional tasks (continued)

Administrator	Task	Information
Security	<ul style="list-style-type: none"> <li>• Define data set profiles</li> <li>• Define program controlled data sets</li> <li>• Define permission to submit xxx* jobs</li> <li>• Define CICS transaction security</li> <li>• Add certificate for SSL</li> <li>• Define X.509 client certificate support</li> <li>• Define groups and profiles for push-to-client</li> <li>• Define profiles for altering client functions</li> <li>• Define started tasks</li> <li>• Define z/OS UNIX server profiles</li> <li>• Define profiles for authorized debugging</li> </ul>	<ul style="list-style-type: none"> <li>• "Security considerations" in <i>Host Configuration Reference</i> (SC14-7290)</li> <li>• "CICSTS considerations" in the <i>Host Configuration Reference</i> (SC14-7290)</li> <li>• "Setting up SSL and X.509 authentication" in the <i>Host Configuration Reference</i> (SC14-7290)</li> </ul>
TCP/IP	Define new TCP/IP ports	"TCP/IP ports" in <i>Host Configuration Reference</i> (SC14-7290)
SCLM	<ul style="list-style-type: none"> <li>• Define SCLM language translators for Java EE support</li> <li>• Define SCLM types for Java EE support</li> </ul>	Chapter 4, "(Optional) SCLM Developer Toolkit," on page 89
CICS TS	<ul style="list-style-type: none"> <li>• Update CICS region JCL</li> <li>• Update CICS region CSD</li> <li>• Define CICS group</li> <li>• Define CICS transaction names</li> <li>• Define a program to CICS</li> <li>• Define debugger to CICS</li> </ul>	<ul style="list-style-type: none"> <li>• Chapter 5, "(Optional) Application Deployment Manager (deprecated)," on page 97</li> <li>• "(Optional) CICS bidirectional language support" on page 118</li> <li>• "(Optional) Integrated debugger" on page 120</li> </ul>
WLM	Assign goals to Developer for System z tasks	"WLM considerations" in the <i>Host Configuration Reference</i> (SC14-7290)
LDAP	Define groups for push-to-client	"Push-to-client considerations" in the <i>Host Configuration Reference</i> (SC14-7290)

---

## Pre-configuration considerations

For information about Developer for System z itself, how it interacts with your system, and with the prerequisite and co-requisite products, see *Host Configuration Reference* (SC14-7290). This information can assist you in creating a setup that supports your current needs and future growth.

### Workload management

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 system services and data. To plan your WLM configuration, see "WLM considerations" in the *Host Configuration Reference* (SC14-7290).

**Note:** Developer for System z consists of multiple tasks that communicate with each other and the client. These tasks use various timers to detect communication loss with their partners. Timeout issues can arise (due to lack of CPU time during the timeout window) on systems with a heavy CPU load or incorrect Workload Management (WLM) settings for Developer for System z.

### Resource usage and system limits

Developer for System z uses a variable number of system resources such as address spaces, and z/OS UNIX processes and threads. The availability of these resources is limited by various system definitions. To estimate the usage of key resources so that you can plan your system configuration, see "Tuning considerations" in the *Host Configuration Reference* (SC14-7290). Developer for System z can run in either 31-bit or 64-bit mode, changing the storage resource limitations drastically.

### Required configuration of requisite products

Consult your MVS system programmer, security administrator, and TCP/IP administrator to verify if the requisite products and software are installed, tested, and working. Some requisite customization tasks that can be overlooked are listed here:

- All Developer for System z users must have READ and EXECUTE access to the Java directories.
- Remote (host-based) actions for z/OS UNIX subprojects require that z/OS UNIX version of REXEC or SSH is active on the host system.

### User ID considerations

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

- TSO access (with a normal region size).

**Note:** A large region size is required for the user ID that runs the Installation Verification Programs (IVPs) because functions requiring a lot of memory (such as Java) are executed. You should set the region size to 131072 kilobytes (128 megabytes) or higher.

- An OMVS segment defined to the security system (for example, RACF), both for the user ID and its default group.
  - The HOME field must refer to a home directory allocated for the user (with READ, WRITE and EXECUTE access).

- The PROGRAM field in the OMVS segment should be /bin/sh or other valid z/OS UNIX shell, such as /bin/tcsh.
- The ASSIZEMAX field should not be set, so that system defaults are used.
- The user ID does not require UID 0.

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

USER=userid

```
OMVS INFORMATION
-----
UID= 0000003200
HOME= /u/userid
PROGRAM= /bin/sh
CPUTIMEMAX= NONE
ASSIZEMAX= NONE
FILEPROCMAx= NONE
PROCUSERMAX= NONE
THREADSMAX= NONE
MMAPAREAMAX= NONE
```

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

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

GROUP group

```
OMVS INFORMATION
-----
GID= 0000003243
```

- READ and EXECUTE access to the Developer for System z installation and configuration directories and files, default /usr/lpp/rdz/\*, /etc/rdz/\*, and /var/rdz/\*.
- READ, WRITE, and EXECUTE access to the Developer for System z WORKAREA directory, default /var/rdz/WORKAREA, and user log directory, default /var/rdz/logs.
- READ access to the Developer for System z installation data sets, default FEK.SFEK\*.
- READ, WRITE, and EXECUTE access to the /tmp directory or a directory referenced in the TMPDIR environment variable.

## Server considerations

Developer for System z consists of multiple 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. There is no specific startup order. The only requirement is that the servers are up and running before the first user tries to connect. The security mechanisms used by Developer for System z servers and services rely on the data sets and file systems they reside in being secure. This implies that only trusted system administrators should be able to update the program libraries and configuration files.

- Debug Manager (DBGMGR) provides debug-related services.
- JES Job Monitor (JMON) provides all JES-related services.
- Remote Systems Explorer (RSE) provides core services such as connecting the client to the host system and starting other servers for specific services. RSE consists of two logical entities:
  - RSE daemon (RSED), which manages connection setup, and which is responsible for running in single server mode.
  - RSE server, which handles individual client requests.

As documented in "TCP/IP ports" in *Host Configuration Reference* (SC14-7290), certain host system services, and thus their ports, must be available for the client to connect to, and must be defined to your firewall protecting the host system. All other ports used by Developer for System z have host-only traffic. Listed below are the ports that are needed for external communication in a basic Developer for System z setup.

- RSE daemon for client-host communication setup (using TCP), default port 4035.
- RSE server for client-host communication (using TCP). By default, any available port is used, but the available ports can be limited to a specified range.

## Configuration method

Developer for System z provides alternative methods to configure the host system side of the product. These are the methods:

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

---

## Predeployment considerations

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

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

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

Developer for System z

- FEK.SFEKAUTH(\*)
- FEK.SFEKLOAD(\*)
- FEK.SFEKPROC(\*)
- FEK.#CUST.PARMLIB(\*)
- FEK.#CUST.PROCLIB(\*)
- /usr/lpp/rdz/\*
- /etc/rdz/\*
- /var/rdz/\* (directory structure only)
- optional parts:
  - FEK.SFEKLPA(\*)
  - FEK.SFEKLMOD(\*)
  - FEK.#CUST.CNTL(\*)
  - definitions, data sets, files, and directories resulting from customization jobs in FEK.#CUST.JCL

Developer for System z Host Utilities

- AKG.SAKGPROC(\*)
- /usr/lpp/rdzutil/\*

**Note:**

- FEK and /usr/lpp/rdz are the high-level qualifier and path used during the installation of the Developer for System z. FEK.#CUST, /etc/rdz and /var/rdz are the default locations used during the customization of the product (see “Customization setup” on page 15 for more information).
- AKG and /usr/lpp/rdzutil are the high-level qualifier and path used during the installation of Developer for System z Host Utilities.
- You should install Developer for System z in a private file system (HFS or zFS) to ease the deploying of the z/OS UNIX parts of the product.
- If you cannot use a private file system, use an archiving tool such as the z/OS UNIX tar command to transport the z/OS UNIX directories from one system to another. This method is for preserving the attributes (such as program control) for the Developer for System z files and directories.

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

- Archive: `cd /SYS1/usr/lpp/rdz; tar -cSf /u/userid/rdz.tar`
- Restore: `cd /SYS2/usr/lpp/rdz; tar -xSf /u/userid/rdz.tar`

---

## Client checklist

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

The checklist in Table 5 lists the required results of mandatory customization steps. Table 6 lists the required results of optional customization steps.

*Table 5. Client checklist: Mandatory parts*

Customization	Value
RSE daemon TCP/IP port number. The default is 4035. See “RSED, RSE daemon started task” on page 25.	

*Table 6. Client checklist: Optional parts*

Customization	Value
Location of the ELAXF* procedures if they are not in a system procedure library. The default FEK.#CUST.PROCLIB. See the note on JCLLIB in “ELAXF* remote build procedures” on page 27.	
Procedure or step names of the ELAXF* procedures if they were changed. See the note on changing them in “ELAXF* remote build procedures” on page 27.	
Location of the AKGCR procedure if it is not in a system procedure library. The default is AKG.#CUST.PROCLIB. See the note on JCLLIB in “Code review” on page 105.	

Table 6. Client checklist: Optional parts (continued)

Customization	Value
<p>Location of the AKGCC procedure if it is not in a system procedure library. The default is AKG.#CUST.PROCLIB.</p> <p>See the note on JCLLIB in “Code coverage” on page 106.</p>	
<p>Location of the FEKRNPLI Include Preprocessor exec statement. The default is FEK.#CUST.CNTL.</p> <p>See “(Optional) Include preprocessor support” on page 115.</p>	
<p>Location of the debugger load modules if not in LINKLIST. The default is FEK.SFEKAUTH. See “(Optional) Integrated debugger” on page 120</p>	
<p>Location of the unit test load modules if not in LINKLIST or STEPLIB of rsed.envvars. The default is FEK.SFEKLOAD.</p> <p>See “(Optional) xUnit support for Enterprise COBOL and PL/I” on page 116.</p>	
<p>Location of the AZUZUNIT procedure if it is not in a system procedure library. The default is FEK.#CUST.PROCLIB.</p> <p>See the note on JCLLIB in “(Optional) xUnit support for Enterprise COBOL and PL/I” on page 116.</p>	
<p>Location of the sample *.xsd and *.xsl XML files used for unit test output formatting. The defaults are /usr/lpp/rdz/samples/zunit/xsd and /usr/lpp/rdz/samples/zunit/xsl.</p> <p>See “(Optional) xUnit support for Enterprise COBOL and PL/I” on page 116.</p>	
<p>(co-requisite) TN3270 port number for Host Connect Emulator. The default is 23.</p> <p>See “TCP/IP ports” in <i>Host Configuration Reference</i> (SC14-7290).</p>	
<p>(co-requisite) REXEC or SSH port number, which, by default are 512 or 22.</p> <p>See “(Optional) z/OS UNIX subprojects” on page 115.</p>	
<p>(co-requisite) Debug Tool server port number (no default).</p> <p>See “(Optional) DB2 and IMS debug support” on page 123.</p>	
<p>Application Deployment Manager port number, which by default is 5129 for web service or 5130 for REST service.</p> <p>See “TCP/IP ports” in <i>Host Configuration Reference</i> (SC14-7290).</p>	
<p>Location of the SFEKSAMP sample library for CARMA RAM samples. The default is FEK.SFEKSAMP.</p> <p>See the <i>CARMA Developer’s Guide</i> (SC23-7660).</p>	
<p>Location of the CRA#ASLM JCL for CARMA SCLM RAM data set allocations. The default is FEK.#CUST.JCL.</p> <p>See the note on CRA#ASLM in “SCLM RAM” on page 78.</p>	



---

## Chapter 2. Basic customization

The following customization steps are for a basic Developer for System z setup. See the chapters about the optional components for their customization requirements.

---

### Requirements and checklist

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

- APF-authorized data set
- Various PARMLIB updates
- Various security software updates
- Various TCP/IP ports for internal and client-host communication
- IPL to activate an optional SVC

To verify the installation and to start using Developer for System z at your site, do the following tasks. Unless otherwise indicated, all tasks are mandatory.

1. Create customizable copies of samples and create the work environment for Developer for System z. For details, see “Customization setup.”
2. Update z/OS UNIX system limits, start started tasks, and define APF-authorized and LINKLIST data sets and, optionally, SVCs and LPA data sets. For details, see “PARMLIB changes” on page 16.
3. Create started task procedures, and compile and link procedures. For details, see “PROCLIB changes” on page 23.
4. Update security definitions. For details, see “Security definitions” on page 29. To establish thread security, you must understand how PassTickets are used. See “Using PassTickets” in *Host Configuration Reference* (SC14-7290).
5. Customize Developer for Developer for System z configuration files. For details, see:
  - “FEJCNFG, the JES Job Monitor configuration file” on page 30
  - “rsed.envvars, the RSE configuration file” on page 35
  - “ISPF.conf, the ISPF's TSO/ISPF Client Gateway configuration file” on page 52

---

### Customization setup

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

Some functions of Developer for System z 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 system configuration when using the Host Configuration utility. The FEKSETUP job and the utility do some of the same tasks, with no way of checking if those tasks have already been performed. Therefore, it is possible to undo changes that have already been made. For this reason, do not use both methods for a single installation.

To create customizable copies of configuration files and configuration JCL, and to create required z/OS UNIX directories, customize and submit the sample FEKSETUP member in the FEK.SFEKSAMP data set. 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 FEK.#CUST.SQL and populate it with sample SQL command files.
- 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.

**Note:**

- The configuration steps in this publication use the member and file locations created by the FEKSETUP job, unless noted otherwise. The original samples, which should not be updated, are in FEK.SFEKSAMP and /usr/lpp/rdz/samples/.
- For more details on which sample members are copied to which data set, and for more details on which directories are created, their permission bitmask, and where the various sample files are copied to, see the comments in FEK.SFEKSAMP(FEKSETUP).
- To aid in migrating an existing setup, the comments in FEK.SFEKSAMP(FEKSETUP) also document the changes between different versions of Developer for System z.
- If you want to keep all of the Developer for System z z/OS UNIX files in the same file system (HFS or zFS), but also want the configuration files placed in /etc/rdz, you can use symbolic links to solve this problem. The following sample z/OS UNIX commands create a new directory in the existing file system (/usr/lpp/rdz/cust) and define a symbolic link (/etc/rdz) to it:

```
mkdir /usr/lpp/rdz/cust
ln -s /usr/lpp/rdz/cust /etc/rdz
```

---

## PARMLIB changes

The following PARMLIB changes are documented in this section:

- “Set the z/OS UNIX limits in BPXPRMxx” on page 17
- “Add the started tasks to COMMNDxx” on page 18
- “SVC definitions in IEASVCxx” on page 18
- “LPA definitions in LPALSTxx” on page 19
- “APF authorizations in PROGxx” on page 19

- “LINKLIST definitions in PROGxx” on page 20
- “Requisite LINKLIST and LPA definitions” on page 22
- “LINKLIST definitions for other products” on page 23

For more information about the PARMLIB definitions listed in the next sections, see *MVS Initialization and Tuning Reference* (SA22-7592). For more information about the sample console commands, see *MVS System Commands* (SA22-7627).

## Set the z/OS UNIX limits in BPXPRMxx

Remote Systems Explorer (RSE), which provides core services such as connecting the client to the host system, is a z/OS UNIX based process. Therefore, it is important to set correct values for the z/OS UNIX system limits in BPXPRMxx, based upon the number of concurrently active Developer for System z users and their average workload. Define OMVS=xx in the IEASYSxx parmlib member to specify which BPXPRMxx parmlib member should be used during IPL.

See "Tuning considerations" in the *Host Configuration Reference* (SC14-7290) for more information about different BPXPRMxx defined limits and their impact on Developer for System z.

MAXASSIZE specifies the maximum address space (process) region size. Set MAXASSIZE in SYS1.PARMLIB(BPXPRMxx) to 2G. This is the maximum value allowed. This is a system-wide limit, and thus active for all z/OS UNIX address spaces. If this is not what you want, you can set the limit only for Developer for System z in your security software, as described in “Define the Developer for System z started tasks” on page 146.

MAXTHREADS specifies the maximum number of active threads for a single process. Set MAXTHREADS in SYS1.PARMLIB(BPXPRMxx) to 1500 or higher. This is a system-wide limit, and thus active for all z/OS UNIX address spaces. If this is not what you want, you can set the limit only for Developer for System z in your security software, as described in “Define the Developer for System z started tasks” on page 146.

MAXTHREADTASKS specifies the maximum number of active MVS tasks for a single process. Set MAXTHREADTASKS in SYS1.PARMLIB(BPXPRMxx) to 1500 or higher. This is a system-wide limit, and thus active for all z/OS UNIX address spaces. If this is not what you want, you can set the limit only for Developer for System z in your security software, as described in “Define the Developer for System z started tasks” on page 146.

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. This setting is intended to be a system-wide limit, because it should be active for each client that uses Developer for System z.

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

**Note:**

- For more information about other locations where address space sizes can be set or limited, see "Address space size" in the *Host Configuration Reference* (SC14-7290).
- The MAXPROCUSER value suggested here is based upon users having a unique z/OS UNIX user ID (UID). Increase this value if your users share the same UID.
- Ensure that other BPXPRMxx values, such as those for MAXPROCSYS and MAXUIDS, are sufficient to handle the expected amount of concurrently active Developer for System z users. See "Tuning considerations" in the *Host Configuration Reference* (SC14-7290) for more details.
- During the SMP/E install of Developer for System z, you were advised to place the code in a separate file system (zFS or HFS) and update BPXPRMxx to mount this file system during system IPL. Included is a repeat of the sample mount command in case this update still must be done:

```

MOUNT FILESYSTEM('#dsn')
  MOUNTPOINT('-PathPrefix-usr/lpp/rdz')
  MODE(RDWR) /* can be MODE(READ) */
  TYPE(ZFS) PARM('AGGRGROW') /* zFS, with extents */
/* TYPE(HFS) */ /* HFS, auto. extent */

```

- During the SMP/E install of Developer for System z Host Utilities, you were advised to place the code in a separate file system (zFS or HFS) and update BPXPRMxx to mount this file system during system IPL. Included is a repeat of the sample mount command in case this update still must be done:

```

MOUNT FILESYSTEM('#dsn')
  MOUNTPOINT('-PathPrefix-usr/lpp/rdzutil')
  MODE(RDWR) /* can be MODE(READ) */
  TYPE(ZFS) PARM('AGGRGROW') /* zFS, with extents */
/* TYPE(HFS) */ /* HFS, auto. extent */

```

## Add the started tasks to COMMNDxx

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

The optional Integrated Debugger requires that the Developer for System z DBGMR server is active on your system.

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

- S RSED
- S JMON
- S DBGMR

**Note:** There is no specific startup order for the servers. The only requirement is that the servers are up and running before the first user tries to connect.

## SVC definitions in IEASVCxx

The optional Integrated Debugger is capable of debugging CICS transactions loaded into read-only memory. This requires that a Developer for System z supervisor call (SVC) is defined to your system.

| Installation-defined SVCs are defined in SYS1.PARMLIB(IEASVCxx) and require an  
| IPL to be activated. The related load module must be loaded in LPA at IPL time.  
| Define SVC=xx in the IEASYSxx parmlib member to specify which IEASVCxx parmlib  
| member should be used during IPL.

| Specify the following in IEASVCxx to define the Developer for System z SVC:  
| SVC Parm 251,REPLACE,TYPE(4),EPNAME(AQESVC01) /\* RDz debug \*/

| If you are not using the default SVC number, change value 251 in the SVC Parm  
| definition to the number you have chosen and update the SVC startup parameter in  
| the DBG MGR started task JCL.

## LPA definitions in LPALSTxx

The optional Common Access Repository Manager (CARMA) service supports different server startup methods for the CARMA server. The CRASTART startup method requires that the CRASTART module in the FEK.SFEKLPA load library is in the Link Pack Area (LPA).

| The optional Integrated Debugger is capable of debugging CICS transactions  
| loaded into read-only memory. This requires that load module AQESVC01 in the  
| FEK.SFEKLPA load library is in the Link Pack Area (LPA) during IPL time.

| LPA data sets are defined in SYS1.PARMLIB(LPALSTxx). Define LPA=xx in the  
| IEASYSxx parmlib member to specify which LPALSTxx parmlib member should be  
| used during IPL.

LPA definitions can be set dynamically (until the next IPL) with the following console command:

- SETPROG LPA,ADD,DSN=FEK.SFEKLPA

### Note:

- Data sets listed in LPALSTxx must be cataloged in the master catalog or a user catalog identified in the LPALSTxx member.
- Adding a new data set to LPALSTxx requires an IPL with CLPA (create LPA) to be activated.
- All libraries that are loaded into LPA are automatically considered to be APF-authorized and program controlled. Ensure you have proper security controls in place for these libraries.
- If you choose to not place a library designed for LPA placement in LPA and you use LINKLIST or STEPLIB instead, ensure that you define the APF authorization and program control status.

## APF authorizations in PROGxx

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

| For the optional Debug Manager to work, the AQEZPCM module in the FEK.SFEKAUTH  
| load library must be APF-authorized.

For the optional SCLM Developer Toolkit service to work, the REXX runtime library (REXX.\*.SEAGLPA) must be APF-authorized.

For ISPF to create the TSO/ISPF Client Gateway, the ISPZTS0 module in SYS1.LINKLIB must be APF-authorized. The TSO/ISPF Client Gateway is used by Developer for System z's TSO Commands service and SCLM Developer Toolkit.

APF authorizations are defined in SYS1.PARMLIB(PROGxx) by default. Define PROG=xx in the IEASYSxx parmlib member to specify which PROGxx parmlib member should be used during IPL.

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=REXX.V1R4M0.SEAGLPA,VOL=volser
- SETPROG APF,ADD,DSN=SYS1.LINKLIB,VOL=volser

**Note:**

- When you use the Alternate Library for REXX product package, the default REXX runtime library name is REXX.\*.SEAGALT, instead of REXX.\*.SEAGLPA as used in the preceding sample.
- LPA libraries, such as REXX.\*.SEAGLPA, are automatically APF-authorized when located in LPA, and thus do not require explicit definitions.
- Some of the co-requisite products, such as IBM File Manager, also require APF authorization. See the related product customization guides for more information.

## LINKLIST definitions in PROGxx

LINKLIST definitions for Developer for System z can be grouped in three categories:

- Developer for System z load libraries that are needed for Developer for System z functions. These definitions are described in this section.
- Requisite load libraries that are needed for Developer for System z functions. These definitions are described in “Requisite LINKLIST and LPA definitions” on page 22.
- Developer for System z load libraries that are needed by other products. These definitions are described in “LINKLIST definitions for other products” on page 23.

*Table 7. Match load modules to functions*

Load library	Load modules	Usage	STEPLIB
FEK.SFEKAUTH	AQE* and CEE*	“(Optional) Integrated debugger” on page 120	ELAXFGO procedure, or CICS
	FEJJ*	“PROCLIB changes” on page 23  (JES Job Monitor started task)	Started task procedure
FEK.SFEKLMOD	IRZ* and IIRZ*	“(Optional) Diagnostic IRZ messages for the generated code” on page 118	CICS, IMS, or MVS batch

Table 7. Match load modules to functions (continued)

Load library	Load modules	Usage	STEPLIB
FEK.SFEKLOAD	AND*	Chapter 5, “(Optional) Application Deployment Manager (deprecated),” on page 97	CICS
	AZU* and IAZU*	“(Optional) xUnit support for Enterprise COBOL and PL/I” on page 116	rzed.envvars or MVS batch
	BWB*	Chapter 4, “(Optional) SCLM Developer Toolkit,” on page 89	rzed.envvars
	CRA*	Chapter 3, “(Optional) Common Access Repository Manager (CARMA),” on page 57	CRASUB* or crastart*.conf
	ELAX*	“ELAXF* remote build procedures” on page 27  (error feedback and include preprocessor)	ELAXF* procedures
	FEJB*	“(Optional) CICS bidirectional language support” on page 118	CICS
FEK.SFEKLPA	CRA*	Chapter 3, “(Optional) Common Access Repository Manager (CARMA),” on page 57	CRASRV.properties

In order for the listed Developer for System z services to work, all modules documented in Table 7 on page 20 that are related to the service must be made available either through STEPLIB or LINKLST (or LPA). Note that the SFEKLMOD library is not used by Developer for System z itself, but by code generated by Developer for System z. See the STEPLIB column in Table 7 on page 20 if you choose to use STEPLIB to learn where the STEPLIB (or DFHRPL for CICS) definition must be made. However, you should be aware of the following things:

- 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.

LINKLIST data sets are defined in SYS1.PARMLIB(PROGxx), if your site followed IBM recommendations. Define PROG=xx in the IEASYSxx parmlib member to specify which PROGxx parmlib member should be used during IPL.

The required definitions will look like the following, where listname is the name of the LINKLIST set that will be activated, and volser is the volume on which the data set resides if it is not cataloged in the master catalog:

- LNKLST ADD NAME(listname) DSNAME(FEK.SFEKAUTH) VOLUME(volser)
- LNKLST ADD NAME(listname) DSNAME(FEK.SFEKLOAD)

LINKLIST definitions can be created dynamically (until the next IPL) with the following group of console commands, where volser is the volume on which the data set resides if it is not cataloged in the master catalog:

1. LNKLST DEFINE,NAME=LLTMP,COPYFROM=CURRENT
2. LNKLST ADD NAME=LLTMP,DSN=FEK.SFEKAUTH,VOL=volser
3. LNKLST ADD NAME=LLTMP,DSN=FEK.SFEKLOAD
4. LNKLST ACTIVATE,NAME=LLTMP

## Requisite LINKLIST and LPA definitions

Remote Systems Explorer (RSE) is a z/OS UNIX process that requires access to MVS load libraries. The JES Job Monitor and Integrated Debugger servers also need access to the system, Language Environment (LE) and C libraries. The following prerequisite libraries must be made available, either through STEPLIB or LINKLIST/LPALIB:

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

The following additional libraries must be made available, either through STEPLIB or LINKLIST/LPALIB, to support the use of optional services. This list does not include data sets that are specific to a product that Developer for System z interacts with, such as IBM File Manager for z/OS:

- REXX runtime library (for SCLM Developer Toolkit and RSE\_DSICALL=TSO)
  - REXX.\*.SEAGLPA
- System load library (for SSL encryption)
  - SYS1.SIEALNKE
- System load library (for Integrated Debugger on z/OS 1.13 and higher)
  - SYS1.SIEAMIGE
- System load library (for Enterprise COBOL and PL/I unit test)
  - SYS1.CSSLIB
  - SYS1.SIXMLOD1

### Note:

- When you use the Alternate Library for REXX product package, the default REXX runtime library name is REXX.\*.SEAGALT, instead of REXX.\*.SEAGLPA as used in the preceding sample.
- All libraries that are loaded into LPA are automatically considered to be APF-authorized and program controlled. Ensure you have proper security controls in place for these libraries.
- Libraries that are designed for LPA placement, such as REXX.\*.SEAGLPA, might require additional program control or APF authorizations if they are accessed through LINKLIST or STEPLIB.

- Some of the co-requisite products, such as IBM File Manager, also require STEPLIB or LINKLIST/LPALIB definitions. See the related product customization guides for more information.

LINKLIST data sets are defined in SYS1.PARMLIB(PROGxx) by default. 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, of these things:

- Using STEPLIB in z/OS UNIX has a negative performance impact.
- If one STEPLIB library is APF-authorized, then all the other STEPLIB libraries 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.

## LINKLIST definitions for other products

The Developer for System z client has a code generation component called Enterprise Service Tools. In order for the generated code to issue diagnostic error messages, all IRZM\* and IIRZ\* modules in the FEK.SFEKLMOD load library must be made available either through STEPLIB or LINKLIST.

LINKLIST data sets are defined in SYS1.PARMLIB(PROGxx) by default.

If you opt to use STEPLIB, you must define the libraries that are not available through LINKLIST in the STEPLIB directive of the task that executes the code (IMS or batch job). However, if one STEPLIB library is APF-authorized, then all other STEPLIB libraries must be authorized. Libraries lose their APF authorization when they are mixed with non-authorized libraries in STEPLIB.

---

## PROCLIB changes

The following PROCLIB changes are documented in this section:

- “JMON, JES Job Monitor started task”
- “DBGMR, Debug manager started task” on page 24
- “RSED, RSE daemon started task” on page 25
- “ELAXF\* remote build procedures” on page 27

Additional information is available in the following sub-sections:

- “JCL limitations for the PARM variable” on page 26
- “TMPDIR processing” on page 27

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

### JMON, JES Job Monitor started task

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

- The high-level qualifier of the (authorized) 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..SFEKAUTH
//ENVIRON DD DISP=SHR,DSN=&CFG
//SYSPRINT DD SYSOUT=*
//SYSOUT  DD SYSOUT=*
//        PEND
//

```

Figure 1. JMON: JES Job Monitor started task

**Note:**

- For more information about the startup parameters, see Chapter 11, “Operator commands,” on page 177.
- The sample JCL is initially named FEK.SFEKSAMP(FEJJJCL) and is renamed to FEK.#CUST.PROCLIB(JMON) in “Customization setup” on page 15.
- Tracing can also be controlled by console commands, as described in Chapter 11, “Operator commands,” on page 177.
- For the recommended Workload Manager (WLM) goals for this task, see “WLM considerations” in the *Host Configuration Reference* (SC14-7290).

## DBGMGR, Debug manager started task

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

- The time-zone offset, default EST5DST
- The port used for external (client-host) communication, default 5335
- The port used for internal (host-confined) communication, default 5336
- The SVC number used for debugging read-only CICS transactions, default 251
- The high-level qualifier of the load library, default FEK

```

/*
/* RDz Debug Manager
/*
//DBGMGR  PROC PRM=, * PRM=DEBUG TO START TRACING
//          LEPRM='RPTOPTS(ON)',
//          TZ='EST5EDT',
//          CLIENT=5335,
//          HOST=5336,
//          SVC=251,
//          HLQ=FEK
/*
//DBGMGR  EXEC PGM=AQEZPCM,REGION=0M,TIME=NOLIMIT,
//          PARM=('&LEPRM ENVAR("TZ=&TZ")/&HOST &CLIENT &SVC &PRM')
//STEPLIB DD DISP=SHR,DSN=&HLQ..SFEKAUTH
//SYSPRINT DD SYSOUT=*
//SYSOUT  DD SYSOUT=*
//          PEND
/*

```

Figure 2. DBGMGR: Debug Manager started task

**Note:**

- This is an optional started task. It is used by the Integrated Debugger feature of Developer for System z. For more information, see “(Optional) Integrated debugger” on page 120.
- The sample JCL is initially named FEK.SFEKSAMP(AQESTC) and is renamed to FEK.#CUST.PROCLIB(DBGMGR) in “Customization setup” on page 15.
- For the recommended Workload Manager (WLM) goals for this task, see “WLM considerations” in the *Host Configuration Reference* (SC14-7290).

## RSED, RSE daemon started task

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

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

```

/*
/* RSE DAEMON
/*
//RSED    PROC IVP=, * 'IVP' to do an IVP test
//          PORT=,
//          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'
//STDOUT  DD SYSOUT=*
//STDERR  DD SYSOUT=*
//          PEND
/*

```

Figure 3. RSED: RSE daemon started task

**Note:**

- For more information about the startup parameters, see Chapter 11, “Operator commands,” on page 177.
- The sample JCL is initially named FEK.SFEKSAMP(FEKRSED) and is renamed to FEK.#CUST.PROCLIB(RSED) in “Customization setup” on page 15.

- Limit the length of the job name to 7 characters or fewer. If an 8-character name is used, the **modify** and **stop** operator commands fail with message "IEE342I MODIFY REJECTED-TASK BUSY". This behavior is caused by the z/OS UNIX design for child processes.
- For the recommended Workload Manager goals for this task and the child processes it creates, see "WLM considerations" in the *Host Configuration Reference* (SC14-7290). The child processes have the same name as the parent task, RSED, appended with a random 1-digit number, for example RSED8.

## JCL limitations for the PARM variable

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

- Use default values.

The `rsed.sh` startup script can be started without arguments, in which case the default argument values are used.

- Use symbolic links.

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

```
ln -s /long/directory/name/usr/lpp/rdz /usr/lpp/rdz
```

- Use STDIN.

When the PARM field is empty, **BPXBATCH** starts a z/OS UNIX shell and executes the shell script that is provided by STDIN. STDIN must be a z/OS UNIX file allocated as ORDONLY. Using STDIN disables the using of PROC variables such as TMPDIR. The shell executes the `/etc/profile` and `$HOME/.profile` shell logon scripts.

To use this method, first update the startup JCL to match something similar to the following sample:

```
//*
//* RSE DAEMON - USING STDIN
//*
//RSED    PROC CNFG='/etc/rdz'
//*
//RSE     EXEC PGM=BPXBATCH,REGION=0M,TIME=NOLIMIT
//STDOUT  DD SYSOUT=*
//STDERR  DD SYSOUT=*
//STDIN   DD PATHOPTS=(ORDONLY),PATH='&CNFG./rsed.stdin.sh'
//       PEND
//*
```

Figure 4. RSED: Alternate RSE daemon startup

Then, create the shell script (`/etc/rdz/rsed.stdin.sh` in this example) that will start the RSE daemon. You can edit the file with the TSO **OEDIT** command. The content of this script looks like the following sample:

```
CNFG=/etc/rdz
PORT=
IVP=
/long/directory/name/usr/lpp/rdz/bin/rsed.sh $IVP -C$CNFG -P$PORT -T$TMPDIR
```

Figure 5. `rsed.stdin.sh`: Alternate RSE daemon startup

**Note:** When using this method, the RSE daemon itself is not be active in the RSED address space but in a RSEDx address space. This is because z/OS UNIX runs

child processes (such as starting a shell) in separate address spaces. Adding an STDENV DD with a `_BPX_SHAREAS=YES` directive does not change this behavior because the directive is interpreted too late. This side effect severely complicates the using of Developer for System z operator commands.

## TMPDIR processing

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

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

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

```

/**
/** RSE DAEMON
/**
//RSED      PROC IVP=,                * 'IVP' to do an IVP test
//          PORT=,
//          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'
//STDOUT    DD SYSOUT=*
//STDERR    DD SYSOUT=*
//STDENV    DD PATHOPTS=(ORDONLY),PATH='&CNFG./rsed.stdenv'
//          PEND
/**

```

Figure 6. *RSED: alternate TMPDIR processing*

Then create the file (`/etc/rdz/rsed.stdenv` in this example) that will hold the `TMPDIR` definition. You can edit the file with the TSO `OEDIT` command. The content of this file looks like the following sample:

```
TMPDIR=/tmp
```

Figure 7. *rsed.stdenv: alternate TMPDIR processing*

Note that even though `rsed.envvars` has a `TMPDIR` variable, which will be used as soon as the started task is able to interpret `rsed.envvars`, you must *not* link `rsed.envvars` to DD `STDENV`, because it will cause startup failure.

## 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, 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 8 on page 28.

Table 8. 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 COBOL compiling and doing Integrated CICS translate and integrated DB2 translate.
ELAXFCOP	Sample procedure for DB2 preprocessing of EXEC SQL statements embedded in COBOL programs.
ELAXFCOT	Sample procedure for CICS translation for EXEC CICS statements embedded in COBOL programs.
ELAXFCPC	Sample procedure for C compiling.
ELAXFCPP	Sample procedure for C++ compiling.
ELAXFCP1	Sample procedure for COBOL compiling 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 DB2 preprocessing of EXEC SQL statements embedded in PLI programs.
ELAXFPLT	Sample procedure for doing CICS translation of EXEC CICS statements embedded in PLI programs.
ELAXFPL1	Sample procedure for PL/I compiling, and integrated CICS translation and integrated DB2 translation.
ELAXFPP1	Sample procedure for PL/I compiling with SCM preprocessor statements (-INC and ++INCLUDE).
ELAXFSP	Sample procedure to register a stored procedure to DB2.
ELAXFSQL	Sample procedure to invoke SQL.
ELAXFTSO	Sample procedure for running and debugging the generated DB2 code in TSO mode.
ELAXFUOP	Sample procedure for generating the UOPT step when building programs that run in CICS or IMS subsystems.

The names of the procedures and the names of the steps in the procedures match the default properties that are included with the Developer for System z client. If the name of a procedure or the name of a step in a procedure is changed, the corresponding properties file on all of the clients must be updated. You should not change the procedure and step names.

Customize the sample build procedure members, FEK.#CUST.PROCLIB(ELAXF\*), as described within the members, and copy them to SYS1.PROCLIB. Provide the correct high-level qualifiers for different product libraries, as described in Table 9 on page 29.

Table 9. ELAXF\* high-level qualifier checklist

Product	Default HLQ	Value
Developer for System z	FEK	
CICS	CICSTS42.CICS	
DB2	DSNA10	
IMS	IMS	
COBOL	IGY.V4R2M0	
PL/I	PLI.V4R2M0	
C/C++	CBC	
LE	CEE	
system LINKLIB	SYS1	
system MACLIB	SYS1	

Some ELAXF\* procedures reference data set names that do not have fixed low-level qualifiers. An example is the DB2 run library, which holds DB2 utilities that are compiled by your DB2 administrator. Use Table 10 to map the default data set names to the names used at your site.

Table 10. ELAXF\*. fully qualified data set checklist

Product	Default DSN	Value
Developer for System z - SQL samples	FEK.#CUST.SQL	
DB2 run libraries	DSNA10.RUNLIB.LOAD	

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

To create the security definitions for Developer for System z, customize and submit the sample FEKRACF member. The user submitting this job must have security administrator privileges, such as being RACF SPECIAL.

FEKRACF is located in FEK.#CUST.JCL, unless you specified a different location when you customized and submitted the FEK.SFEKSAMP(FEKSETUP) job. For more details, see "Customization setup" on page 15.

### Note:

- For those sites that use CA ACF2™ for z/OS, see the product page on the CA support site (<https://support.ca.com>) and check for the related Developer for System z Knowledge Document, TEC492389. This Knowledge Document has details on the security commands that are necessary to properly configure Developer for System z.
- For those sites that use CA Top Secret® for z/OS, see the product page on the CA support site (<https://support.ca.com>) and check for the related Developer

for System z Knowledge Document, TEC492091. This Knowledge Document has details on the security commands necessary to properly configure Developer for System z.

The following list of security-related definitions for Developer for System z are discussed in detail in Chapter 9, “Security definitions,” on page 143.

- Activate security settings and classes
- Define an OMVS segment for Developer for System z users
- Define data set profiles
- Define the JMON and RSED started tasks
- Define JES command security
- Define RSE as a secure z/OS UNIX server
- Define MVS program controlled libraries for RSE
- Define application security for RSE
- Define PassTicket support for RSE
- Define z/OS UNIX program controlled files for RSE

**Note:** 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.

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

---

## FEJJCENFG, the 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 the FEK.SFEKSAMP (FEKSETUP) job. For more details, see “Customization setup” on page 15.

Customize the sample JES Job Monitor configuration member FEJJCENFG, as shown in the following sample. Comment lines start with a number sign (#) when using a US code page. Data lines can have only a directive and its assigned value. Comments are not allowed on the same line.

**Note:** For your changes to take effect, the JMON started task must be restarted.

```

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

```

Figure 8. FEJJCNFG, JES Job Monitor configuration file

### SERV\_PORT

The port number for JES Job Monitor. The default port is 6715. The port can be changed if needed.

#### Note:

- 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. To learn how to define the variable for RSE, see “`rsed.envvars`, the RSE configuration file” on page 35.
- Before selecting a port, verify that the port is available on your system by using the **NETSTAT** and **NETSTAT PORTL** TSO commands.

**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 value to represent your time zone. Additional information can be found in the *UNIX System Services Command Reference* (SA22-7802).

The following definitions are optional. If omitted, default values are used as specified below:

### APPLID

Specifies the application identifier used for identifying JES Job Monitor to your security software. The default is FEKAPPL. Uncomment and change to the required application ID.

**Note:** This value must match the application ID set for RSE in the `rsed.envvars` configuration file. If these values differ, RSE cannot connect the client to JES Job Monitor. To learn how to define the variable for RSE, see “`rsed.envvars`, the RSE configuration file” on page 35.

### AUTHMETHOD

The default is SAF, which means that the System Authorization Facility (SAF) security interface is used. Do not change unless directed to do so by the IBM support center.

## CODEPAGE

The workstation code page. The default is UTF-8. The workstation code page is set to UTF-8 and generally should not be changed. If you have difficulty with multilingual characters, such as the currency symbol, you might need to uncomment the directive and change UTF-8 to match the workstation's code page.

## CONCHAR

Specifies the JES console command character. CONCHAR defaults to CONCHAR=\$ for JES2, or CONCHAR=\* for JES3. Uncomment and change to the requested command character.

## CONSOLE\_NAME

Specifies the name of the EMCS console used for issuing commands against jobs (Hold, Release, Cancel, and Purge). The default is JMON. Uncomment and change to the required console name, using the following guidelines.

- CONSOLE\_NAME must be either a console name consisting of 2 to 8 alphanumeric characters, or '&SYSUID' (without quotes).
- If a console name is specified, a single console by that name is used for all users. If the console by that name is already in use, the command issued by the client will fail.
- If &SYSUID is specified, the client user ID is used as the console name. Thus, a different console is used for each user. If the console by that name is already in use (for example, the user is using the SDSF ULOG), the command issued by the client might fail, depending on the GEN\_CONSOLE\_NAME setting.

No matter which console name is used, the user ID of the client that is requesting the command is used as the LU of the console, leaving a trace in syslog messages IEA630I and IEA631.

```
IEA630I OPERATOR console NOW ACTIVE, SYSTEM=sysid, LU=id
IEA631I OPERATOR console NOW INACTIVE, SYSTEM=sysid, LU=id
```

## GEN\_CONSOLE\_NAME

Enables or disables automatic generating of alternative console names. The default is OFF. To enable alternative console names, uncomment and change to ON.

This directive is used only when CONSOLE\_NAME equals &SYSUID and the user ID is not available as console name.

If GEN\_CONSOLE\_NAME=ON, an alternative console name is generated by appending a single numeric digit to the user ID. The digits 0 through 9 are attempted. If no available console is found, the command issued by the client fails.

If GEN\_CONSOLE\_NAME=OFF, the command issued by the client fails.

**Note:** The only valid settings are ON and OFF.

## HOST\_CODEPAGE

The host system codepage. The default is IBM-1047. Uncomment and change to match your host system codepage.

Note that this codepage is not used for data interpretation, only for server operations and client connection setup. The Developer for System z client provides the codepage to be used for data interpretation (which is retrieved from the properties of the "MVS Files" subsystem).

## LIMIT\_COMMANDS

Defines against which jobs the user can issue selected JES commands (Show JCL, Hold, Release, Cancel, and Purge). The default (LIMIT\_COMMANDS=USERID) limits the commands to the jobs owned by the user. To allow the user to issue commands against all spool files, if permitted by your security product, uncomment this directive and specify LIMITED or NOLIMIT.

Table 11. LIMIT\_COMMANDS command permission matrix

LIMIT_COMMANDS	Job owner	
	User	Other
USERID (default)	Allowed	Not allowed
LIMITED	Allowed	Allowed only if explicitly permitted by security profiles
NOLIMIT	Allowed	Allowed if permitted by security profiles or when the JESSPOOL class is not active

**Note:** The only valid settings are USERID, LIMITED, and NOLIMIT.

## LIMIT\_CONSOLE

Defines how much authority is granted to the console that is used to execute supported JES commands (Hold, Release, Cancel, and Purge). The default (LIMIT\_CONSOLE=LIMITED) limits the authority to commands protected by a security profile in the OPERCMDS class. To allow execution of supported JES commands that are not protected by a security profile, uncomment this directive and specify NOLIMIT.

When a security profile exists for a command, the user must have sufficient permission to execute the command, regardless of the LIMIT\_CONSOLE setting. The only valid settings are LIMITED and NOLIMIT.

## LIMIT\_VIEW

Defines what output the user can view. The default (LIMIT\_VIEW=NOLIMIT) allows the user to view all JES output, if permitted by your security product. To limit the view to output owned by the user, uncomment this directive and specify USERID.

**Note:** The only valid settings are USERID and NOLIMIT.

## LISTEN\_QUEUE\_LENGTH

The TCP/IP listen queue length. The default is 5. Do not change unless directed to do so by the IBM support center.

## LOOPBACK\_ONLY

Defines whether JES Job Monitor binds to the loopback address only, or to every available TCP/IP stack. Binding to loopback is more secure, because only tasks local to this z/OS system will be able to contact JES Job Monitor. The default is ON. Uncomment this directive and specify OFF if you want JES Job Monitor to bind to all TCP/IP stacks.

## MAX\_DATASETS

The maximum number of spooled output data sets that JES Job Monitor will return to the client (for example, SYSOUT, SYSPRINT, SYS00001, and so on). The default is 32. The maximum value is 2147483647.

### **MAX\_THREADS**

Maximum number of users that can be using one JES Job Monitor at a time. The default is 200. The maximum value is 2147483647. Increasing this number might require you to increase the size of the JES Job Monitor address space.

### **TIMEOUT**

The length of time, in seconds, before a thread is killed due to lack of interaction with the client. The default is 3600 (1 hour). The maximum value is 2147483647. TIMEOUT=0 disables the function.

### **TIMEOUT\_INTERVAL**

The number of seconds between timeout checks. The default is 1200. The maximum value is 2147483647.

### **TRACE\_STORAGE**

Enable storage tracing. The default is OFF. The only valid values are ON and OFF. Use only when directed by the IBM support center. To write a storage report to DD SYSOUT after each command, uncomment this directive and specify ON.

### **SEARCHALL**

Collect APPC and z/OS UNIX output that matches the JES Job Monitor filter, for example output written to SYSOUT by a Developer for System z CARMA server started using the CRASTART method. The default is OFF. The only valid values are ON and OFF. To collect the additional spool files, uncomment this directive and specify ON.

### **SUBMIT\_TIMEOUT**

The number of seconds that Developer for System z will wait for the completion of the TSO\_TEMPLATE job. The default is 30. The maximum value is 2147483647. Note: SUBMIT\_TIMEOUT has no effect unless SUBMITMETHOD=TSO is also specified.

### **SUBMITMETHOD=TSO**

Submit jobs through TSO. The default (SUBMITMETHOD=JES) submits jobs directly into JES. To submit the job through TSO **SUBMIT** command, uncomment this directive and specify TSO. This method allows TSO exits to be called; however, this method has a performance drawback.

#### **Note:**

- The only valid settings are TSO and JES.
- If SUBMITMETHOD=TSO is specified, TSO\_TEMPLATE must also be defined.

### **TSO\_TEMPLATE**

Wrapper JCL for submitting the job through TSO. The default value is FEK.#CUST.CNTL(FEJTSO). This statement refers to the fully qualified member name of the JCL to be used as a wrapper for the TSO **SUBMIT** command. For more information, see the SUBMITMETHOD statement.

#### **Note:**

- A sample wrapper job is provided in FEK.#CUST.CNTL(FEJTSO). See this member for more information about the customization needed.
- TSO\_TEMPLATE has no effect unless SUBMITMETHOD=TSO is also specified.

---

## rzed.envvars, the RSE configuration file

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

Remote Systems Explorer (RSE) provides core services such as connecting the client to the host system and starting other servers for specific services.

`rzed.envvars` is located in `/etc/rdz/`, unless you specified a different location when you customized and submitted the `FEK.SFEKSAMP(FEKSETUP)` job. For more details, see “Customization setup” on page 15. You can edit the file with the TSO **OEDIT** command.

See the following sample `rzed.envvars` file, which must be customized to match your system environment. Comment lines start with a number sign (#) when using a US code page. Data lines can have only a directive and its assigned value; comments are not allowed on the same line. Line continuations and spaces around the equal sign (=) are not supported.

**Note:** For your changes to take effect, the RSED started task must be restarted.

```

#####
# (1) required definitions
JAVA_HOME=/usr/lpp/java/J6.0
RSE_HOME=/usr/lpp/rdz
_RSE_RSED_PORT=4035
_RSE_JMON_PORT=6715
RSE_HLQ=FEK
_RSE_HOST_CODEPAGE=IBM-1047
TZ=EST5EDT
LANG=C
PATH=/bin:/usr/sbin
_CEE_DMPTARG=/tmp
STEPLIB=NONE
#STEPLIB=$STEPLIB:CEE.SCEERUN:CEE.SCEERUN2:CBC.SCLBDLL
_RSE_JAVAOPTS=""
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Xms128m -Xmx512m"
_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="
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dlog.retention.period=5"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dmaximum.clients=30"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dmaximum.threads=520"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dminimum.threadpool.process=1"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dmaximum.threadpool.process=100"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dipv6=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Ddisplay.users=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dkeep.all.logs=false"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dkeep.last.log=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Denable.standard.log=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Denable.port.of.entry=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Denable.certificate.mapping=false"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Denable.automount=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Denable.audit.log=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Daudit.cycle=30"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Daudit.retention.period=0"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Daudit.log.mode=RW.R."
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Daudit.action=<user_exit>"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Daudit.action.id=<userid>"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dlogon.action=<user_exit>"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dlogon.action.id=<userid>"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Ddeny.nonzero.port=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dsingle.logon=false"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dprocess.cleanup.interval=0"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dreject.logon.threshold=1000000"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dinclude.c=/etc/rdz/include.conf"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dinclude.cpp=/etc/rdz/include.conf"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DCPP_CLEANUP_INTERVAL=60000"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DRIS_BUFFER=8"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DAPPLID=FEKAPPL"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DRSE_DSICALL=TSO"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dsearch.server.limit.hits=0"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dsearch.server.limit.datasets=0"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dsearch.server.limit.lines=0"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dsearch.server.limit.timeout=0"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDISABLE_TEXT_SEARCH=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDENY_PASSWORD_SAVE=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DHIDE_ZOS_UNIX=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDISABLE_REMOTE_INDEX_SEARCH=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDISABLE_DELETE_IN_SUBPROJECT=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDSTORE_IDLE_SHUTDOWN_TIMEOUT=3600000"

```

Figure 9. rsed.envvars: RSE configuration file

```

# RSE_JAVAOPTS="$ RSE_JAVAOPTS -DDSTORE_SSL_ALGORITHM=TLSv1.2"
# RSE_JAVAOPTS="$ RSE_JAVAOPTS -DDSTORE_TCP_NO_DELAY=true"
# RSE_JAVAOPTS="$ RSE_JAVAOPTS -DDSTORE_TRACING_ON=true"
# RSE_JAVAOPTS="$ RSE_JAVAOPTS -DDSTORE_MEMLOGGING_ON=true"
#=====
# (2) required definitions for TSO/ISPF Client Gateway
CGI_ISPHOME=/usr/lpp/ispf
CGI_ISPCONF=/etc/rdz
CGI_ISPWORK=/var/rdz
#STEPLIB=$STEPLIB:ISP.SISPLOAD:ISP.SISPLPA:SYS1.LINKLIB
_RSE_ISPF_OPTS=""
# RSE_ISPF_OPTS="$ RSE_ISPF_OPTS&ISPPROF=&SYSUID..ISPPROF"
#CGI_ISPPREF="$ RSE_ISPF_OPTS&ISPPREF..ISPF.VCMISPF"
#=====
# (3) required definitions for SCLM Developer Toolkit
_SCLMDT_CONF_HOME=/var/rdz/scldmt
#STEPLIB=$STEPLIB:$RSE_HLQ.SFEKAUTH:$RSE_HLQ.SFEKLOAD
# SCLMDT_TRANTABLE=FEK.#CUST.LSTRANS.FILE
#ANT_HOME=/usr/lpp/Apache/Ant/apache-ant-1.7.1
#=====
# (4) optional definitions
# RSE_PORTRANGE=8108-8118
# BPXK_SETIBMOPT_TRANSPORT=TCPIP
#TMPDIR=/tmp
# RSE_FEK_SAF_CLASS=FACILITY
# RSE_LDAP_SERVER=ldap_server_url
# RSE_LDAP_PORT=389
# RSE_LDAP_PTC_GROUP_SUFFIX="o=PTC,c=DeveloperForZ"
#GSK_CRL_SECURITY_LEVEL=HIGH
#GSK_LDAP_SERVER=ldap_server_url
#GSK_LDAP_PORT=ldap_server_port
#GSK_LDAP_USER=ldap_userid
#GSK_LDAP_PASSWORD=ldap_server_password
#STEPLIB=$RSE_HLQ.SFEKLOAD:SYS1.CSSLIB:SYS1.SIXMLOD1
#RSE_UBLD_DD=$CGI_ISPCONF/ISPF.conf
#RSE_UBLD_STEPLIB=$STEPLIB
#=====
# (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_DSN_SFEKLOAD=$RSE_HLQ.SFEKLOAD
RSE_LIB=$RSE_HOME/lib
PATH=.:$JAVA_HOME/bin:$RSE_HOME/bin:$CGI_ISPHOME/bin:$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/mvsutil.jar:$RSE_LIB/jesutils.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/lucene-core-2.3.2.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/cdtparser.jar:$RSE_LIB/wdzBidi.jar
CLASSPATH=$CLASSPATH:$ RSE_SAF_CLASS
CLASSPATH=.:$CLASSPATH
_RSE_PTC=$ RSE_LDAP_PTC_GROUP_SUFFIX
_RSE_ISPF_OPTS="$ RSE_ISPF_OPTS &SESSION=SPAWN$ RSE_ISPF_OPTS"
_RSE_JAVAOPTS="$ RSE_JAVAOPTS -Dldap.server.address=$ RSE_LDAP_SERVER"
_RSE_JAVAOPTS="$ RSE_JAVAOPTS -Dldap.server.port=$ RSE_LDAP_PORT"
_RSE_JAVAOPTS="$ RSE_JAVAOPTS -Dldap.ptc.group.name.suffix=$ RSE_PTC"
_RSE_JAVAOPTS="$ RSE_JAVAOPTS -DISPF_OPTS='$ RSE_ISPF_OPTS'"
_RSE_JAVAOPTS="$ RSE_JAVAOPTS -DA_PLUGIN_PATH=$RSE_LIB"

```

Figure 10. rsed.envvars: RSE configuration file (continued)

```

_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_INITIAL_SIZE=0"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDSTORE_MAX_FREE=0"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDSTORE_SPIRIT_ON=false"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DSPIRIT_EXPIRY_TIME=90"
_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.info.timeout=10000"
_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_SERVER_TIMEOUT=120000
_SCLMDT_BASE_HOME=$RSE_HOME
_SCLMDT_WORK_HOME=$CGI_ISPHOME
CGI_DTWORK=$ _SCLMDT_WORK_HOME
_CMDSERV_BASE_HOME=$CGI_ISPHOME
_CMDSERV_CONF_HOME=$CGI_ISPCONF
_CMDSERV_WORK_HOME=$CGI_ISPWORK
#####
# (6) additional environment variables

```

Figure 11. *rсед.envvars*: RSE configuration file (continued)

**Note:** Symbolic links are allowed when specifying values and directories in *rсед.envvars*, as long as the symbols are defined in *rсед.envvars*.

The following definitions are required:

**JAVA\_HOME**

Java home directory. The default is `/usr/lpp/java/J6.0`. 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 needed.

**Note:**

- Before selecting a port, verify that the port is available on your system by using the TSO commands `NETSTAT` and `NETSTAT PORTL`.
- This port is used for client-host communication.
- The RSED started task can override the port number specified here.

**\_RSE\_JMON\_PORT**

JES Job Monitor port number. The default is 6715. Can be changed if needed.

**Note:**

- This value must match the port number set for JES Job Monitor in the FEJJC�FG configuration file. If these values differ, RSE cannot connect the client to JES Job Monitor. To learn how to define the variable for JES Job Monitor, see “FEJJC�FG, the JES Job Monitor configuration file” on page 30.
- Before selecting a port, verify that the port is available on your system by using the TSO commands **NETSTAT** and **NETSTAT PORTL**.
- All communication on this port is confined to your z/OS host system.

#### **RSE\_HLQ**

The high-level qualifier used to install Developer for System z. The default is FEK. Change to match the location of your Developer for System z data sets.

#### **\_RSE\_HOST\_CODEPAGE**

The host system codepage. The default is IBM-1047. Change to match your host system codepage. Note that this codepage is not used for data interpretation, only for server operations and client connection setup. The Developer for System z client provides the codepage to be used for data interpretation (which is retrieved from the properties of the "MVS Files" subsystem).

**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.

Additional information can be found in the *UNIX System Services Command Reference* (SA22-7802).

#### **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.

**PATH** Command path. The default is /bin:/usr/sbin:.. Can be changed if needed.

#### **\_CEE\_DMPTARG**

Language Environment (LE) z/OS UNIX dump location used by the Java Virtual Machine (JVM). The default is /tmp.

#### **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. For more information about the usage of the libraries in the following list, see “PARMLIB changes” on page 16:

```
# RSE
STEPLIB=$STEPLIB:CEE.SCEERUN:CEE.SCEERUN2:CBC.SCLBDLL
# ISPF
STEPLIB=$STEPLIB:ISP.SISPLOAD:ISP.SISPPLA:SYS1.LINKLIB
# SCLM Developer Toolkit
STEPLIB=$STEPLIB:$RSE_HLQ.SFEKAUTH:$RSE_HLQ.SFEKLOAD
# zUnit, xUnit support for Enterprise COBOL and PL/I
STEPLIB=$STEPLIB:$RSE_HLQ.SFEKLOAD:SYS1.CSSLIB:SYS1.SIXML0D1
```

#### **Note:**

- Using STEPLIB in z/OS UNIX has a negative performance impact.

- If one STEPLIB library is APF-authorized, then all the other STEPLIB libraries 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 and 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**

Additional RSE-specific Java options. For more information about this definition, see “Defining extra Java startup parameters with \_RSE\_JAVAOPTS” on page 45.

The following definitions are required if ISPFs TSO/ISPF Client Gateway is used for the TSO Commands service or SCLM Developer Toolkit.

#### **CGI\_ISPHOME**

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. This directive is required only when ISPFs TSO/ISPF Client Gateway is used.

#### **CGI\_ISPCONF**

ISPF base configuration directory. The default is `/etc/rdz`. Change to match the location of `ISPF.conf`, the TSO/ISPF Client Gateway customization file. This directive is required only when ISPFs TSO/ISPF Client Gateway is used.

#### **CGI\_ISPWORK**

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. This directive is required only when ISPFs TSO/ISPF Client Gateway is used.

#### **Note:**

- The TSO/ISPF Client Gateway adds `/WORKAREA` to the path specified in `CGI_ISPWORK`. Do not add it yourself.
- If you did not use the `SFEKSAMP(FEKSETUP)` sample job to build the customizable environment, verify that the `WORKAREA` directory exists in the path specified in `CGI_ISPWORK`. The directory permission bits must be `777`.

#### **STEPLIB**

STEPLIB is described previously in the required definitions section.

#### **\_RSE\_ISPF\_OPTS**

Additional TSO/ISPF Client Gateway-specific Java options. The default is `"`. For more information about this definition, see “Defining the extra Java startup parameters with \_RSE\_ISPF\_OPTS” on page 52. This directive is required only when ISPFs TSO/ISPF Client Gateway is used.

#### **CGI\_ISPPREF**

High-level qualifier for the temporary data set created by the TSO/ISPF Client Gateway. The default is `"&SYSPREF..ISPF.VCMISPF"`. Uncomment and change to match your data set naming conventions. This directive is required only when ISPF's TSO/ISPF Client Gateway is used.

The following variables can be used in the data set name:

- &SYSUID. to substitute the developer's user ID
- &SYSPREF. to substitute the developer's TSO prefix or, if the TSO prefix cannot be determined, the user ID
- &SYSNAME. to substitute the system name as specified in the IEASYMxx parmlib member

**Note:** This directive requires ISPF APAR OA38740.

The following definitions are required if SCLM Developer Toolkit is used.

#### **\_SCLMDT\_CONF\_HOME**

SCLM Developer Toolkit base configuration directory. The default is /var/rdz/sc1mdt. Change to match the location of the CONFIG directory used by SCLMDT to store SCLM project information. This directive is required only when SCLMDT is used.

**Note:** SCLMDT adds /CONFIG and /CONFIG/PROJECT to the path specified in SCLMDT\_CNF\_HOME. Do not add it yourself.

#### **STEPLIB**

STEPLIB is described previously in the required definitions section.

#### **\_SCLMDT\_TRANTABLE**

Name of the long/short name translation VSAM. The default is FEK.#CUST.LSTRANS.FILE. Uncomment and change to match the name used in the ISP.SISPSAMP(FLM02LST) SCLM sample job. This directive is required only if the long/short name translation in SCLM Developer Toolkit is used.

#### **ANT\_HOME**

Home directory for your Ant installation. The default is /usr/lpp/apache/Ant/apache-ant-1.7.1. Change to match your Ant installation. This directive is required only when the Java EE build support is used with SCLM Developer Toolkit.

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

#### **\_RSE\_PORTRANGE**

Specifies the port range that the RSE server can open for communication with a client. Any port can be used by default. For more information about this definition, see "Defining the PORTRANGE available for RSE server" on page 44. This is an optional directive.

#### **\_BPXK\_SETIBMOPT\_TRANSPORT**

Specifies the name of the TCP/IP stack to be used. The default is TCPIP. Uncomment and change to the requested TCP/IP stack name, as defined in the TCPIPJOBNAME statement in the related TCPIP.DATA. This is an optional directive.

#### **Note:**

- Coding a SYSTCPD DD statement in the server JCL does not set the requested stack affinity.
- When this directive is not active, RSE binds to every available stack on the system (BIND INADDRANY).

#### **TMPDIR**

Specifies the path used to store temporary files. The default is /tmp. Uncomment and change to use the requested path. This is an optional directive.

#### **\_RSE\_FEK\_SAF\_CLASS**

Specifies the security class where FEK.\* profiles are defined. The default is FACILITY. To enforce the usage of the specified value, uncomment and change. This is an optional directive.

#### **\_RSE\_LDAP\_SERVER**

Specifies the LDAP server host name used by the push-to-client function. The default is the current z/OS host name. To enforce the usage of the specified value, uncomment and change. This is an optional directive.

#### **\_RSE\_LDAP\_PORT**

Specifies the LDAP server port used by the push-to-client function. The default is 389. To enforce the usage of the specified value, uncomment and change. This is an optional directive.

#### **\_RSE\_LDAP\_PTC\_GROUP\_SUFFIX**

Specifies the "O=<organization>,C=<country>" suffix needed to find the push-to-client groups within the LDAP server. The default is "O=PTC,C=DeveloperForZ". To enforce the usage of the specified value, uncomment and change. This is an optional directive.

#### **GSK\_CRL\_SECURITY\_LEVEL**

Specifies the level of security SSL applications use when contacting LDAP servers to check CRLs for revoked certificates during certificate validation. The default is MEDIUM. To enforce the usage of the specified value, uncomment and change. This is an optional directive. The following values are valid:

- LOW: Certificate validation does not fail if the LDAP server cannot be contacted.
- MEDIUM: Certificate validation requires the LDAP server to be contactable, but does not require a CRL to be defined. This value is the default.
- HIGH: Certificate validation requires the LDAP server to be contactable and a CRL to be defined.

**Note:** This directive requires z/OS 1.9 or later.

#### **GSK\_LDAP\_SERVER**

Specifies one or more blank-separated LDAP server host names. To enforce the usage of the specified LDAP servers to obtain their CRL, uncomment and change. This is an optional directive.

The host name can either be a TCP/IP address or a URL. Each host name can contain an optional port number separated from the host name by a colon sign (:).

#### **GSK\_LDAP\_PORT**

Specifies the LDAP server port. The default is 389. To enforce the usage of the specified value, uncomment and change. This is an optional directive.

#### **GSK\_LDAP\_USER**

Specifies the distinguished name to use when connecting to the LDAP server. To enforce the usage of the specified value, uncomment and change. This is an optional directive.

#### **GSK\_LDAP\_PASSWORD**

Specifies the password to use when connecting to the LDAP server. To enforce the usage of the specified value, uncomment and change. This is an optional directive.

| **RSE\_UBLD\_DD**

| Specifies the DD statements that will be used when generating JCL for IBM  
| Rational Team Concert™ user builds from a Developer for System z client  
| that invoke TSO or ISPF commands. By default, Developer for System z  
| uses the definitions in ISPF.conf, which is referenced by CGI\_ISPCONF in  
| rsed.envvars. Uncomment and change to use the DD definitions in the  
| specified file, which must follow the syntax rules specified in *ISPF.conf*, the  
| *ISPF's TSO/ISPF Client Gateway configuration file*. This is an optional  
| directive.

| **RSE\_UBLD\_STEPLIB**

| Specifies the STEPLIB statement that will be used when generating JCL for  
| IBM Rational Team Concert user builds from a Developer for System z  
| client that invoke TSO or ISPF commands. By default, Developer for  
| System z uses the STEPLIB definition in rsed.envvars. Uncomment and  
| change to use the specified STEPLIB definition. This is an optional directive.

The following definitions are required, and should not be changed unless directed by the IBM support center:

**\_CEE\_RUNOPTS**

Language Environment (LE) runtime options. The default is "ALL31(0N) HEAP(32M,32K,ANYWHERE,KEEP,,) TRAP(0N)". Do not modify.

**\_BPX\_SHAREAS**

Run foreground processes in the same address space as the shell. The default is YES. Do not modify.

**\_BPX\_SPAWN\_SCRIPT**

Run shell scripts directly from the spawn() function. The default is YES. Do not modify.

**\_EDC\_ADD\_ERRNO2**

Show the reason code in z/OS UNIX error messages. The default is 1. Do not modify.

**JAVA\_PROPAGATE**

Propagates the security and workload context during thread creation (Java version 1.4 and older only). The default is NO. Do not modify.

**RSE\_DSN\_SFEKLOAD**

Fully qualified data set name of the SFEKLOAD load library. The default is \$RSE\_HLQ.SFEKLOAD. Do not modify.

**RSE\_LIB**

RSE library path. The default is \$RSE\_HOME/lib. Do not modify.

**PATH**

Command path. The default is .:\$JAVA\_HOME/bin:\$RSE\_HOME/bin:\$CGI\_ISPHOME/bin:\$PATH. Do not modify.

**LIBPATH**

Library path. The default is too long to repeat. Do not modify.

**CLASSPATH**

Class path. The default is too long to repeat. Do not modify.

**\_RSE\_ISPF\_OPTS**

Additional TSO Commands service-specific Java options. The default is "&SESSION=SPAWN\$\_RSE\_ISPF\_OPTS". Do not modify.

#### **\_RSE\_JAVAOPTS**

Additional RSE-specific Java options. The default is too long to repeat. Do not modify.

#### **\_RSE\_SERVER\_CLASS**

Java class for the RSE server. The default is `org.eclipse.dstore.core.server.Server`. Do not modify.

#### **\_RSE\_DAEMON\_CLASS**

Java class for the RSE daemon. The default is `com.ibm.etools.zos.server.RseDaemon`. Do not modify.

#### **\_RSE\_POOL\_SERVER\_CLASS**

Java class for the RSE thread pool. The default is `com.ibm.etools.zos.server.ThreadPoolProcess`. Do not modify.

#### **\_RSE\_SERVER\_TIMEOUT**

Time out value for the RSE server (waiting on the client) in milliseconds. The default is 120000 (2 minutes). Do not modify.

#### **SCLMDT\_BASE\_HOME**

Home directory for SCLM Developer Toolkit code. The default is `$RSE_HOME`. Do not modify.

#### **SCLMDT\_WORK\_HOME**

SCLM Developer Toolkit base work directory. The default is `$CGI_ISPHOME`. Do not modify.

#### **CGI\_DTWORK**

SCLM Developer Toolkit support for older clients. The default is `$_SCLMDT_WORK_HOME`. Do not modify.

#### **\_CMDSERV\_BASE\_HOME**

ISPF TSO/ISPF Client Gateway service support. The default is `$CGI_ISPHOME`. Do not modify.

#### **\_CMDSERV\_CONF\_HOME**

ISPF TSO/ISPF Client Gateway service support. The default is `$CGI_ISPCONF`. Do not modify.

#### **\_CMDSERV\_WORK\_HOME**

ISPF TSO/ISPF Client Gateway service support. The default is `$CGI_ISPWORK`. Do not modify.

## **Defining the PORTRANGE available for RSE server**

This is a part of `rsed.envvars` customization that specifies the ports on which the RSE server can communicate with the client. This range of ports has no connection with the RSE daemon port.

To help understand the port usage, a brief description of RSE's connection process follows:

1. The client connects to host system port 4035, RSE daemon.
2. The RSE daemon creates an RSE server thread.
3. The RSE server opens a host system port for the client to connect. The selection of this port can be configured by using the `_RSE_PORTRANGE` definition in `rsed.envvars`.
4. The RSE daemon returns the port number to the client.
5. The client connects to the host system port.

**Note:**

- The process is similar for the optional alternative connection method using REXEC/SSH.
- For more information, see "Understanding Developer for System z" in the *Host Configuration Reference* (SC14-7290).

To specify the port range, for the client to communicate with z/OS, uncomment and customize the following line in `rsed.envvars`:

```
#_RSE_PORTRANGE=8108-8118
```

**Note:** Before selecting a port range, verify that the range is available on your system by using the **NETSTAT** and **NETSTAT PORTL** commands.

The format of `PORTRANGE` is: `_RSE_PORTRANGE=min-max`. The value for `max` is non-inclusive; for example, the expression `_RSE_PORTRANGE=8108-8118` means port numbers from 8108 up to 8117 are usable. The port number used by the RSE server is determined in the following order:

1. If a nonzero port number is specified in the subsystem properties on the client, the specified port number is used. If the port is not available, the connection fails. This setup is not recommended.

**Note:** The host system can deny this type of connection request by specifying the `deny.nonzero.port=true` directive in `rsed.envvars`. For more information about this directive, see "Defining extra Java startup parameters with `_RSE_JAVAOPTS`."

2. If the port number in the subsystem properties is 0 and if `_RSE_PORTRANGE` is specified in `rsed.envvars`, the port range specified by `_RSE_PORTRANGE` is used. If no port in the range is available, the connection fails.

RSE server does not need the port exclusively for the duration of the client connection. It is only in the time span between the server bind and the client connect that no other RSE server can bind to the port. This means that most connections use the first port in the range, with the rest of the range being a buffer in case of multiple simultaneous logons.

3. If the port number in the subsystem properties is 0 and `_RSE_PORTRANGE` is not specified in `rsed.envvars`, any available port is used.

## Defining extra Java startup parameters with `_RSE_JAVAOPTS`

With the different `_RSE_*OPTS` directives, `rsed.envvars` provides the facility to give extra parameters to Java when it starts the RSE processes. The sample options included in `rsed.envvars` can be activated by uncommenting them.

`_RSE_JAVAOPTS` defines standard and RSE-specific Java options.

```
_RSE_JAVAOPTS=""
```

Variable initialization. Do not modify.

```
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Xms128m -Xmx512m"
```

Set initial (`Xms`) and maximum (`Xmx`) heap size. The defaults are 128M and 512M respectively. Change to enforce the required heap size values. If this directive is commented out, the Java default values are used, which are 4M and 512M (1M and 64M for Java 5.0).

**Note:** To determine the optimal values for this directive, see "Key resource definitions" in the *Host Configuration Reference* (SC14-7290).

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

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

**Note:** If this directive or its counterpart, the home directory, does not specify an absolute path (where the path does not start with a forward slash (/)), the actual log location is relative to the configuration directory, which, by default is `/etc/rdz`.

**`_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 required location. If this directive is commented out or the value is a null string, the home directory of the client user ID is used. The home directory is defined in the OMVS security segment of the user ID.

**Note:**

- If this directive or its counterpart, the home directory, does not specify an absolute path (the path does not start with a forward slash (/)), the actual log location is relative to the configuration directory, which, by default is `/etc/rdz`.
- 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 clients user ID in uppercase.
- Ensure that the permission bits for `userlog/dstorelog` are set so that each client can create `$LOGNAME`.

**`_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DDSTORE_LOG_DIRECTORY="`**

This directory is appended to the path specified in the `user.log` directive. Together they create the path leading to the user-specific logs. The default is a null-string. Change to enforce the usage of the specified directory. If this directive is commented out, `.eclipse/RSE/` is used.

**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 clients user ID in uppercase.
- The directory specified here is relative to the directory specified in `user.log` and might, therefore, not start with a forward slash (/).
- Ensure that the permission bits for `userlog/dstorelog` are set so that each client can create `$LOGNAME`.

**`_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dlog.retention.period=5"`**

Number of days daemon and user logs are kept. The default is 5. Customize this directive to delete the logs after a given number of days. Specify 0 to set no limit. The maximum value is 365. Note that daemon log cleanup happens at the next action that requires daemon activity. User logs are cleaned up the next time the user connects.

The following directives are commented out by default.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Dmaximum.clients=30"**

Maximum amount of clients serviced by one thread pool. The default is 30. To limit the number of clients per thread pool, uncomment and customize. Other limits might prevent RSE from reaching this limit.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Dmaximum.threads=520"**

Maximum amount of active threads in one thread pool to allow new clients. The default is 520. To limit the number of clients in each thread pool, based on the number of threads in use, uncomment and customize. Each client connection uses multiple threads (17 or more) and other limits might prevent RSE from reaching this limit.

**Note:** This value must be lower than the setting for MAXTHREADS and MAXTHREADTASKS in SYS1.PARMLIB(BPXPRMxx).

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Dminimum.threadpool.process=1"**

The minimum number of active thread pools. The default is 1. To start at least the listed number of thread pool processes, uncomment and customize. Thread pool processes are used for load balancing the RSE server threads. More new processes are started when they are needed. Starting the new processes upfront helps prevent connection delays but uses more resources during idle times.

**Note:** If the single.logon directive is active, at least 2 thread pools are started, even if minimum.threadpool.process is set to 1. The default setting for single.logon in rsed.envvars is active.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Dmaximum.threadpool.process=100"**

The maximum number of active thread pools. The default is 100. To limit the number of thread pool processes, uncomment and customize. Thread pool processes are used for load balancing the RSE server threads, so limiting them will limit the amount of active client connections.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Dipv6=true"**

TCP/IP version. The default is false, which means that an IPv4 interface is used. To use an IPv6 interface, uncomment and specify true.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Ddisplay.users=true"**

Automated display of active users. The default is false. Uncomment and specify true to enable an automated display of active users in rserver.log on each user logon and logoff.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Dkeep.all.logs=false"**

Use file names with an embedded time-stamp for daemon and user logs. The default is true, which implies that the logs are kept until removed by the log.retention.period setting. Uncomment and specify false to use fixed log file names, which are replaced each time the daemon is started or the user connects.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Dkeep.last.log=true"**

Keep a copy of the host log files belonging to the previous session. The default is false. To rename the previous log files to \*.last during server startup and client connect, uncomment and specify true. Note that the .dstore\* user trace files are not removed automatically upon client reconnect, nor are they part of the keep.last.log processing. Removing these files is a manual action. The keep.all.logs directive must be set to false for keep.last.log to take effect.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Denable.standard.log=true"**

Write the stdout and stderr streams of the thread pools to a log file. The

default is false. To save the stdout and stderr streams, uncomment and specify true. The resulting log files are located in the directory referenced by the `daemon.log` directive.

**Note:**

- The **MODIFY RSESTANDARDLOG** operator command can be used to dynamically stop or start the update of the stream log files.
- There are no user-specific `stdout.log` and `stderr.log` log files when the `enable.standard.log` directive is active. The user-specific data is now written to the matching RSE thread pool stream.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Denable.port.of.entry=true"**

Port Of Entry (POE) check option. The default is false. To enforce POE checking for client connections, uncomment and specify true. During POE checking, the IP address of the client is mapped into a network access security zone by your security software. The client user ID must have permission to use the profile that defines the security zone.

**Note:**

- POE checking must also be enabled in your security product.
- Enabling POE checking enables the product for other z/OS UNIX services also, such as INETD.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Denable.certificate.mapping=false"**

Use your security software to authenticate a logon with a X.509 certificate. The default is true. To have RSE daemon do the authentication without relying on the X.509 support of your security software, uncomment and specify false.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Denable.automount=true"**

Support home directories created by z/OS UNIX automount. The default is false. To ensure that z/OS UNIX automount uses the client user ID as owner of the directory, uncomment and specify true.

**Note:** z/OS UNIX automount uses the user ID of the process that called the service when creating a file system. If this option is disabled, this process is the RSE thread pool server, with the STCRSE user ID. If this option is enabled, a new, temporary process is created by using the client user ID before invoking the service.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Denable.audit.log=true"**

Audit option. The default is false. To enforce audit logging of actions done by clients, uncomment and specify true. Audit logs are written to the RSE daemon log location. To know the location, see the `daemon.log` option of the `_RSE_JAVAOPTS` variable.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Daudit.cycle=30"**

Number of days stored in 1 audit log file. The default is 30. To control how much audit data is written to 1 audit log file, uncomment and customize. The maximum value is 365.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Daudit.retention.period=0"**

Number of days audit logs are kept. The default is 0, which means no limit is specified. To delete audit logs after a given number of days, uncomment and customize. The maximum value is 365.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Daudit.log.mode=RW.R."**

Access permission mask for audit logs. The default is `RW.R.`, which allows

the owner read and write access. The owner's default group has read access and everyone else has no access. To set the required access permissions, uncomment and customize.

UNIX standards dictate that permissions can be set for three types of users: owner, group, and other. The fields in the `audit.log.mode` mask match this order, and the fields are separated by a period (.). Each field can be empty, or have R, W, or RW as values, where R = read and W = write.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Daudit.action=<user exit>"**

Name of a user exit which is called when an audit log file is closed. There is no default value, but a sample exit is provided in `/usr/lpp/rdz/samples/process_audit.rex`. To enable post-processing of audit logs, uncomment and specify the full pathname of the user exit program.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Daudit.action.id=<userid>"**

User ID to be used for running the exit specified in the `audit.action` variable. The default is the user ID assigned to RSE daemon. To use the specified ID for executing the audit post-processing exit, uncomment and specify a user ID.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Dlogon.action=<user\_exit>"**

Name of a user exit which is called when a user logs on. There is no default value, but a sample exit is provided in `/usr/lpp/rdz/samples/process_logon.sh`. To enable post-processing of a logon, uncomment and specify the full pathname of the user exit program.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Dlogon.action.id=<userid>"**

User ID to be used for running the exit specified in the `logon.action` variable. The default is the user ID assigned to RSE daemon. To use the specified ID for executing the logon post-processing exit, uncomment and specify a user ID.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Ddeny.nonzero.port=true"**

Disallow the client to choose the communication port number. The default is false. To refuse connections where the client specifies which host system port must be used by RSE server for the connection, uncomment and specify true. For more information, see "Defining the PORTRANGE available for RSE server" on page 44.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Dsingle.logon=false"**

Disallow a user ID to log on multiple times. The default is true. To allow a user ID to log on multiple times to a single RSE daemon, uncomment and specify false.

**Note:**

- A second logon attempt causes the first one to be canceled by the host system if this directive is not active or set to true. This cancellation action is accompanied by console message FEK210I.
- If the `single.logon` directive is active, at least 2 thread pools are started, even if `minimum.threadpool.process` is set to 1. The default setting for `minimum.threadpool.process` in `rsed.envvars` is 1.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Dprocess.cleanup.interval=0"**

Automatically remove RSE thread pools that are in an unrecoverable error state. By default, erroneous RSE thread pools are not automatically removed. To automatically remove erroneous RSE thread pool servers at every interval, where the interval unit is seconds, uncomment and customize. Specifying 0 does not start an interval timer, but erroneous RSE

thread pool servers are removed when the RSE daemon checks the RSE thread pools during a new client logon or the DISPLAY PROCESS command.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Dreject.logon.threshold=1000000"**

A thread pool opening a file larger than the specified size will not accept new logon requests until the file is loaded. The default file size is 1000000 bytes. To specify the file size at which a thread pool is to ignore logon requests when such a file is opened, uncomment and customize. Other thread pools can still accept new logon requests.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Dinclude.c=/etc/rdz/include.conf"**

This variable points to a fully qualified z/OS UNIX file containing a list of forced includes for content assist on C code. A forced include consists of a file or directory, data set, or data set member which is parsed when a content assist operation is performed, regardless of whether that file or member was included in the source code using a pre-processor directive. To specify the name of the configuration file, uncomment and customize.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Dinclude.cpp=/etc/rdz/include.conf"**

This variable points to a fully qualified z/OS UNIX file containing a list of forced includes for content assist on C++ code. A forced include consists of a file or directory, data set, or data set member which is parsed when a content assist operation is performed, regardless of whether that file or member was included in the source code using a pre-processor directive. To specify the name of the configuration file, uncomment and customize.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -DCPP\_CLEANUP\_INTERVAL=60000"**

Cleanup interval for unused C/C++ header files in milliseconds. The default is 60000, which means 1 minute. To change the cleanup interval, uncomment and customize. Specifying a value of 0 prevents caching of C/C++ header files, thereby reducing performance of remote content assist in the editor.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -DRIS\_BUFFER=8"**

Buffer size, in megabytes, used during remote index creation. The default is 8 MB. To change the buffer size, uncomment and customize. Valid values are whole numbers between 1 and 2000 (both inclusive). A bigger buffer speeds up index creation, but uses a bigger portion of the thread pool's Java heap. The buffer is automatically flushed to the index if it is full before index creation ends.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -DAPPLID=FEKAPPL"**

RSE server application ID. The default is FEKAPPL. To enforce the use of the required application ID, uncomment and customize.

**Note:**

- The application ID must be defined to your security software. Failure to do so prevents the client from logging on.
- For the security implications when changing this value, see "Using PassTickets" in *Host Configuration Reference* (SC14-7290).
- This value must match the application ID set for JES Job Monitor in the FEJJCNFG configuration file. If these values differ, RSE cannot connect the client to JES Job Monitor. To learn how to define the variable for JES Job Monitor, see "FEJJCNFG, the JES Job Monitor configuration file" on page 30.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -DRSE\_DSICALL=TSO"**

Method used to collect data set information. By default, the service uses

the same TSO/ISPF Client Gateway infrastructure that is used for TSO commands issued by the user. To use an alternative, less resource-intensive, method to collect data set information, uncomment and specify TSO. This alternative method needs additional prerequisites:

- FEK.SFEKLOAD(FEKDSI) must be in LINKLIST or STEPLIB, that is, defined in rsed.envvars.
- The REXX runtime library (REXX.\*.SEAGLPA) must be in LPA, LINKLIST, or STEPLIB, that is, defined in rsed.envvars).

This variable is not taken into account when APPC is used instead of ISPF's TSO/ISPF Client Gateway to support the TSO Commands service.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Dsearch.server.limit.hits=0"**

Limit the resource usage of non-indexed file and text searches. The default is 0 (no limit). Uncomment and customize this directive to stop a search after the specified number of results have been found.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Dsearch.server.limit.datasets=0"**

Limit the resource usage of non-indexed file and text searches. The default is 0 (no limit). Uncomment and customize this directive to stop a search after the specified number of data sets has been scanned.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Dsearch.server.limit.lines=0"**

Limit the resource usage of non-indexed file and text searches. The default is 0 (no limit). Uncomment and customize this directive to stop a search after the specified number of lines has been scanned.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -DDISABLE\_TEXT\_SEARCH=true"**

Disable non-indexed text searches. The default is false. To prevent users from starting a full text search on the host, uncomment and specify true.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS  
-DDISABLE\_REMOTE\_INDEX\_SEARCH=true"**

Disable the Remote Index Search menu item on the client. The default is false. To prevent users from creating remote indexes for host system data sets, uncomment and specify true. This option works only with clients version 8.5.1 and later.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -DDENY\_PASSWORD\_SAVE=true"**

Password save option. The default is false. To prevent users from saving their host password on the client, uncomment and specify true. Previously saved passwords are removed. This option works only with clients version 7.1 and later.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -DHIDE\_ZOS\_UNIX=true"**

Hide z/OS UNIX option. The default is false. To prevent users from seeing z/OS UNIX elements, that is the directory structure and command line, on the client, uncomment and specify true. This option works only with clients version 7.6 and later.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS  
-DDISABLE\_DELETE\_IN\_SUBPROJECT=true"**

Disable the Delete menu item in the context menu of z/OS subprojects. The default is false. To prevent users from using the Delete menu item in the context menu of z/OS subprojects, uncomment and specify true. This option works only with clients version 8.0.1 and later.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS  
-DDSTORE\_IDLE\_SHUTDOWN\_TIMEOUT=3600000"**

Disconnect idle clients. By default, idle clients are not disconnected. To

disconnect clients who are idle for the listed number of milliseconds, uncomment and customize. 3600000 milliseconds equal 1 hour.

```
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS  
-DDSTORE_SSL_ALGORITHM=TLSv1.2"
```

Use TLS instead of SSL for encrypted communication with the client. The default is SSL. To start using TLS (v1.2) for Developer for System z client-host communication, uncomment and specify TLSv1.2. This option works only with Java version 7.0 and later and clients version 9.0 and later. Note that by default, the version 9.0 client also uses SSL. On the client, -DDSTORE\_SSL\_ALGORITHM=TLSv1.2 must be specified in eclipse.ini to enable TLS encrypted communication.

```
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DDSTORE_TCP_NO_DELAY=true"
```

Disable the TCP/IP DELAY ACK function. The default is false. To stop TCP/IP from doing DELAY ACK for Developer for System z client-host communication, uncomment and specify true.

```
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DDSTORE_TRACING_ON=true"
```

Start dstore tracing. Use only when directed by the IBM support center. The resulting .dstoreTrace log file is created in Unicode (ASCII), not EBCDIC.

```
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DDSTORE_MEMLOGGING_ON=true"
```

Start dstore memory tracing. Use only when directed by the IBM support center. The resulting .dstoreMemLogging log file is created in Unicode (ASCII), not EBCDIC.

## Defining the extra Java startup parameters with \_RSE\_ISPF\_OPTS

With the different `_RSE_*OPTS` directives, `rsed.envvars` provides the facility to give extra parameters to Java when it starts the RSE processes. The sample options included in `rsed.envvars` can be activated by uncommenting them.

The `_RSE_ISPF_OPTS` directives are RSE-specific Java options and are, by default, in effect only when ISPF's TSO/ISPF Client Gateway is used by the Developer for System z.

```
_RSE_ISPF_OPTS=""
```

Variable initialization. Do not modify.

```
_RSE_ISPF_OPTS="$_RSE_ISPF_OPTS&ISPPROF=&SYSUID..ISPPROF"
```

Use an existing ISPF profile for the ISPF initialization. To use the specified ISPF profile, uncomment and change the data set name.

The following variables can be used in the data set name:

- `&SYSUID.` to substitute the developer's user ID
- `&SYSPREF.` to substitute the developer's TSO prefix, or user ID if the TSO prefix cannot be determined
- `&SYSNAME.` to substitute the system name as specified in the IEASYMxx parmlib member

---

## ISPF.conf, the 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. These services include the TSO Commands service and SCLM Developer Toolkit.

ISPF.conf is located in /etc/rdz/, unless you specified a different location when you customized and submitted job FEK.SFEKSAMP(FEKSETUP). For more details, see "Customization setup" on page 15. You can edit the file with the TSO **OEDIT** command.

Definitions must start in column 1. Comment lines start with an asterisk (\*) when using a US code page. Data lines can have only a directive and its assigned value. Comments are not allowed on the same line. Line continuations are not supported. When concatenating data set names, add them on the same line and separate the names with a comma (,).

In addition to providing the correct names for the ISPF data sets, also add the TSO Commands service data set name, FEK.SFEKPROC, to the SYSPROC or SYSEXEC statement, as shown in the following example.

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

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

Figure 12. 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. For more details, see "Customizing the TSO environment" in the *Host Configuration Reference* (SC14-7290).
- You cannot define a STEPLIB directive. Use the STEPLIB directive in rsed.envvars instead.
- The TSO/ISPF Client Gateway might not function properly if you use a third-party product that intercepts ISPF commands, such as **ISPSTART**. To see how it can be disabled for Developer for System z, check the documentation for that product. If the product requires the allocation of a specific DD statement to DUMMY, you can simulate this behavior in ISPF.conf by allocating that DD statement to nullfile.

For example:

```
ISPTRACE=nullfile
```

- When using the allocjob directive, be careful not to undo the DD definitions done earlier in ISPF.conf.
- System abend 522 for module ISPZTS0 is to be expected if the JWT parameter in the SMFPRMxx parmlib member is set lower than the ISPF\_timeout value in ISPF.conf. This does not impact Developer for System z operations because the TSO/ISPF Client Gateway is restarted automatically when needed.
- Changes are active for all new invocations. No server restart is needed.

---

## Optional components

This section combines a variety of optional customization tasks. Follow the instructions in the appropriate section to configure the required service.

Customizations to Developer for Developer for System z stand-alone components:

- Chapter 3, “(Optional) Common Access Repository Manager (CARMA),” on page 57
- Chapter 4, “(Optional) SCLM Developer Toolkit,” on page 89
- Chapter 5, “(Optional) Application Deployment Manager (deprecated),” on page 97
- Chapter 6, “(Optional) Host-based code analysis,” on page 105

Customizations to Developer for System z configuration files:

- “(Optional) pushtoclient.properties, the host-based client control” on page 107
- “(Optional) ssl.properties, the RSE SSL encryption” on page 110
- “(Optional) rsecomm.properties, the RSE tracing” on page 112
- “(Optional) include.conf, Forced includes for C/C++ content assist” on page 113

Developer for System z related customizations to or for other products:

- “(Optional) z/OS UNIX subprojects” on page 115
- “(Optional) Include preprocessor support” on page 115
- “(Optional) xUnit support for Enterprise COBOL and PL/I” on page 116
- “(Optional) Enterprise Service Tools support” on page 117
- “(Optional) CICS bidirectional language support” on page 118
- “(Optional) Diagnostic IRZ messages for the generated code” on page 118
- “(Optional) Problem Determination Tools support” on page 122
- “(Optional) DB2 and IMS debug support” on page 123
- “(Optional) File Manager support” on page 123
- “(Optional) WORKAREA and /tmp cleanup” on page 124

---

## Installation verification

The detailed description of the various installation verification programs (IVPs) is located in Chapter 8, “Installation verification,” on page 127 because some of the IVPs are for the optional components.

You can test the basic functions with the following scenario:

1. Start the JMON started task or user job. The startup information in DD STDOUT should end with the following message:  
FEJ211I Server ready to accept connections.  
If the job ends with return code 66, FEK.SFEKAUTH is not APF-authorized.
2. 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. Check DD STDOUT for messages indicating that the following IVPs were successful:
  - Java startup
  - JES Job Monitor connection
  - TCP/IP setup

3. Start the RSED started task or user job without the IVP parameter. RSE daemon issues the following console message upon successful startup:

```
FEK002I RseDaemon started. (port=4035)
```

4. Issue the following operator commands and verify in the resulting console messages that the tests ran successfully:

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

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

```
F RSED,APPL=IVP ISPF,userid
```

Replace userid with a valid TSO user ID.



---

## Chapter 3. (Optional) Common Access Repository Manager (CARMA)

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 and source code examples for creating your own RAM.

SCMs that are based on host systems need single-user address spaces to access their services, which requires CARMA to start a CARMA server for each user. It is not possible to create a single server supporting multiple users.

---

### Requirements and checklist

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

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

To start using CARMA at your site, do the following tasks. Unless otherwise indicated, all tasks are mandatory.

1. Choose a method to start CARMA and choose which RAMs should be activated. Several combinations of RAMs and server startup methods are available as a preconfigured setup. For details, see “Select the server startup method and active RAM” on page 58.
2. Create CARMA VSAM data sets. For details, see “CARMA VSAM data sets” on page 75 and “CARMA Repository Access Managers (RAMs)” on page 77.
3. Initial customization of the RSE configuration files to interface with CARMA. The complete customization is dependent on the method chosen to start CARMA. For details, see “CRASRV.properties, the RSE interface to CARMA” on page 68.
4. Depending on the chosen CARMA startup method and the chosen RAMs, do the required customization of the related configuration files. For details see:
  - “crastart\*.conf, the CRASTART server startup” on page 71
  - “CRASUB\*, the batch submit server startup” on page 74
5. Optionally, customize the CA Endeavor<sup>®</sup> SCM-specific configuration members. For details see “CRACFG, CRASHOW and CRATMAP, the CA Endeavor<sup>®</sup> SCM RAM configuration files” on page 79 and “CA Endeavor<sup>®</sup> SCM RAM batch actions” on page 81.

6. Optionally, update the data set allocation exec. For details, see “CRANDVRA, the CA Endeavor® SCM RAM allocation exec” on page 80, “CRAALLOC, the custom RAM allocation exec” on page 83, and “(Optional) Custom allocation exec” on page 86.
7. Optionally, create a startup user exit. For details, see (Optional) CARMA user exit.
8. Optionally, create CRAXJCL as replacement for IRXJCL. For details, see “(Optional) IRXJCL versus CRAXJCL” on page 87.

**Note:** The sample members referenced in this chapter are located in FEK.#CUST.\* and /etc/rdz, unless you specified a different location when you customized and submitted the FEK.SFEKSAMP(FEKSETUP) job. For more details, see “Customization setup” on page 15.

---

## Select the server startup method and active RAM

Developer for System z supports multiple methods to start a CARMA server. Developer for System z also provides multiple Repository Access Managers (RAMs), which can be divided into two groups, production RAMs and sample RAMs. This publication describes several possible combinations of RAMs and server startup methods. Each of the described configuration scenarios is available as a preconfigured setup.

### CARMA server startup

Developer for System z supports multiple methods to start a CARMA server. Each method has benefits and drawbacks.

#### **CRASTART**

The "CRASTART" method starts the CARMA server as a subtask within RSE. This method provides a very flexible setup by using a separate configuration file that defines data set allocations and program invocations that are needed to start a CARMA server. This method provides the best performance and uses the fewest resources, but requires that the CRASTART module be located in LPA.

#### **Batch submit**

The "batch submit" method starts the CARMA server by submitting a job. This is the default method that is used in the provided sample configuration files. The benefit of this method is that the CARMA logs are easily accessible in the job output. It also allows the use of custom server JCL for each developer, which is maintained by the developer himself. However, this method uses one JES initiator for each developer who starts a CARMA server.

#### **(Deprecated) TSO/ISPF Client Gateway**

The "TSO/ISPF Client Gateway" method uses ISPF's TSO/ISPF Client Gateway to create a TSO or ISPF environment, in which the CARMA server is started. This method allows for flexible data set allocations by using the possibilities of ISPF.conf. However, this method is not suited to access SCMs that interfere with normal TSO or ISPF operations.

**Note:** The "TSO/ISPF Client Gateway" connection method has been marked as deprecated. Although it is still supported, this function will no longer be enhanced, and the documentation has moved to a white paper, *Using ISPF Client Gateway to provide CARMA services* (SC14-7292), available in the Developer for System z library, <http://www-01.ibm.com/software/awdtools/rdz/library/>.

## Production RAMs

Production type RAMs are fully functional, pre-built RAMs that can be used to access an SCM in a production environment.

### CA Endeavor<sup>®</sup> SCM RAM

The IBM Rational Developer for System z Interface for CA Endeavor<sup>®</sup> Software Configuration Manager gives Developer for System z clients direct access to CA Endeavor<sup>®</sup> SCM. From here on, IBM Rational Developer for System z Interface for CA Endeavor<sup>®</sup> SCM is abbreviated to CA Endeavor<sup>®</sup> SCM RAM.

### CA Endeavor<sup>®</sup> SCM packages RAM

The CA Endeavor<sup>®</sup> SCM packages RAM gives Developer for System z clients direct access to CA Endeavor<sup>®</sup> SCM packages.

## Sample RAMs

Sample RAMs are provided for the purpose of testing the configuration of your CARMA environment and as examples for developing your own RAMs. Source code is included.

**Attention:** Do not use the provided sample RAMs in a production environment.

### PDS RAM

The PDS RAM gives a data set list similar to **MVS Files -> My Data Sets** in the Remote Systems view.

### Skeleton RAM

The skeleton RAM gives a functional framework that can be used as starting point to develop your own RAM.

### SCLM RAM

The SCLM RAM gives a basic entry into SCLM, ISPF's Software Configuration Manager. The SCLM RAM is not enabled by default.

## Preconfigured RAM and server startup combinations

Several combinations of RAMs and server startup methods are available as a preconfigured setup. The listed scenarios need only minor customization to fit your environment.

- "CRASTART with CA Endeavor<sup>®</sup> SCM RAM"
- "CRASTART with sample RAMs" on page 62
- "Batch submit with CA Endeavor<sup>®</sup> SCM RAM" on page 64
- "Batch submit with sample RAMs" on page 66

Detailed information on the different steps of each scenario can be found in "CARMA configuration details" on page 68.

It is possible to add a RAM to any CARMA setup, now or somewhere in the future. See "(Optional) Supporting multiple RAMs" on page 84 for more information on adding a RAM to an existing setup.

---

## CRASTART with CA Endeavor<sup>®</sup> SCM RAM

The information in this section describes how to set up CARMA with the following specifications:

- Server startup: CRASTART method. This method requires that CRASTART is in LPA.
- RAM: CA Endeavor<sup>®</sup> SCM RAM.

This customization step can be omitted if you want to use one of the other scenarios with different specifications.

## Create the CARMA VSAM data sets

To define and populate the CARMA-related VSAM data sets, customize and submit the following JCL jobs. For customization instructions, see the documentation within the member. Existing VSAM data sets are replaced.

For more details on this step, see “CARMA VSAM data sets” on page 75.

- FEK.#CUST.JCL(CRA\$VCAD)
- FEK.#CUST.JCL(CRA\$VCAS)
- FEK.#CUST.JCL(CRA\$VMSG)

## Customize CRASRV.properties

RSE server uses the settings in `/etc/rdz/CRASRV.properties` to start and connect to a CARMA server. You can edit the file with the TSO **OEDIT** command. For the changes to take effect, restart the RSED started task.

When you use the default file locations, the only required changes are changing the value of the `clist.dsname` directive to `*CRASTART` and changing the value of `crastart.configuration.file` to `/etc/rdz/crastart.endeavor.conf`. For more information about the different directives, see “CRASRV.properties, the RSE interface to CARMA” on page 68.

```
clist.dsname=*CRASTART
crastart.configuration.file=crastart.endeavor.conf
```

*Figure 13. CRASRV.properties: CRASTART with CA Endeavor<sup>®</sup> SCM RAM*

## Customize crastart.endeavor.conf

CRASTART uses the definitions in `/etc/rdz/crastart.endeavor.conf` to create a valid TSO/ISPF environment to start a CARMA server. You can edit the file with the TSO **OEDIT** command. Changes are in effect for all CARMA servers that are started after the update.

For customization instructions, see the documentation within the file. For more information about the CRASTART startup method, see “`crastart*.conf`, the CRASTART server startup” on page 71.

**Note:** Due to page width limitations, some lines in the following sample wrapped onto the next line. All lines that start with an indentation should be added to the end of the previous line.

```

* DD used by RAM
TYPEMAP = FEK.#CUST.PARMLIB(CRATMAP)
SHOWVIEW= FEK.#CUST.PARMLIB(CRASHOW)
CRACFG = FEK.#CUST.PARMLIB(CRACFG)
* uncomment CRABCFG and CRABSKEL to use batch actions
*CRABCFG = FEK.#CUST.PARMLIB(CRABCFG)
*CRABSKEL= FEK.#CUST.CNTL
CONLIB = CA.NDVR.CSIQLOAD
-COMMAND=ALLOC FI(JCLOUT) SYSOUT(A) WRITER(INTRDR) RECFM(F) LRECL(80)
  BLKSIZE(80)
-COMMAND=ALLOC FI(EXT1ELM) NEW DELETE DSORG(PS) RECFM(V,B) LRECL(4096)
  BLKSIZE(27998) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(EXT2ELM) NEW DELETE DSORG(PS) RECFM(V,B) LRECL(4096)
  BLKSIZE(27998) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(EXT1DEP) NEW DELETE DSORG(PS) RECFM(V,B) LRECL(4096)
  BLKSIZE(27998) SPACE(5,5) TRACKS UNIT(SYSALLDA)
C1EXMSGS= SYSOUT(H)
C1MSGSI = SYSOUT(H)
MSG3FILE= DUMMY

* DD used by CARMA server (CRASERV)
* pay attention to APF authorizations when using TASKLIB
TASKLIB = FEK.SFEKLOAD,CA.NDVR.CSIQAUTH,CA.NDVR.CSIQAUTU
CRADEF = FEK.#CUST.CRADEF
CRAMSG = FEK.#CUST.CRAMSG
CRASTRS = FEK.#CUST.CRASTRS
CARMALOG= SYSOUT(H)
SYSPRINT= SYSOUT(H)

* DD used by ISPF (via NDVRC1)
-COMMAND=ALLOC FI(ISPCTL0) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(ISPCTL1) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(ISPPROF) NEW DELETE DSORG(PO) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA) DIR(5)
ISPTABL = -ISPPROF
ISPTLIB = -ISPPROF,ISP.SISPTENU
ISPMLIB = ISP.SISPMENU
ISPPLIB = ISP.SISPPENU
ISPSLIB = ISP.SISPSENU

* DD used by TSO (IKJEFT01)
SYSPROC = FEK.SFEKPROC * CRANDVRA
SYSTSIN = DUMMY
SYSTSPRT= SYSOUT(H)

PROGRAM=IKJEFT01 %CRANDVRA NDVRC1 PGM(CRASERV) PARM(&CRAPRM1.
&CRAPRM2. &CRAPRM3. &CRAPRM4. &CRAPRM5. &CRAPRM6. &CRAPRM7.
&CRAPRM8. )

```

Figure 14. *crastart.endevor.conf*: CRASTART with CA Endeavor® SCM RAM

## (Optional) Additional CA Endeavor® SCM RAM customization

The CA Endeavor® SCM RAM has additional components that can be customized if needed.

- CARMA startup processing has an optional user exit. For more information see (Optional) CARMA user exit.
- The CA Endeavor® SCM RAM has multiple configuration files FEK.#CUST.PARMLIB(CRACFG), FEK.#CUST.PARMLIB(CRASHOW) and

FEK.#CUST.PARMLIB(CRATMAP), that can be customized. For more information, see “CRACFG, CRASHOW and CRATMAP, the CA Endeavor® SCM RAM configuration files” on page 79.

- The CA Endeavor® SCM RAM has an allocation exec, FEK.SFEKPROC(CRANDVRA), that can be customized. For more information, see “CRANDVRA, the CA Endeavor® SCM RAM allocation exec” on page 80.
- The CA Endeavor® SCM RAM supports doing CA Endeavor® SCM actions in batch mode. Batch-actions requires a configuration file, FEK.#CUST.PARMLIB(CRABCFG), and a skeleton JCL, FEK.#CUST.CNTL(CRABATCA), that must be customized. For more information, see “CA Endeavor® SCM RAM batch actions” on page 81.

---

## CRASTART with sample RAMs

The information in this section describes how to set up CARMA with the following specifications:

- Server startup: CRASTART method. This method requires that CRASTART is in LPA.
- RAM: sample RAMs, which are not to be used for production purposes.

This customization step can be bypassed if you want to use one of the other scenarios with different specifications.

## Create the CARMA VSAM data sets

Customize and submit the following JCL jobs to define and populate the CARMA-related VSAM data sets. For customization instructions, see the documentation within the member. Existing VSAM data sets are replaced.

For more details on this step, see “CARMA VSAM data sets” on page 75 and “CARMA Repository Access Managers (RAMs)” on page 77.

### CARMA

- FEK.#CUST.JCL(CRA\$VDEF)
- FEK.#CUST.JCL(CRA\$VMSG)
- FEK.#CUST.JCL(CRA\$VSTR)

### Sample RAMs

- FEK.#CUST.JCL(CRA#VPDS)

## Customize CRASRV.properties

RSE server uses the settings in /etc/rdz/CRASRV.properties to start and connect to a CARMA server. You can edit the file with the TSO **EDIT** command. For the changes to take effect, the RSED started task must be restarted.

When using the default file locations, the only required change is changing the value of the `clist.dsname` directive to `*CRASTART`. For more information about the different directives, see “CRASRV.properties, the RSE interface to CARMA” on page 68.

```

|          clist.dsname=*CRASTART
|          crastart.configuration.file=crastart.conf

```

Figure 15. CRASRV.properties: CRASTART with sample RAMs

## Customize crastart.conf

CRASTART uses the definitions in /etc/rdz/crastart.conf to create a valid TSO/ISPF environment to start a CARMA server. You can edit the file with the TSO **EDIT** command. Changes are in effect for all CARMA servers that are started after the update.

For customization instructions, see the documentation within the file. For more information about the CRASTART startup method, see “crastart\*.conf, the CRASTART server startup” on page 71.

```

|          * DD used by RAM
|          CRARAM1 = FEK.#CUST.CRARAM1                                * PDS RAM
|          * DD used by CARMA server (CRASERV)
|          TASKLIB = FEK.SFEKLOAD
|          CRADEF = FEK.#CUST.CRADEF
|          CRAMSG = FEK.#CUST.CRAMSG
|          CRASTRS = FEK.#CUST.CRASTRS
|          CARMALOG= SYSOUT(H)
|          SYSPRINT= SYSOUT(H)
|
|          * DD used by ISPF (ISPSTART)
|          -COMMAND=ALLOC FI(ISPCTL0) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(80)
|             BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA)
|          -COMMAND=ALLOC FI(ISPCTL1) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(80)
|             BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA)
|          -COMMAND=ALLOC FI(ISPPROF) NEW DELETE DSORG(PO) RECFM(F,B) LRECL(80)
|             BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA) DIR(5)
|          ISPTABL = -ISPPROF
|          ISPTLIB = -ISPPROF,ISP.SISPTENU
|          ISPMLIB = ISP.SISPMENU
|          ISPLLIB = ISP.SISPPENU
|          ISPSLIB = ISP.SISPSENU
|
|          * DD used by TSO (IKJEFT01)
|          SYSPROC = #hlq.SFEKPROC                                    * CRAALLOC
|          SYSTSIN = DUMMY
|          SYSTSPRT= SYSOUT(H)
|
|          PROGRAM=IKJEFT01 %CRAALLOC ISPSTART PGM(CRASERV) PARM(&CRAPRM1.
|             &CRAPRM2. &CRAPRM3. &CRAPRM4. &CRAPRM5. &CRAPRM6. &CRAPRM7.
|             &CRAPRM8. )

```

Figure 16. crastart.conf: CRASTART with sample RAMs

**Note:** Due to page width limitations, some lines in the sample wrapped onto the next line. All lines that start with an indentation should be added to the end of the previous line.

---

## (Optional) Additional custom RAM customization

The custom RAMs have additional components that can be customized if needed.

- CARMA startup processing has an optional user exit. For more information see (Optional) CARMA user exit.

- Custom RAM startup has an allocation exec, FEK.SFEKPROC(CRAALLOC), that can be customized. For more information, see CRAALLOC, the custom RAM allocation exec.

---

## Batch submit with CA Endeavor® SCM RAM

The information in this section describes how to set up CARMA with the following specifications:

- Server startup: batch submit method. This method requires JES initiators.
- RAM: CA Endeavor® SCM RAM.

This customization step can be omitted if you want to use one of the other scenarios with different specifications.

### Create the CARMA VSAM data sets

Customize and submit the following JCLs to define and populate the CARMA-related VSAM data sets. For customization instructions, see the documentation within the member. Existing VSAM data sets are replaced.

For more details on this step, see “CARMA VSAM data sets” on page 75.

- FEK.#CUST.JCL(CRA\$VCAD)
- FEK.#CUST.JCL(CRA\$VCAS)
- FEK.#CUST.JCL(CRA\$VMSG)

### Customize CRASRV.properties

RSE server uses the settings in `/etc/rdz/CRASRV.properties` to start and connect to a CARMA server. You can edit the file with the TSO **EDIT** command. For the changes to take effect, the RSED started task must be restarted.

When using default file locations, the only required change is changing the value of the `clist.dsname` directive to `FEK.#CUST.CNTL(CRASUBCA)`. For more information about the different directives, see “CRASRV.properties, the RSE interface to CARMA” on page 68.

```
clist.dsname='FEK.#CUST.CNTL(CRASUBCA)'
```

*Figure 17. CRASRV.properties: Batch submit with CA Endeavor® SCM RAM*

### Customize CRASUBCA

The `FEK.#CUST.CNTL(CRASUBCA)` CLIST and embedded JCL submits a CARMA server. Changes are in effect for all CARMA servers that are started after the update.

For customization instructions, see the documentation within the member. For more information about the batch submit startup method, see “CRASUB\*, the batch submit server startup” on page 74.

```

PROC 8 CRAPRM1 CRAPRM2 CRAPRM3 CRAPRM4 CRAPRM5 CRAPRM6 CRAPRM7 CRAPRM8
SUBMIT * END($$)
//CRA&PORT JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
//*
//RUN      EXEC PGM=IKJEFT01,DYNAMNBR=125,REGION=0M,TIME=NOLIMIT
//*
//* DD used by RAM
//TYPEMAP DD DISP=SHR,DSN=FEK.#CUST.PARMLIB(CRATMAP)
//SHOWVIEW DD DISP=SHR,DSN=FEK.#CUST.PARMLIB(CRASHOW)
//CRACFG DD DISP=SHR,DSN=FEK.#CUST.PARMLIB(CRACFG)
//* uncomment CRABCFG and CRABSKEL to use batch actions
//*CRABCFG DD DISP=SHR,DSN=FEK.#CUST.PARMLIB(CRABCFG)
//*CRABSKEL DD DISP=SHR,DSN=FEK.#CUST.CNTL
//CONLIB DD DISP=SHR,DSN=CA.NDVR.CSIQLOAD
//JCLOUT DD SYSOUT=(A,INTRDR),DCB=(LRECL=80,RECFM=F,BLKSIZE=80)
//EXT1ELM DD DISP=(NEW,DELETE),UNIT=SYSALLDA,
//          RECFM=VB,LRECL=4096,BLKSIZE=27998,SPACE=(TRK,(5,5))
//EXT2ELM DD DISP=(NEW,DELETE),UNIT=SYSALLDA,
//          RECFM=VB,LRECL=4096,BLKSIZE=27998,SPACE=(TRK,(5,5))
//EXT1DEP DD DISP=(NEW,DELETE),UNIT=SYSALLDA,
//          RECFM=VB,LRECL=4096,BLKSIZE=27998,SPACE=(TRK,(5,5))
//C1MSG1 DD SYSOUT(H)
//C1EXMSG1 DD SYSOUT(H)
//MSG3FILE DD DUMMY
//*
//* DD used by CARMA server (CRASERV)
//* pay attention to APF authorizations when using STEPLIB
//STEPLIB DD DISP=SHR,DSN=FEK.SFEKLOAD
//          DD DISP=SHR,DSN=CA.NDVR.CSIQAUTH
//          DD DISP=SHR,DSN=CA.NDVR.CSIQAUTU
//CRADEF DD DISP=SHR,DSN=FEK.#CUST.CRADEF
//CRAMSG DD DISP=SHR,DSN=FEK.#CUST.CRAMSG
//CRASTRS DD DISP=SHR,DSN=FEK.#CUST.CRASTRS
//CARMALOG DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//*
//* DD used by ISPF (via NDVRC1)
//ISPPROF DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          LRECL=80,RECFM=FB,SPACE=(TRK,(1,1,5))
//ISPCTL0 DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          LRECL=80,RECFM=FB,SPACE=(TRK,(5,5))
//ISPCTL1 DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          LRECL=80,RECFM=FB,SPACE=(TRK,(5,5))
//ISPMLIB DD DISP=SHR,DSN=ISP.SISPMENU
//ISPPLIB DD DISP=SHR,DSN=ISP.SISPPENU
//ISPSLIB DD DISP=SHR,DSN=ISP.SISPSENU
//ISPTLIB DD DISP=SHR,DSN=ISP.SISPTENU
//*
//* DD used by TSO (IKJEFT01)
//SYSPROC DD DISP=SHR,DSN=FEK.SFEKPROC * CRANDVRA
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
%CRANDVRA NDVRC1 PGM(CRASERV) PARM(&CRAPRM1 &CRAPRM2 &CRAPRM3 +
&CRAPRM4 &CRAPRM5 &CRAPRM6 &CRAPRM7 &CRAPRM8 )
$$
EXIT CODE(0)

```

Figure 18. CRASUBCA: Batch submit with CA Endeavor® SCM RAM

## (Optional) Additional CA Endeavor® SCM RAM customization

The CA Endeavor® SCM RAM has additional components that can be customized if needed.

- The CA Endeavor<sup>®</sup> SCM RAM has multiple configuration files, FEK.#CUST.PARMLIB(CRACFG), FEK.#CUST.PARMLIB(CRASHOW) and FEK.#CUST.PARMLIB(CRATMAP), that can be customized. For more information, see “CRACFG, CRASHOW and CRATMAP, the CA Endeavor<sup>®</sup> SCM RAM configuration files” on page 79.
- The CA Endeavor<sup>®</sup> SCM RAM has an allocation exec, FEK.SFEKPROC(CRANDVRA), that can be customized. For more information, see “CRANDVRA, the CA Endeavor<sup>®</sup> SCM RAM allocation exec” on page 80.
- The CA Endeavor<sup>®</sup> SCM RAM supports doing CA Endeavor<sup>®</sup> SCM actions in batch mode. Batch-actions requires a configuration file, FEK.#CUST.PARMLIB(CRABCFG), a skeleton JCL, FEK.#CUST.CNTL(CRABATCA), and an optional default job card, FEK.#CUST.CNTL(CRABJOB), that must be customized. For more information, see “CA Endeavor<sup>®</sup> SCM RAM batch actions” on page 81.

---

## Batch submit with sample RAMs

The information in this section describes how to set up CARMA with the following specifications:

- Server startup: batch submit method, which requires JES initiators
- RAM: sample RAMs, which are not to be used for production purposes

This customization step can be omitted if you want to use one of the other scenarios with different specifications.

## Create the VSAM data sets

Customize and submit the following JCL jobs to define and populate the CARMA-related VSAM data sets. For customization instructions, see the documentation within the member. Existing VSAM data sets are replaced.

For more details on this step, see “CARMA VSAM data sets” on page 75 and “CARMA Repository Access Managers (RAMs)” on page 77.

### CARMA

- FEK.#CUST.JCL(CRA\$VDEF)
- FEK.#CUST.JCL(CRA\$VMSG)
- FEK.#CUST.JCL(CRA\$VSTR)

### Sample RAMs

- FEK.#CUST.JCL(CRA#VPDS)

## Customize CRASRV.properties

RSE server uses the settings in /etc/rdz/CRASRV.properties to start and connect to a CARMA server. You can edit the file with the TSO **EDIT** command. For the changes to take effect, the RSED started task must be restarted.

Because this is the default scenario for Developer for System z, no changes are required when starting from an unmodified copy of the file. For more information about the different directives, see “CRASRV.properties, the RSE interface to CARMA” on page 68.

```
cllist.dsname='FEK.#CUST.CNTL(CRASUBMT)'
```

Figure 19. CRASRV.properties: Batch submit with sample RAMs

## Customize CRASUBMT

The FEK.#CUST.CNTL(CRASUBMT) CLIST and embedded JCL submits a CARMA server. Changes are in effect for all CARMA servers that are started after the update.

For customization instructions, see the documentation within the member. For more information about the batch submit startup method, see “CRASUB\*, the batch submit server startup” on page 74.

```
PROC 8 CRAPRM1 CRAPRM2 CRAPRM3 CRAPRM4 CRAPRM5 CRAPRM6 CRAPRM7 CRAPRM8
SUBMIT * END($$)
//CRA&PORT JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
//*
//RUN      EXEC PGM=IKJEFT01,DYNAMNBR=125,REGION=0M,TIME=NOLIMIT
//*
//* DD used by RAM
//CRARAM1 DD DISP=SHR,DSN=FEK.#CUST.CRARAM1          * PDS RAM
//*
//* DD used by CARMA server (CRASERV)
//STEPLIB DD DISP=SHR,DSN=FEK.SFEKLOAD
//CRADEF  DD DISP=SHR,DSN=FEK.#CUST.CRADEF
//CRAMSG  DD DISP=SHR,DSN=FEK.#CUST.CRAMSG
//CRASTRS DD DISP=SHR,DSN=FEK.#CUST.CRASTRS
//CARMALOG DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//*
//* DD used by ISPF (ISPSTART)
//ISPPROF DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          LRECL=80,RECFM=FB,SPACE=(TRK,(1,1,5))
//ISPCTL0 DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          LRECL=80,RECFM=FB,SPACE=(TRK,(5,5))
//ISPCTL1 DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          LRECL=80,RECFM=FB,SPACE=(TRK,(5,5))
//ISPMLIB DD DISP=SHR,DSN=ISP.SISPMENU
//ISPPLIB DD DISP=SHR,DSN=ISP.SISPPENU
//ISPSLIB DD DISP=SHR,DSN=ISP.SISPSENU
//ISPTLIB DD DISP=SHR,DSN=ISP.SISPTENU
//*
//* DD used by TSO (IKJEFT01)
//SYSPROC DD DISP=SHR,DSN=#h1q.SFEKPROC          * CRAALLOC
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
%CRALLOC ISPSTART PGM(CRASERV) PARM(&CRAPRM1 &CRAPRM2 &CRAPRM3 +
&CRAPRM4 &CRAPRM5 &CRAPRM6 &CRAPRM7 &CRAPRM8 )
$$
EXIT CODE(0)
```

Figure 20. CRASUBMT: Batch submit with sample RAMs

## (Optional) Additional custom RAM customization

The custom RAMs have additional components that can be customized if needed.

- CARMA startup processing has an optional user exit. For more information see (Optional) CARMA user exit.

- Custom RAM startup has an allocation exec, FEK.SFEKPROC(CRAALLOC), that can be customized. For more information, see CRAALLOC, the custom RAM allocation exec.

---

## CARMA configuration details

The different configuration scenarios that are documented in this publication share many of the CARMA configuration files. The details of these configuration files are documented here, and they are referenced from within the various scenarios.

### CRASRV.properties, the RSE interface to CARMA

The CARMA server provides a standard API for other products that use host systems to access one or more Software Configuration Managers (SCMs). However, it does not provide methods for direct communication with a client computer. For this communication, it relies on other products, such as the RSE server. The RSE server uses the settings in `CRASRV.properties` to start and connect to a CARMA server.

`CRASRV.properties` is located in `/etc/rdz/`, unless you specified a different location when you customized and submitted FEK.SFEKSAMP(FEKSETUP) job. For more details, see “Customization setup” on page 15. You can edit the file with the TSO **EDIT** command.

**Note:** For the changes to take effect, the RSED started task must be restarted.

```
# CRASRV.properties - CARMA configuration options
#
port.start=0
#port.range=100
#user.exit='FEK.SFEKSAMP(CRAEXIT) '
startup.script.name=carma.startup.rex
clist.dsname='FEK.#CUST.CNTL(CRASUBMT) '
crastart.configuration.file=crastart.conf
#crastart.stub=/usr/lpp/rdz/bin/CRASTART
#crastart.syslog=Partial
#crastart.timeout=420
#crastart.steplib=FEK.SFEKLPA
#crastart.tasklib=TASKLIB
```

Figure 21. `CRASRV.properties` – CARMA configuration file

#### **port.start**

When the value of `port.start` is 0 (zero), CARMA uses an ephemeral port for communication between CARMA and the RSE server. In this scenario, TCP/IP assigns a random free port number. When the value of `port.start` is non-zero, it is interpreted as the starting point of a port range used for communication between CARMA and the RSE server, in which case the `port.range` variable must also be defined. The default port is 0. Communication on this port is confined to your host system.

**Note:** Before selecting a port, verify that the port is available on your system by using the **NETSTAT** and **NETSTAT PORTL** commands. For more information, see "Reserved TCP/IP ports" in the *Host Configuration Reference* (SC14-7290).

#### **#port.range**

Range of ports, starting at `port.start`, which is used for CARMA communication if `port.start` is non-zero. The default is 100. To specify the size of the port range, uncomment and customize. For example, when

port.start is 5227 and port.range is 100, port 5227 until 5326 (both inclusive) can be used by CARMA. Each CARMA connection uses a port exclusively, so specifying a port range limits the maximum number of concurrent CARMA sessions.

#### **#user.exit**

Defines user-specified code to be executed during CARMA startup. Uncomment and specify the data set name of the code to be executed.

With quotes (') the data set name is an absolute reference, without quotes (') the data set name is prefixed with the client's user ID, not the TSO prefix. The latter requires that all CARMA users must maintain their own exit code.

A sample user exit is provided as SFEKSAMP(CRAEXIT). This sample also documents the startup arguments passed to the user exit. For more information see (Optional) CARMA user exit.

#### **startup.script.name**

Defines the CARMA startup script. The default is carma.startup.rex. This REXX exec triggers the startup of a CARMA server. The file name can be specified in several ways:

- Null string, which means that the variable is not specified. In this case, the default value is used.
- Only a file name, which is the default method. CARMA searches the directories in the PATH environment variable to find the file. The directory holding Developer for System z executables (/usr/lpp/rdz/bin by default) is automatically added to the PATH environment variable.
- Relative path, which is the directory and file name, without a leading forward slash (/). CARMA adds your configuration directory (/etc/rdz/ by default) to the provided path to make it an absolute path.
- Absolute path, which is the directory and file name, with a leading forward slash (/). CARMA uses the specified file location.

#### **clist.dsname**

Defines the startup method for the CARMA server. For more details about the different startup methods, see "Select the server startup method and active RAM" on page 58.

- \*CRASTART indicates that the CARMA server should be started as a subtask within RSE using CRASTART. If you specify \*CRASTART, you must also specify the crastart.\* directives, or use their default values.
- \*ISPF indicates that the CARMA server should be started using ISPF's TSO/ISPF Client Gateway. This startup method is deprecated.
- Any other value defines the location of the CRASUBMT CLIST, using TSO-like naming conventions. With single quotation marks (') the data set name is an absolute reference, without the single quotation marks (') the data set name is prefixed with the client's user ID, not the TSO prefix. The latter requires that all CARMA users must maintain their own CRASUBMT CLIST.

The default is 'FEK.#CUST.CNTL(CRASUBMT)'. This CLIST starts a CARMA server when opening a connection by using the batch submit method.

#### **crastart.configuration.file**

Specifies the name of the CRASTART configuration file. The default is crastart.conf. This file specifies the data set allocations and program

invocations that are needed to start a CARMA server. This directive is used only if the `clist.dsname` directive has `*CRASTART` as value. The file name can be specified in several ways:

- Null string, which means that the variable is not specified. The default value is used.
- Only a file name, which is the default method. CARMA searches your configuration directory (`/etc/rdz` by default) to find the file.
- Relative path, which is the directory and file name, without a leading forward slash (`/`). CARMA adds your configuration directory (`/etc/rdz/` by default) to the provided path to make it an absolute path.
- Absolute path, which is the directory and file name, with a leading forward slash (`/`). CARMA uses the specified file location.

**#crastart.stub**

z/OS UNIX stub for calling CRASTART. The default is CRASTART. This stub makes the MVS based CRASTART load module available to z/OS UNIX processes. To specify a specific path, uncomment and customize. This directive is used only if the `clist.dsname` directive has `*CRASTART` as value. The file name can be specified in several ways:

- Null string, which means that the variable is not specified. The default value is used.
- Only a file name, which is the default method. CARMA searches the directories in the `PATH` environment variable to find the file. The directory holding Developer for System z executables (`/usr/lpp/rdz/bin` by default) is automatically added to the `PATH` environment variable.
- Relative path, which is the directory and file name, without a leading forward slash (`/`). CARMA adds your configuration directory (`/etc/rdz/` by default) to the provided path to make it an absolute path.
- Absolute path, which is the directory and file name, with a leading forward slash (`/`). CARMA uses the specified file location.

**#crastart.syslog**

Specifies how much information is written to the system log while CRASTART starts a CARMA server. The default is `Partial`. Valid values are listed in the following table.

A (All)	All tracing information is printed to SYSLOG
P (Partial)	Only connect, disconnect, and error information is printed to SYSLOG
anything else	Only error conditions are printed to SYSLOG

To specify the required detail level for system log messages, uncomment and customize. This directive is used only if the `clist.dsname` directive has `*CRASTART` as value.

**#crastart.timeout**

The length of time, in seconds, before a CARMA server ends due to lack of activity. The default is 420 (7 minutes). To specify the required timeout value, uncomment and customize. This directive is used only if the `clist.dsname` directive has `*CRASTART` as value.

**Note:** System abend 522 for module CRASERV will occur if the `JWT` parameter in the `SMFPRMxx` parmlib member is set to a value lower than the

crastart.timeout value in CRASRV.properties. This occurrence does not impact CARMA operations because the server is restarted automatically if needed.

#### #crastart.steplib

The location of the CRASTART module when accessed through the STEPLIB directive in rsed.envvars. The default is FEK.SFEKLPA. If the CRASTART module cannot be part of LPA or LINKLIST, uncomment and customize this directive. Program control and APF issues might arise if the CRASTART module is not in LPA. This directive is used only if the clist.dsname directive has \*CRASTART as value.

#### #crastart.tasklib

Alternate name for the TASKLIB DD name in crastart.conf. The default is TASKLIB. If the DD name TASKLIB has a special meaning for your SCM or RAM and cannot be used as STEPLIB replacement, uncomment and customize this directive. This directive is used only if the clist.dsname directive has \*CRASTART as value.

## crastart\*.conf, the CRASTART server startup

RSE starts the CRASTART load module, which uses the definitions in crastart\*.conf to create a valid environment to execute batch TSO and ISPF commands. Developer for System z uses this environment to run the CARMA server, CRASERV.

crastart\*.conf is located in /etc/rdz/, unless you specified a different location when you customized and submitted job FEK.SFEKSAMP(FEKSETUP). For more details, see “Customization setup” on page 15. You can edit the file with the TSO **EDIT** command.

**Note:** Changes are in effect for all CARMA servers that are started after the update.

Developer for System z provides multiple crastart\*.conf configuration files. Each of these sample files is preconfigured for a specific customization scenario:

- crastart.endevor.conf is configured for CRASTART startup with CA Endeavor<sup>®</sup> SCM RAM.
- crastart.conf is configured for CRASTART startup with sample RAMs.

The function of the crastart\*.conf file is similar in concept to a JCL job stream, but is more restrictive.

- The following samples show valid line formats:
  - \* comment
  - ddname=dsn1,dsn2,dsn3 \* comment
  - ddname=SYSOUT(c) \* comment
  - ddname=DUMMY \* comment
  - -COMMAND=<any bpxwdyn command> \* comment
  - PROGRAM = progname parms \* comment

**Note:** The **BPXWDYN** command is documented in *Using REXX and z/OS UNIX System Services* (SA22-7806) and allows complex allocation constructs.

- All input is changed to uppercase.
- Line continuations are not supported.
- There is no limitation on line length.

- One or more blank spaces are allowed around the equal sign (=).
- DD allocations must precede the related PROGRAM statement.
- DD names allocated here are freed at the end of program execution. They do not accumulate.
- DD names allocated by the called programs are not freed.
- Multiple data sets can be concatenated to a DD name. The data set names must be separated by a comma (,), and the concatenation is searched in the listed order.
- All data set allocations are done with DISP=SHR, except for allocations done using -COMMAND.
- Inline data is not supported. All data must be in cataloged files.
- Variables can be used only on the right side of the equal sign (=).
- The following variables are supported:

&CRAUSER.	Client user ID
&CRADATE.	Current <sup>®</sup> date in Dyyyymmdd format (7 char Julian)
&CRATIME.	Current time in Thhmmss format (hour min sec)
&CRAPRM1.	Port number
&CRAPRM2.	Timeout value
System symbol	Any SYS1.PARMLIB(IEASYMxx) system symbol
-<ddname>	A hyphen (-) followed by a previously defined DD name acts like a *.ddname backward reference in JCL. The original DD must be allocated using the -COMMAND statement.

**Note:** There is no variable for the TSO prefix because TSO is not active when the configuration file is interpreted. If you have a need for the TSO prefix or other variable that is not available, see “(Optional) Custom allocation exec” on page 86.

Figure 22 on page 73 shows a basic crastart\*.conf skeleton that includes ISPF services.

```

* DD used by RAM

* DD used by CARMA server (CRASERV)
TASKLIB = FEK.SFEKLOAD
CRADEF = FEK.#CUST.CRADEF
CRAMSG = FEK.#CUST.CRAMSG
CRASTRS = FEK.#CUST.CRASTRS
CARMALOG= SYSOUT(H)
SYSPRINT= SYSOUT(H)

* DD used by ISPF (ISPSTART)
-COMMAND=ALLOC FI(ISPCTL0) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(ISPCTL1) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(ISPPROF) NEW DELETE DSORG(PO) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA) DIR(5)
ISPTABL = -ISPPROF
ISPTLIB = -ISPPROF,ISP.SISPTENU
ISPMLIB = ISP.SISPMENU
ISPPLIB = ISP.SISPPENU
ISPSLIB = ISP.SISPSENU

* DD used by TSO (IKJEFT01)
SYSPROC = #h1q.SFEKPROC * CRAALLOC
SYSTSIN = DUMMY
SYSTSPRT= SYSOUT(H)

PROGRAM=IKJEFT01 %CRAALLOC ISPSTART PGM(CRASERV) PARM(&CRAPRM1.
&CRAPRM2. &CRAPRM3. &CRAPRM4. &CRAPRM5. &CRAPRM6. &CRAPRM7.
&CRAPRM8. )

```

Figure 22. *crastart\*.conf*: CARMA server startup using CRASTART

**Note:**

- Due to page width limitations, some lines in the sample wrapped onto the next line. All lines that start with an indentation should be added to the end of the previous line.
- If you alter the PROGRAM line, ensure that there is at least one blank before the closing round bracket (“)”) of the PARM() statement to simplify processing of the string.
- You can add your own DD statements and data set concatenations to customize the CARMA TSO environment, thus mimicking a TSO logon procedure.
- The DD name TASKLIB acts like STEPLIB in JCL. Its DD name must match the value specified for *crastart.tasklib* in *CRASRV.properties*, which is described in “CRASRV.properties, the RSE interface to CARMA” on page 68.
- Regular APF rules apply for TASKLIB allocations. Libraries lose their APF authorization when a non-APF authorized library is part of the concatenation.
- System abend 522 for module CRASERV occurs if the JWT parameter in the *SMFPRMxx* parmlib member is set to a value lower than the *crastart.timeout* value in *CRASRV.properties*. The system abend does not impact CARMA operations because the server is restarted automatically if needed.
- Details of the CARMA server startup are shown in *rsecomm.log* when the server ends. For more information on setting the detail level of *rsecomm.log*, see “(Optional) *rsecomm.properties*, the RSE tracing” on page 112.

**Collecting the CRASTART log files**

CRASTART creates a TSO environment as a child process of RSE, which runs in a separate address space. Non-trivial actions might be needed to keep the CARMA output sent to *SYSOUT(\*)*, which complicates the collecting of log files. This

difficulty can be resolved by writing the log files to a user-specific data set, as shown in the following sample allocation:

```
-COMMAND=ALLOC FI(CARMALOG) MOD CATALOG DSORG(PS) RECFM(F,B) LRECL(133)
  BLKSIZE(27930) SPACE(5,5) TRACKS UNIT(SYSALLDA)
  DA(&CRAUSER..&SYSNAME..CRA.CARMALOG)
```

**Note:**

- Due to page width limitations, some lines in the sample wrapped onto the next line. All lines that start with an indentation should be added to the end of the previous line.
- To be able to create user-specific log files, this log file must be allocated using the `-COMMAND` statement.
- You can also allocate the log data sets in an allocation exec if you need more flexibility; for example, only send the log to a data set for specific users. For more information about allocation execs, see “(Optional) Custom allocation exec” on page 86.

If you are writing log files to `SYSOUT`, remember that `SYSOUT` allocated by z/OS UNIX processes is treated as special output in JES. This is similar to `SYSOUT` allocated by APPC transactions.

- While the CARMA server is still active, the output can be seen using the **DA** command in SDSF. The job will have the user's user ID followed by a random one-digit number as job name and an STC job ID. The user is the job owner.
- If the output was written to a HOLD output class, when the CARMA server ends, due to inactivity or the user ending the connection, the output can be seen using the **APPC ON** and **H ALL** commands in SDSF. The job name, job ID, and job owner remain the same. Each DD shows up as a separate spool file, without any indication which DD it is.
- JES Job Monitor can also show the output if `SEARCHALL=ON` is active in `FEJJCNFG` and the output resides on the spool in a HOLD output class. For more information about the `SEARCHALL` directive, see “`FEJJCNFG`, the JES Job Monitor configuration file” on page 30.

## CRASUB\*, the batch submit server startup

RSE starts CLIST `CRASUB*`, which in turn submits an embedded JCL to create a valid environment to execute batch TSO and ISPF commands. Developer for System z uses this environment to run the CARMA server, `CRASERV`.

`CRASUB*` is located in `FEK.#CUST.CNTL`, unless you specified a different location when you customized and submitted the `FEK.SFEKSAMP(FEKSETUP)` job. For more details, see “Customization setup” on page 15.

**Note:** Changes are in effect for all CARMA servers that are started after the update.

Developer for System z provides multiple `CRASUB*` JCL jobs. Each of these sample files is pre-configured for a specific customization scenario:

- `CRASUBCA` is configured for batch startup with CA Endeavor<sup>®</sup> SCM RAM.
- `CRASUBMT` is configured for batch startup with sample RAMs.

Figure 23 on page 75 shows a basic `CRASUB*` skeleton that includes ISPF services.



**Note:**

- If you need to merge the definitions for a (custom) RAM into an existing VSAM configuration, see the FEK.#CUST.JCL(CRA#UADD) sample job. This job must be customized and submitted for each CARMA VSAM file that changes. For more information about the record structure used by the different CARMA VSAM files, see the *Common Access Repository Manager Developer's Guide* (SC23-7660).
- Use the FEK.#CUST.JCL(CRA#UQRY) sample job to extract the active definitions from a VSAM to a sequential data set.

**CRADEF, the configuration data set**

This VSAM data set describes the functions that are supported by the defined RAMs. RAM developers require UPDATE access to this data set. The data set can be created by one of these sample jobs:

- CRA\$VCAD populates the data set with CA Endeavor® SCM RAM data.
- CRA\$VDEF populates the data set with sample RAM data.

The mentioned sample jobs can be used to disable a defined RAM during VSAM creation. Doing so enables you to create a customized CARMA setup by using a single master input file, which can be one provided by IBM or customized by your RAM developers.

**CRAMSG, the message data set**

This VSAM data set holds messages issued by the CARMA server itself. The data set can be created by one of these sample jobs:

- CRA\$VMSG populates the data set with generic server data.

**CRASTRS, the custom string data set**

This VSAM data set holds the messages that are issued by the defined RAMs. RAM developers require UPDATE access to this data set. The data set can be created by one of these sample jobs:

- CRA\$VCAS populates the data set with CA Endeavor® SCM RAM data.
- CRA\$VSTR populates the data set with sample RAM data.

**CARMA VSAM migration notes**

1. Beginning with version 7.6.1, Developer for System z supports a new data structure layout for the CARMA custom information VSAM data set, CRASTRS, to remove message length limitations.

Prior to Developer for System z version 7.6.1, strings defined in the CARMA custom information VSAM data set are limited to predefined lengths. This limitation forces RAM developers to shorten descriptive strings, or to use client-side plug-ins to display full-length strings.

The new VSAM record structure supports a variable-length data structure layout for the CARMA custom information VSAM data set, CRASTRS, where strings are separated by a delimiter character instead of being of fixed length.

Customize and submit the FEK.SFEKSAMP(CRA#VS2) JCL to convert your existing, fixed-length, CARMA custom information VSAM data set, CRASTRS, to the new variable-length format.

**Note:**

- Beginning with version 7.6.1, the sample CARMA custom information VSAM data set is provided in variable-length format.

- Beginning with version 7.6.1, the CARMA load module, CRASERV, supports both the fixed-length format and the variable-length format for the CARMA custom information VSAM data set.
  - Older versions of the CARMA load module do not support the variable-length format and produce garbled strings when used with a variable-length CARMA custom information VSAM data set.
2. The introduction of the CA Endeavor<sup>®</sup> SCM packages RAM removes the need for providing package-related actions in the CA Endeavor<sup>®</sup> SCM element RAM. Therefore, providing package-related actions in the CA Endeavor<sup>®</sup> SCM RAM is deprecated, and the CARMA VSAM data sets in version 8.5 no longer hold the definitions required for this functionality.

You can restore this functionality by customizing and submitting the FEK.SFEKSAMP(CRA#UADD) JCL to merge the removed items back into the CARMA VSAM data sets.

- Merge FEK.SFEKVSM2(CRA0VPKD) into FEK.#CUST.CRADEF.
- Merge FEK.SFEKVSM2(CRA0VPKS) into FEK.#CUST.CRASTRS.

This merge action is required each time the VSAM data sets are updated during product maintenance.

## CARMA Repository Access Managers (RAMs)

A Repository Access Manager (RAM) is an Application Programming Interface (API) for a z/OS based Software Configuration Manager (SCM). In turn, Developer for System z or user-written applications can start a CARMA server, which loads the RAMs and provides a standard interface to access any supported SCM.

The CARMA server must be able to find the RAM load modules, either through LINKLIST or STEPLIB/TASKLIB.

The CRAR\* RAM load modules that are provided by Developer for System z are located in FEK.SFEKLOAD, and the sample source code and compile jobs are located in FEK.SFEKSAMP, unless you used a different high level qualifier during the SMP/E install of Developer for System z.

The following sections have customization notes for the RAMs that are available with Developer for System z. The referenced sample members are located in FEK.#CUST.\*, unless you specified a different location when you customized and submitted the FEK.SFEKSAMP(FEKSETUP) sample job. For more details, see “Customization setup” on page 15.

For in-depth knowledge of CARMA and for more information on the sample RAMs and sample source code provided, see *Common Access Repository Manager Developer's Guide* (SC23-7660).

### CA Endeavor<sup>®</sup> SCM RAM

- The CA Endeavor<sup>®</sup> SCM RAM is a production-type RAM.
- The CA Endeavor<sup>®</sup> SCM RAM gives Developer for System z clients direct access to CA Endeavor<sup>®</sup> SCM elements.
- The load module name is CRARNDVR.
- The CA Endeavor<sup>®</sup> SCM RAM has many additional settings compared to a conventional CARMA setup. Use one of the preconfigured setups that support the CA Endeavor<sup>®</sup> SCM RAM as starting point, and customize it to fit your needs.

- The deprecated TSO/ISPF Client Gateway startup method cannot be used with the CA Endeavor<sup>®</sup> SCM RAM.
- The CA Endeavor<sup>®</sup> SCM RAM has multiple configuration files, FEK.#CUST.PARMLIB(CRACFG), FEK.#CUST.PARMLIB(CRASHOW) and FEK.#CUST.PARMLIB(CRATMAP), that can be customized. For more information, see “CRACFG, CRASHOW and CRATMAP, the CA Endeavor<sup>®</sup> SCM RAM configuration files” on page 79.
- The CA Endeavor<sup>®</sup> SCM RAM has an allocation exec, FEK.SFEKPROC(CRANDVRA), that can be customized. See “CRANDVRA, the CA Endeavor<sup>®</sup> SCM RAM allocation exec” on page 80 for more information.
- The CA Endeavor<sup>®</sup> SCM RAM supports doing CA Endeavor<sup>®</sup> SCM actions in batch mode, in the background. Batch-actions requires a configuration file, FEK.#CUST.PARMLIB(CRABCFG), and a skeleton JCL, FEK.#CUST.CNTL(CRABATCA), that must be customized. For more information, see “CA Endeavor<sup>®</sup> SCM RAM batch actions” on page 81.

### CA Endeavor<sup>®</sup> SCM packages RAM

- The CA Endeavor<sup>®</sup> SCM packages RAM is a production-type RAM.
- The CA Endeavor<sup>®</sup> SCM packages RAM gives Developer for System z clients direct access to CA Endeavor<sup>®</sup> SCM packages.
- The load module name is CRARPKGS.
- The CA Endeavor<sup>®</sup> SCM packages RAM does not have customizable settings, and must be used in combination with the CA Endeavor<sup>®</sup> SCM RAM.

### PDS RAM

- The PDS RAM is a sample RAM. Do not use in a production environment.
- The PDS RAM gives a data set list similar to **MVS Files -> My Data Sets** in the Remote Systems view.
- The load module name is CRARPDS.
- The PDS RAM requires that ISPF services be available.
- The PDS RAM requires an additional VSAM data set to be allocated to DD CRARAM1. This VSAM data set can be allocated and primed with the FEK.#CUST.JCL(CRA#VPDS) sample job. For customization instructions, see the documentation within the member.
- Source code and compile jobs are available in FEK.SFEKSAMP. For more information, see *Common Access Repository Manager Developer's Guide* (SC23-7660).

### Skeleton RAM

- The skeleton RAM is a sample RAM. Do not use in a production environment.
- The skeleton RAM gives a functional framework that can be used as starting point to develop your own RAM.
- The load module name is CRARTEST.
- Source code and compile jobs are available in FEK.SFEKSAMP. For more information, see *Common Access Repository Manager Developer's Guide* (SC23-7660).

### SCLM RAM

- The SCLM RAM is a sample RAM. Do not use in a production environment.
- The SCLM RAM gives a basic entry into SCLM, ISPF's Software Configuration Manager. This RAM is not enabled by default.
- The load module name is CRARSCLM.
- The SCLM RAM needs the ISPF services to be available.

- The SCLM RAM requires an additional VSAM data set to be allocated to DD CRARAM2. This VSAM data set can be allocated and primed with the FEK.#CUST.JCL(CRA#VSLM) sample job. For customization instructions, see the documentation within the member.
- The SCLM RAM requires the various user-specific data sets to exist. Customize FEK.#CUST.JCL(CRA#ASLM) to allocate these data sets. For customization instructions, see the documentation within the member. Each user must submit CRA#ASLM once before using CARMA with the SCLM RAM. Failing to do so will result in an allocation error.
- The SCLM RAM is not enabled by default. To enable the RAM, it must be defined in the CARMA VSAM data sets referenced by DD CRADEF and CRASTRS. Use the FEK.#CUST.JCL(CRA#UADD) sample job to merge FEK.SFEKVSM2(CRA0SLMD) into CRADEF and FEK.SFEKVSM2(CRA0SLMS) into CRASTRS. For customization instructions, see the documentation within the member.
- Source code and compile jobs are available in FEK.SFEKSAMP. For more information, see *Common Access Repository Manager Developer's Guide (SC23-7660)*.

## CRACFG, CRASHOW and CRATMAP, the CA Endeavor® SCM RAM configuration files

The following CA Endeavor® SCM RAM-specific CARMA components can be customized, regardless of the chosen server startup method. The sample members referenced below are located in FEK.#CUST.PARMLIB, unless you specified a different location when you customized and submitted the FEK.SFEKSAMP(FEKSETUP) job. For more details, see “Customization setup” on page 15.

### CRACFG, CA Endeavor® SCM RAM interaction with the SCM

CRACFG specifies how the CA Endeavor® SCM RAM interacts with CA Endeavor® SCM. Refer to the documentation within the member for customization instructions if you want to change the defaults.

```
# ENTRY-STAGE-COPY-MODE = RETRIEVE-ADD
```

*Figure 24. CRACFG - CA Endeavor® SCM RAM interaction with the SCM*

### CRASHOW, CA Endeavor® SCM RAM default filters

CRASHOW defines default filters for CA Endeavor® SCM environments, systems, and so forth. Refer to the documentation within the member for customization instructions if you want to change the defaults.

```
ENV=*
TOENV=
STGID=*
TOSTGID=
SYS=*
SUBSYS=*
ELEM=*
TOELEM=
TYPE=*
#FILTER-DEP=YES
```

*Figure 25. CRASHOW - CA Endeavor® SCM RAM default filters*

**Note:** FILTER-DEP is not a common CA Endeavor® SCM variable, but a Developer for System z specific variable that controls dependency scans for elements with footprint references to other CA Endeavor® SCM repository locations.

## CRATMAP, the CA Endeavor® SCM RAM file extension mappings

CRATMAP overrides the CA Endeavor® SCM type to file extension mappings. If you want to change the defaults, see the customization instructions in the documentation within the member.

```
# *      = cb1
# COBOL  = cb1
# COPY   = cpy
# ASM    = asm
# MACRO  = asm
# PROCESS = jcl
```

Figure 26. CRATMAP: CA Endeavor® SCM RAM default filters

## CRANDVRA, the CA Endeavor® SCM RAM allocation exec

Both the batch submit and the CRASTART startup method call the CRANDVRA REXX exec to allocate user-specific data sets used by CA Endeavor® SCM RAM. The allocations are done in a separate exec, because an exec allows more flexibility than what is possible within the batch submit CRASUBCA JCL and the CRASTART crastart.endeavor.conf configuration file. The allocation exec is also responsible for calling the optional user exit.

DD	Data set name	Type
DEPEND	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.DEPEND	Permanent
BROWSE	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.BROWSE	Temporary
C1PRINT	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.LISTING	Temporary
SPCLLIST	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.SPCLLIST	Temporary
PKGSCLS	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.PKGSCLS	Temporary
CRABJCLO	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.CRABJCLO	Temporary
ENHCEDIT	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.ENHCEDIT	Temporary
CRAPARM	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.CRAPARM	Temporary

You can customize a copy of this allocation REXX exec if certain defaults, such as the data set name, do not match your site standards. CRANDVRA is located in FEK.SFEKPROC, unless you used a different high-level qualifier during the SMP/E install of Developer for System z.

For customization instructions, see the documentation within the member. For more information about allocation execs, see “(Optional) Custom allocation exec” on page 86.

**Note:** You should copy the sample allocation REXX to a new data set and customize this copy to avoid overwriting it when applying maintenance. When you do this, you must update the reference to SFEKPROC in the SYSEXEC DD of your chosen CARMA startup method to match your new data set name.

## CA Endeavor<sup>®</sup> SCM RAM batch actions

Normally, CA Endeavor<sup>®</sup> SCM actions such as “Generate Element” are executed “online”, in the CARMA server address space. This behavior causes problems if your CA Endeavor<sup>®</sup> SCM procedures call TSO, because TSO is already active and that means that the required DDs such as SYSTSIN and SYSTSPRT are in use.

To resolve this problem, the CA Endeavor<sup>®</sup> SCM RAM supports “batch actions” since version 8.0.3. When batch-actions is enabled, the CA Endeavor<sup>®</sup> SCM RAM submits a customizable batch job to perform actions like “Generate Element”. Using a batch job results in the allocation of DDs such as SYSTSIN and SYSTSPRT by your CA Endeavor<sup>®</sup> SCM procedures, because the submitted JCL does not require TSO to be active.

CA Endeavor<sup>®</sup> SCM RAM batch-actions are the Developer for System z equivalent of background CA Endeavor<sup>®</sup> SCM actions.

When a request is issued to execute an action that is supported by batch-actions, the CA Endeavor<sup>®</sup> SCM RAM checks for the existence of the CRABCFG DD, in CRASUBCA or crastart.endeavor.conf, and checks that the setup behind this DD is valid. If CRABCFG exists and the setup is valid, the action is performed in batch. If CRABCFG does not exist, the action is performed online. Version 8.0.3 or later clients have the facility to override this behavior.

For example:

```
/* uncomment CRABCFG and CRABSKEL to use batch actions
/*CRABCFG DD DISP=SHR,DSN=FEK.#CUST.PARMLIB(CRABCFG)
/*CRABSKEL DD DISP=SHR,DSN=FEK.#CUST.CNTL
```

### Note:

- The TSO-free environment is available only for selected CA Endeavor<sup>®</sup> SCM actions. Batch-actions does not support a TSO-free environment outside this scope.
- The CRABCFG configuration file documents which CA Endeavor<sup>®</sup> SCM actions are supported.
- A functional sample job, FEK.#CUST.CNTL(CRABATCA), is provided to execute the batch actions, but the intent of batch-actions is that this sample is customized to start your current CA Endeavor<sup>®</sup> SCM procedures.
- Ensure that there are sufficient JES initiators available in the class used to submit the batch-action JCLs.
- When using JES in a SYSPLEX environment, ensure that the job runs on the current system, or that the completion information is routed back to the system hosting Developer for System z, so that the CA Endeavor<sup>®</sup> SCM RAM can check the status.
- If both Developer for System z client and host system are at version 8.5.1 or later, the client can provide a customized JOB card and additional JCL statements to the batch-action JCL before submission.

## CRABCFG, the CA Endeavor<sup>®</sup> SCM RAM batch-action configuration

CRABCFG defines the configuration variables related to CA Endeavor<sup>®</sup> SCM RAM batch-actions.

CRABCFG is located in FEK.#CUST.PARMLIB, unless you specified a different location when you customized and submitted the FEK.SFEKSAMP (FEKSETUP) job. For more details, see “Customization setup” on page 15.

See the following CRABCFG sample file, which must be customized to match your system environment. Comment lines start with a number sign (#) when using a US code page. Comments behind a directive and its assigned value are supported. Spaces around the equal sign (=) are supported. Line continuations are not supported.

**Note:** Changes are in effect for all CARMA servers that are started after the update.

```
# Location of batch action JCL
SKELETON-DD = CRABSKEL
#
# batch action JCL members within SKELETON-DD
DEFAULT-JOBCARD = CRABJOBC
ADD-ELEMENT     = CRABATCA
GENERATE-ELEMENT = CRABATCA
MOVE-ELEMENT    = CRABATCA
DELETE-ELEMENT  = CRABATCA
RETRIEVE-ELEMENT = CRABATCA
SIGNIN-ELEMENT  = CRABATCA
PRINT-ELEMENT   = CRABATCA
PRINT-MEMBER    = CRABATCA
#
# Command substitution key within batch action JCL
BSTIPT01-KEY = <CRA_BSTIPT01>
```

Figure 27. CRABCFG: CA Endeavor<sup>®</sup> SCM RAM batch-action configuration

### **SKELETON-DD**

Name of the DD statement that references one or more PDS(E) data sets that hold the batch-action skeleton JCLs. The sample value is CRABSKEL. Can be changed if needed. This DD must be defined to the CARMA server in CRASUBCA or crastart.endeavor.conf.

### **DEFAULT-JOBCARD**

Name of the member holding a default JOB card. If not overruled by a user-specific JOB card stored on the version 8.5.1 or later Developer for System z client, this default JOB card is used to substitute the <JOB CARD> key in a skeleton JCL. Can be changed if needed.

### **GENERATE-ELEMENT and other CA Endeavor<sup>®</sup> SCM actions**

The key names represent the CA Endeavor<sup>®</sup> SCM actions that are supported by batch-action and cannot be changed. The value assigned to each key is the member name of the related skeleton JCL. The sample value is CRABATCA for all keys. Can be changed if needed.

### **BSTIPT01-KEY**

Substitution key for the actual CA Endeavor<sup>®</sup> SCM command string. The sample value is <CRA\_BSTIPT01>. Can be changed if needed. The first occurrence, but not in a comment, of this substitution key within the skeleton JCL is replaced by the command string that instructs CA Endeavor<sup>®</sup> SCM to do the requested action against the requested element.

### **CRABATCA, the CA Endeavor<sup>®</sup> SCM RAM batch action JCL**

CRABATCA is a sample skeleton JCL used for batch-actions. To change the defaults, see the customization instructions in the documentation within the member.

CRABATCA is located in FEK.#CUST.CNTL, unless you specified a different location when you customized and submitted the FEK.SFEKSAMP(FEKSETUP) job. For more details, see “Customization setup” on page 15.

Changes are active for all new invocations. No server restart is needed.

```
//<JOB CARD>
//*
//CRABATCA EXEC PGM=NDVRC1,DYNAMNBR=1500,REGION=4096K,PARM='C1BM3000'
//STEPLIB DD DISP=SHR,DSN=CA.NDVRO.AUTHLIB * NDVR R12
// DD DISP=SHR,DSN=CA.NDVR.AUTHLIB * NDVR R12
//* DD DISP=SHR,DSN=CA.NDVR.CSIQAUTU * NDVR R14
//* DD DISP=SHR,DSN=CA.NDVR.CSIQAUTH * NDVR R14
//CONLIB DD DISP=SHR,DSN=CA.NDVR.CONLIB * NDVR R12
//*CONLIB DD DISP=SHR,DSN=CA.NDVR.CSIQLOAD * NDVR R14
//C1MSG1 DD SYSOUT=*
//C1MSG2 DD SYSOUT=*
//C1PRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133)
//SYSOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYMDUMP DD DUMMY
//SYSIN DD DUMMY
//BSTIPT01 DD *
SET STOPRC 16 .
<CRA_BSTIPT01>
/*
```

Figure 28. CRABATCA: CA Endeavor® SCM RAM batch-action JCL

### CRABJOB, the CA Endeavor® SCM RAM batch action JOB card

CRABJOB is a sample default JOB card used for batch-action skeleton JCL that specifies the <JOB CARD> key. To change the defaults, see customization instructions in the documentation within the member.

CRABJOB is located in FEK.#CUST.CNTL, unless you specified a different location when you customized and submitted the FEK.SFEKSAMP(FEKSETUP) job. For more details, see “Customization setup” on page 15.

Changes are active for all new invocations. No server restart is needed.

```
//<USERID>B JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
//*PROCS JCLLIB ORDER=(COBOL.V4R1M0.SIGYPROC,CBC.SCCNPRC)
```

Figure 29. CRABJOB: CA Endeavor® SCM RAM batch-action JOB card

### CRAALLOC, the custom RAM allocation exec

Both the batch submit and the CRASTART startup method call the CRAALLOC REXX exec to allocate user-specific data sets that can be used by a user-written RAM. The allocations are done in a separate exec, because an exec allows more flexibility than what is possible within the batch submit CRASUBMT JCL and the CRASTART crastart.conf configuration file. The allocation exec is also responsible for calling the optional user exit.

DD	Data set name	Type
CRAPARM	&SYSPREF.&SYSUID.. &SYSNAME..CRA\$CUST.CRAPARM	Temporary

You can customize a copy of this allocation REXX exec if certain defaults, such as the data set name, do not match your site standards. CRAALLOC is located in FEK.SFEKPROC, unless you used a different high-level qualifier during the SMP/E install of Developer for System z.

For customization instructions, see the documentation within the member. For more information about allocation execs, see (Optional) Custom allocation exec.

**Note:** You should copy the sample allocation REXX to a new data set and customize this copy to avoid overwriting it when applying maintenance. When you do this, you must update the reference to SFEKPROC in the SYSEXEC DD of your chosen CARMA startup method to match your new data set name.

## CARMA return codes

CARMA can report various error codes to the client or in the host system logs. The details that are provided with the error, and the information in Table 12, can help you locate the error and work towards a resolution.

Table 12. CARMA return codes

Error range	Error type
4-99	Generic CARMA errors
100-199	Generic RAM errors
200-399	CRASERV (CARMA server) errors
400-499	RSE (CARMA miner) errors
500-899	RAM-specific errors
900-999	TSO and TCP/IP errors

Some common return codes are these:

- 220: CARMA server ends due to inactivity timeout. This is not an error.
- 990: CARMA server is unable to connect to the port on which CARMA miner is listening.

---

## (Optional) Supporting multiple RAMs

CARMA has the facility for defining multiple RAMs and running them concurrently. However, because there is only one CARMA server active for a user, even when there are multiple RAMs, some configuration changes might be required to make this setup work.

RAMs are defined by a RAM developer in the CARMA configuration VSAM data set, CRADEF. During startup, the CARMA server, CRASERV, identifies all of the defined RAMs and sends the information to the CARMA client. The user can then select one or more RAMs, which is loaded into the CARMA server.

Because RAMs are active as plug-ins of the CARMA server, ensure that all prerequisites, such as data set allocations, for each of the RAMs are available in the address space of the CARMA server. This requirement might need changes to the CARMA configuration samples, such as CRASUBMT or crastart.conf, which are included with Developer for System z.

## Example

In the following example, you start from an existing setup with the CA Endeavor<sup>®</sup> SCM RAM, using the CRASTART startup method, and add the sample PDS RAM.

Definitions for the CA Endeavor<sup>®</sup> SCM RAM:

- FEK.SFEKVSM2(CRA0VCAD): CRADEF definitions
- FEK.SFEKVSM2(CRA0VCAS): CRASTRS definitions
- /etc/rdz/crastart.endeavor.conf: CRASTART configuration file

Definitions for the PDS RAM:

- FEK.SFEKVSM2(CRA0VDEF): CRADEF definitions
- FEK.SFEKVSM2(CRA0VSTR): CRASTRS definitions
- FEK.#CUST.CRARAM1: CRARAM1 definitions

The process starts with a RAM developer gathering the data and information needed by the system programmer to complete the setup.

1. Extract the data that is specific for the PDS RAM from the SFEKVSM2 members. These members hold definitions for all sample RAMs, not just the PDS RAM.
2. Merge this data with the CA Endeavor<sup>®</sup> SCM RAM SFEKVSM2 members.
3. Create a list of PDS RAM-specific prerequisites:
  - DD CRARAM1, linked to FEK.#CUST.CRARAM1
  - TSO environment

The system programmer then uses this data to create the updated CARMA VSAM data sets and uses the prerequisite information to create a CRASTART configuration file that is capable of supporting both RAMs.

1. Use the combined data as input for the CRA\$VDEF and CRA\$VSTR jobs to create the updated CARMA configuration and custom information VSAM data sets, CRADEF and CRASTRS. The CRAMSG VSAM is specific for the CARMA server, and thus identical for both RAMs.
2. Add a CRARAM1 definition to crastart.endeavor.conf:  
CRARAM1 = FEK.#CUST.CRARAM1
3. Verify the PROGRAM statement in crastart.endeavor.conf to ensure that it is capable of providing the environment needed by both RAMs:  
PROGRAM=IKJEFT01 %CRANDVRA NDVRC1 PGM(CRASERV)  
PARM(&CRAPRM1. &CRAPRM2.)
  - IKJEFT01: TSO, used to allow certain authorized calls in a non-authorized environment, and used as environment to run the CA Endeavor<sup>®</sup> SCM RAM pre-allocation exec.
  - %CRANDVRA: CA Endeavor<sup>®</sup> SCM RAM pre-allocation exec, located in FEK.SFEKPROC, that allocates temporary user-specific working data sets.
  - NDVRC1: CA Endeavor<sup>®</sup> back end, which has a built-in mechanism to execute TSO and ISPF commands.
  - PGM(CRASERV): Command to start a CARMA server, in ISPF command format.
  - PARM(&CRAPRM1. &CRAPRM2.): Parameters for CRASERV, in ISPF command format. &CRAPRM1 is the port to be used and &CRAPRM2 is the timeout value.

The CA Endeavor<sup>®</sup> SCM RAM is active in an ISPF environment, which implies that the TSO environment required by the PDS RAM is also available.

---

## (Optional) Custom allocation exec

All CARMA server startup methods have limitations regarding data set allocation. For example, TSO prefix substitution is not available in JCL or CRASTART.

However, by creating an exec that is called after TSO or ISPF starts, and before CARMA is started, you can use the whole range of variables and services available in TSO or ISPF to do the required allocations.

Developer for System z uses an allocation exec in each of the pre-configured setups described earlier in this chapter. FEK.SFEKPROC(CRANDVRA), the allocation exec for CA Endeavor<sup>®</sup> SCM RAM and FEK.SFEKPROC(CRAALLOC), the allocation exec for custom RAMs, The exec allocates cataloged temporary data sets that have the user's TSO prefix as high-level qualifier. The allocation exec is also responsible for calling the optional user exit.

Customization instructions are documented within the exec. Changing the allocation exec is supported, but not advised, as customizations must be redone when PTF service updates the exec. If possible, use the CARMA user exit instead, which is described in "(Optional) CARMA user exit."

### Note:

- When updating an allocation exec, ensure you do not destroy allocations done earlier in the CARMA startup process by CRASTART or your startup JCL.
- Output generated by the allocation exec is shown in DD SYSTSPRT of the CARMA server.

When updating an allocation exec, ensure you do not destroy allocations done earlier in the CARMA startup process by CRASTART or your startup JCL.

The following samples show how to start an allocation exec that requires only TSO.

```
crastart*.conf
SYSPROC = my.exec.library
PROGRAM = IKJEFT01 %myexec ISPSTART PGM(CRASERV) PARM(&CRAPRM1. &CRAPRM2. )

CRASUB*
//SYSPROC DD DISP=SHR,DSN=my.exec.library
//SYSTSIN DD *
%myexec ISPSTART PGM(CRASERV) PARM(&CRAPRM1. &CRAPRM2. )
//*
```

---

## (Optional) CARMA user exit

CARMA supports the invocation of a user exit to allow for specialized initialization during startup and specialized cleanup during shutdown of the CARMA server. The usage of a user exit reduces the need to alter the allocation exec, which is maintained by PTF service.

The user exit is invoked by the allocation exec, and is executed twice. The initialization invocation is after the allocation of the temporary data sets and before the CARMA server is invoked. The cleanup invocation is after the CARMA server ended and before the temporary files are removed. If the first invocation ends with return code 99 or higher, CARMA startup is interrupted. This implies that neither CARMA server nor the second invocation of this user exit is executed.

A sample user exit is provided as FEK.SFEKSAMP(CRAEXIT), unless you used a different high-level qualifier during the SMP/E install of Developer for System z. This sample user exit documents in detail the startup arguments passed to the user exit:

Startup argument	Description
(STARTUP)   (ENDING)	Indicator whether the exit invocation is before or after CARMA server invocation.
EXIT_RC=rc	Return code of the previous invocation of the exit.  rc Is always 0 during (STARTUP) invocation.
CARMA_RC=rc	Return code of the invocation of CARMA server.  rc Is always 0 during (STARTUP) invocation.
...	CARMA server startup command and startup arguments. For example ISPSTART PGM(CRASERV) PARM(1312 420 EXIT=CRAEXIT CLIENT=9.0.1 . . . . )

Output generated by the user exit is shown in DD SYSTSPRT of the CARMA server.

---

## (Optional) IRXJCL versus CRAXJCL

If the CARMA server is started using TSO (IKJEFTxx), problems might occur if your RAMs call services which in turn call the IRXJCL REXX batch interface. The problem can occur when the processors called by the RAM previously ran either without TSO, or only in online TSO, and dynamically allocates DD SYSTSIN or SYSTSPRT. A sample program, CRAXJCL, is provided to work around this problem.

Your processor might fail if it attempts to allocate SYSTSIN or SYSTSPRT, which is required for IRXJCL, because batch TSO required for CARMA already has those DD names allocated and open. The CRAXJCL replacement module attempts to allocate SYSTSIN and SYSTSPRT to DUMMY but ignores the errors which occur if the allocations fail. It then calls IRXJCL to do the actual work.

This means that when your processors run in a CARMA environment started by TSO, the allocations to SYSTSIN and SYSTSPRT are the same as those used by CARMA. When the processors are run outside of TSO/CARMA, the SYSTSIN and SYSTSPRT allocations are created by CRAXJCL. Therefore, your processors must not rely on the contents of the data set allocated to SYSTSIN.

It is assumed that calls to IRXJCL use the PARM field to pass the REXX name and startup parameters, as documented in *TSO/E REXX Reference* (SA22-7790). This means that SYSTSIN can safely be used by CARMA. Any output sent to SYSTSPRT by IRXJCL is written in CARMA's log.

Processors that call the CRAXJCL replacement module should not attempt to allocate DD SYSTSIN or SYSTSPRT before calling CRAXJCL.

## Create CRAXJCL

The CRAXJCL replacement module is provided in source format because you must customize it to specify the specific allocations to use for SYSTSPRT. The allocation for SYSTSIN should usually be to a dummy data set.

Sample assembler source code and a sample compile/bind job are available as FEK.#CUST.ASM(CRAXJCL) and FEK.#CUST.JCL(CRA#CIRX), unless you specified a different location when you customized and submitted FEK.SFEKSAMP(FEKSETUP) job. For more details, see “Customization setup” on page 15.

Customize the CRAXJCL assembler source code as needed, using the documentation within the member. Afterward, customize and submit the CRA#CIRX JCL to create the CRAXJCL load module. For customization instructions, see the documentation within the member.

If needed, you can rename IRXJCL to something else. Adjust the CRAXJCL source to call this new name for IRXJCL and compile it, and then rename the CRAXJCL load module to IRXJCL. This setup might be easier than changing all your calls to IRXJCL.

---

## Chapter 4. (Optional) SCLM Developer Toolkit

SCLM Developer Toolkit provides the tools that are needed to extend the capabilities of SCLM to the client. SCLM itself is a host system-based source code manager that is included 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 heritage code development and support for full Java and Java EE development on the workstation with synchronization to SCLM on the mainframe including building, assembling, and deployment of the Java EE code from the mainframe.

---

### Requirements and checklist

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

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

To start using SCLM Developer Toolkit at your site, you must perform the following tasks. Unless otherwise indicated, all tasks are mandatory.

1. Verify and adjust the prerequisites and PARMLIB updates. For details, see “Prerequisites.”
2. Customize Developer for System z configuration files. For details see:
  - “ISPF.conf updates for SCLMDT” on page 90
  - “rsed.envvars updates for SCLMDT” on page 91
3. Optionally define long/short name translation support. For details, see “(Optional) Long/short name translation” on page 91.
4. Optionally install and customize Ant to use the Java EE build support. For details, see “(Optional) Install and customize Ant” on page 94.
5. Update SCLM to define SCLMDT-specific parts. For details, see “SCLM updates for SCLMDT” on page 95.
6. Optionally set up automation to periodically clean up the SCLMDT work area. For details, see “Remove old files from WORKAREA and /tmp” on page 96.

---

### Prerequisites

For a list of required SCLM maintenance, see *IBM Rational Developer for System z Prerequisites* (SC23-7659).

This publication also documents the Ant specifications needed for Java EE builds in SCLM Developer Toolkit.

**Attention:** SCLM Developer Toolkit uses ISPF's TSO/ISPF Client Gateway, which implies that z/OS 1.8 or later is required.

As described in “PARMLIB changes” on page 16, SCLM Developer Toolkit requires additional customization of system settings. These changes include the following items:

- (BPXPRMxx) Increase the maximum number of processes per z/OS UNIX user ID.
- (PROGxx) APF authorize SYS1.LINKLIB and the REXX runtime, REXX.V1R4M0.SEAGLPA or REXX.V1R4M0.SEAGALT.
- (PROGxx/LPALSTxx) Place ISP.SISPLPA, ISP.SISPLOAD, SYS1.LINKLIB and the REXX runtime in LINKLIST/LPALIB.

Also, SCLM Developer Toolkit uses SDSF or the TSO **OUTPUT** command to retrieve job completion status and job output. Both methods require additional attention:

- SDSF must be ordered, installed, and configured separately. SDSF also requires JES2.
- The default settings for the TSO **OUTPUT** command enable a user to retrieve only those job outputs that begin with that specific user ID. To use the **OUTPUT** facility fully, the sample TSO/E exit IKJEFF53 might need to be modified so that a user can retrieve the job output the user owns, but that does not begin that user's user ID. For more information about this exit, see *TSO/E Customization* (SA22-7783).

Users require READ, WRITE, and EXECUTE permission to the z/OS UNIX directories /tmp/ and /var/rdz/WORKAREA/. Directory WORKAREA/ is located in /var/rdz/, unless you specified a different location when you customized and submitted the FEK.SFEKSAMP(FEKSETUP) job. For more details, see “Customization setup” on page 15.

---

## ISPF.conf updates for SCLMDT

SCLM Developer Toolkit uses the standard ISPF/SCLM skeletons, so ensure that the ISP.SISPSLIB skeleton library is allocated to the ISPSLIB concatenation in ISPF.conf. Using the ISP.SISPSENU data set is optional.

ISPF.conf is located in /etc/rdz/, unless you specified a different location when you customized and submitted the FEK.SFEKSAMP(FEKSETUP) job. For more details, see “Customization setup” on page 15. You can edit the file with the TSO **OEDIT** command.

**Note:** Changes are in effect for all clients that connect to the host system after the update.

The following sample code shows the ISPF.conf file, which must be customized to match your system environment. Comment lines start with an asterisk (\*). Add data sets to the concatenation on the same line and separate the names with a comma (,). For more details on customizing ISPF.conf, see “ISPF.conf, the ISPF's TSO/ISPF Client Gateway configuration file” on page 52.

```

* REQUIRED:
sysproc=ISP.SISPCLIB,FEK.SFEKPROC
isplib=ISP.SISPMENU
isptlib=ISP.SISPTEU
ispplib=ISP.SISPPENU
ispslib=ISP.SISPSLIB
ispllib=ISP.SISPLOAD

```

```

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

```

Figure 30. ISPF.conf updates for SCLMDT

**Note:**

- You can add your own DD-like statements and data set concatenations to customize the TSO environment, thus mimicking a TSO logon procedure. For more details, see "Customizing the TSO environment" in the *Host Configuration Reference* (SC14-7290).
- When you are doing batch builds, ensure that the customized version of the FLMLIBS skeleton is concatenated before the ISPF/SCLM skeleton library.  

```
ispslib=h1q.USERSKEL,ISP.SISPSLIB
```

## rased.envvars updates for SCLMDT

SCLM Developer Toolkit uses some directives set in `rased.envvars` to locate data sets and directories.

`rased.envvars` is located in `/etc/rdz/`, unless you specified a different location when you customized and submitted the `FEK.SFEKSAMP` (`FEKSETUP`) job. For more details, see "Customization setup" on page 15. You can edit the file with the TSO **EDIT** command.

**Note:** For the changes to take effect, restart the RSED started task.

The following code sample shows the SCLMDT directives in `rased.envvars`, which must be customized to match your system environment. For more details on customizing `rased.envvars`, see "rased.envvars, the RSE configuration file" on page 35.

```

_SCLMDT_CONF_HOME=/var/rdz/sc1mdt
#STEPLIB=$STEPLIB:FEK.SFEKAUTH:FEK.SFEKLOAD
#_SCLMDT_TRANTABLE=FEK.#CUST.LSTRANS.FILE
#ANT_HOME=/usr/lpp/apache/Ant/apache-ant-1.7.1
_SCLMDT_BASE_HOME=$RSE_HOME
_SCLMDT_WORK_HOME=$CGI_ISPHOME
CGI_DTWORK=$_SCLMDT_WORK_HOME

```

Figure 31. rased.envvars updates for SCLMDT

## (Optional) Long/short name translation

SCLM Developer Toolkit provides the ability to store long name files into SCLM. Long file names are files with names that have more than 8 characters or are in mixed case. Storing of long file names is achieved through the use of a VSAM file that contains the mapping of the long file name to the 8-character member name used in SCLM.

**Note:**

- For versions previous to z/OS 1.8, this facility is provided through a base ISPF/SCLM PTF that addresses APAR OA11426.
- The long/short name translation is also used by other SCLM-related products, such as IBM SCLM Administrator Toolkit.

## **Create LSTRANS.FILE, the long/short name translation VSAM**

To create the long/short name translation VSAM, customize and submit the sample FLM02LST member in the ISP.SISPSAMP ISPF sample library. The configuration steps in this publication require the VSAM to be named FEK.#CUST.LSTRANS.FILE, as shown in the following sample setup JCL.

```

//FLM02LST JOB <job parameters>
//*
/* CAUTION: This is neither a JCL procedure nor a complete job.
/* Before using this sample, you will have to make the following
/* modifications:
/* 1. Change the job parameters to meet your system requirements.
/* 2. Change ***** to the volume that will hold the VSAM.
/* 3. Change all references of FEK.#CUST.LSTRANS.FILE to
/*    match your naming convention for the SCLM translate VSAM.
/*
//CREATE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE FEK.#CUST.LSTRANS.FILE
SET MAXCC=0
DEFINE CLUSTER(NAME(FEK.#CUST.LSTRANS.FILE) -
              VOLUMES(*****)) -
              RECORDSIZE(58 2048) -
              SHAREOPTIONS(3 3) -
              CYLINDERS(1 1) -
              KEYS(8 0) -
              INDEXED) -
DATA (NAME(FEK.#CUST.LSTRANS.FILE.DATA)) -
INDEX (NAME(FEK.#CUST.LSTRANS.FILE.INDEX))

/* DEFINE ALTERNATE INDEX WITH NONUNIQUE KEYS -> ESDS */

DEFINE ALTERNATEINDEX(-
              NAME(FEK.#CUST.LSTRANS.FILE.AIX) -
              RELATE(FEK.#CUST.LSTRANS.FILE) -
              RECORDSIZE(58 2048) -
              VOLUMES(*****)) -
              CYLINDERS(1 1) -
              KEYS(50 8) -
              UPGRADE -
              NONUNIQUEKEY) -
DATA (NAME(FEK.#CUST.LSTRANS.FILE.AIX.DATA)) -
INDEX (NAME(FEK.#CUST.LSTRANS.FILE.AIX.INDEX))

/*
/*
//PRIME EXEC PGM=IDCAMS,COND=(0,LT)
//SYSPRINT DD SYSOUT=*
//INITREC DD *
INITREC1
/*
//SYSIN DD *
REPRO INFILE(INITREC) -
      OUTDATASET(FEK.#CUST.LSTRANS.FILE)
IF LASTCC = 4 THEN SET MAXCC=0

BLDINDEX IDS(FEK.#CUST.LSTRANS.FILE) -
          ODS(FEK.#CUST.LSTRANS.FILE.AIX)

IF LASTCC = 0 THEN -
  DEFINE PATH (NAME(FEK.#CUST.LSTRANS.FILE.PATH) -
              PATHENTRY (FEK.#CUST.LSTRANS.FILE.AIX))
/*

```

Figure 32. FLM02LST: Long/short name translation setup JCL

**Note:** Users need UPDATE authority to this VSAM data set, as described in "Security considerations" in *Host Configuration Reference* (SC14-7290).

## rshed.envvars updates for long/short name translation

Before using the long/short name translation, uncomment and set the rshed.envvars environment variable `_SCLMDT_TRANTABLE` to match the name of the long/short name translation VSAM.

rshed.envvars is located in `/etc/rdz/`, unless you specified a different location when you customized and submitted the FEK.SFEKSAMP (FEKSETUP) job. For more details, see “Customization setup” on page 15. You can edit the file with the TSO **EDIT** command.

**Note:** For the changes to take effect, restart the RSED started task.

---

## (Optional) Install and customize Ant

This step is required only if you plan to use the Java EE build support in SCLM.

Apache Ant is an open source Java build tool and can be downloaded from <http://ant.apache.org/>. Ant consists of text files and scripts, which are distributed in ASCII format and thus require an ASCII/EBCDIC translation to run in z/OS UNIX.

Perform the following steps to implement Ant on z/OS, and to define it to Developer for System z:

- Download, in binary format, the latest Ant compressed file into the z/OS UNIX file system. Download the .zip version of ANT.
- Open a z/OS UNIX command-line session to continue the installation, for example with the **TSO OMVS** command.
- Make a home directory for the Ant installation by using the **mkdir -p /home-dir** command and make it your current directory with the **cd /home-dir** command.
- Use the JAR extract command, **jar -xf apache-ant-1.7.1.zip**, to extract the file to the current directory. A Java bin directory must exist in your local z/OS UNIX PATH to use the **jar** command. Otherwise, fully qualify the command with the Java bin location (for example, `/usr/lpp/java/J6.0/bin/jar -xf apache-ant-1.7.1.zip`).
- Convert all Ant text files to EBCDIC by optionally customizing and executing the `/usr/lpp/rdz/samples/BWBTRANT` sample script.

**Note:** Execute this script only once. Multiple runs will corrupt your Ant install.

- To check for successful translation, locate and open a text file within the ANT directory, such as `apache-ant-1.7.1/README`. If the file is readable, the translation was successful.
- Use the **chmod -R 755 \*** command to enable all users to read and execute files in the ANT directory.
- Before using Ant, set the rshed.envvars environment variables `JAVA_HOME` and `ANT_HOME`.
  - `JAVA_HOME` is required to point to the Java home directory, for example:  
`JAVA_HOME=/usr/lpp/java/IBM/J6.0`
  - `ANT_HOME` is required to point to the Ant home directory, for example:  
`ANT_HOME=/usr/lpp/apache/Ant/apache-ant-1.7.1`

For example:

- **TSO OMVS**

- `mkdir -p /usr/lpp/Apache/Ant`
- `cd /usr/lpp/Apache/Ant`
- `jar -xf /u/userid/apache-ant-1.7.1`
- `/usr/lpp/rdz/samples/BWBTRANT`
- `cat ./apache-ant-1.7.1/README`
- `chmod -R 755 *`
- `oedit /etc/rsed.envvars`

To test that the Ant initialization has been successful:

- Add the Ant and Java bin directories to the environment variable PATH.

Example:

```
export PATH=/usr/lpp/Apache/Ant/apache-ant-1.7.1/bin:$PATH
export PATH=/usr/lpp/java/J6.0/bin:$PATH
```

- To display the version, if successfully installed, execute `ant -version`.

Example:

```
ant -version
```

**Note:** Setting the PATH statement in this way is necessary for testing only, not for operational use.

---

## SCLM updates for SCLMDT

SCLM itself also requires customization to work with SCLM Developer Toolkit. For more information about the required customization tasks, see *IBM Rational Developer for System z SCLM Developer Toolkit Administrator's Guide (SC23-9801)*:

- Define language translators for Java EE support
- Define SCLM types for Java EE support

To complete the customization and project definition tasks, the SCLM administrator must know several Developer for System z customizable values, as described in Table 13.

Table 13. SCLM administrator checklist

Description	<ul style="list-style-type: none"> <li>• Default value</li> <li>• Where to find the answer</li> </ul>	Value
Developer for System z sample library	<ul style="list-style-type: none"> <li>• FEK.SFEKSAMV</li> <li>• SMP/E installation</li> </ul>	
Developer for System z sample directory	<ul style="list-style-type: none"> <li>• /usr/lpp/rdz/samples</li> <li>• SMP/E installation</li> </ul>	
Java bin directory	<ul style="list-style-type: none"> <li>• /usr/lpp/java/J6.0/bin</li> <li>• rsed.envvars - \$JAVA_HOME/bin</li> </ul>	
Ant bin directory	<ul style="list-style-type: none"> <li>• /usr/lpp/Apache/Ant/apache-ant-1.7.1/bin</li> <li>• rsed.envvars - \$ANT_HOME/bin</li> </ul>	
WORKAREA home directory	<ul style="list-style-type: none"> <li>• /var/rdz</li> <li>• rsed.envvars - \$CGI_ISPWORK</li> </ul>	
SCLMDT project configuration home directory	<ul style="list-style-type: none"> <li>• /var/rdz/sclmdt</li> <li>• rsed.envvars - \$_SCLMDT_CONF_HOME</li> </ul>	

Table 13. SCLM administrator checklist (continued)

Description	<ul style="list-style-type: none"> <li>• Default value</li> <li>• Where to find the answer</li> </ul>	Value
Long/short name translation VSAM	<ul style="list-style-type: none"> <li>• FEK.#CUST.LSTRANS.FILE</li> <li>• rsed.envvars - \$_SCLMDT_TRANTABLE</li> </ul>	

---

## Remove old files from WORKAREA and /tmp

SCLM Developer Toolkit and ISPF's TSO/ISPF Client Gateway share the same WORKAREA and /tmp directory, both of which might need a periodic cleanup. For more information about this task, see “(Optional) WORKAREA and /tmp cleanup” on page 124.

---

## Chapter 5. (Optional) Application Deployment Manager (deprecated)

**Note:** The Application Deployment Manager has been marked as deprecated. Although it is still supported, this function will no longer be enhanced.

Developer for System z uses certain functions of Application Deployment Manager as a common deployment approach for various components. The customization steps listed in this chapter are required if any of the following functions are used:

- Enterprise Service Tools
- BMS Screen Designer
- MFS Screen Designer
- CICSTS Code Generation

**Note:** Enterprise Service Tools encompasses multiple tools, such as the Service Flow Modeler (SFM) and XML Services for the Enterprise.

Customizing Application Deployment Manager adds the CICS Resource Definition (CRD) server, which runs as a CICS application on z/OS to support the following functions:

- CICS resource queries
- CICS resource definition install and uninstall requests in both CICSplex SM and non-CICSplex SM environments
- Program and mapset phase-in requests
- Pipeline scan requests
- Manifest export, import, and update requests

CICS administrators can find more information about the CRD server in "CICSTS considerations" in the *Host Configuration Reference* (SC14-7290).

---

### Requirements and checklist

You 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:

- Define a TCP/IP port for external communication
- Update the CICS region JCL
- Update the CICS region CSD
- Define a group to CICS region
- Create a security rule to allow administrators to update to an Application Deployment Manager VSAM
- Set up CICSTS security
- (Optional) Define CICS transaction names
- (Optional) Create a security rule to allow users to update to an Application Deployment Manager VSAM

To start using Application Deployment Manager at your site, perform the following tasks. Unless otherwise indicated, all tasks are mandatory.

1. Create the CRD repository. For details, see “CRD repository.”
2. Choose the CICS interface (RESTful or Web Service) to be used. The interfaces can co-exist. For details, see “RESTful versus Web Service” on page 99.
3. If required, do the REST service-specific customizations. For details, see “CRD server that uses the RESTful interface” on page 99.
  - Define the CRD server to the CICS primary connection region.
  - Optionally, define the CRD server to CICS non-primary connection regions.
  - Optionally, customize the CRD server transaction IDs.
4. If required, do the Web Service-specific customizations. For details, see “CRD server that uses the Web Service interface” on page 100.
  - Add the pipeline message handler to the CICS RPL concatenation.
  - Define the CRD server to the CICS primary connection region.
  - Optionally, define the CRD server to CICS non-primary connection regions.
5. Optionally, create the manifest repository. For details, see “(Optional) Manifest repository” on page 102.

---

## CRD repository

Customize and submit the ADNVCRD job to allocate and initialize the CRD repository VSAM data set. For customization instructions, see the documentation within the member.

ADNVCRD is located in FEK.#CUST.JCL, unless you specified a different location when you customized and submitted the FEK.SFEKSAMP (FEKSETUP) job. For more details, see “Customization setup” on page 15.

Create a separate repository for each CICS primary connection region. Sharing the repository implies that all related CICS regions will use the same values stored in the repository.

### Note:

- An existing CRD server repository must be enlarged to enable the URIMAP support added to the Administrative utility in Developer for System z version 7.6.1. For more details, see “Administrative utility migration notes” in the *Host Configuration Reference* (SC14-7290).
- Unless notified otherwise, the current CRD server repository, which holds the customized values, can be reused across Developer for System z releases.

Users require read access to the CRD repository, CICS administrators require update access.

## CICS administrative utility

Developer for System z provides the administrative utility that enables CICS administrators to provide the default values for CICS resource definitions. These defaults can be read-only, or can be editable by the application developer.

The administrative utility is called by the ADNJSPAU sample job. To use this utility, update access to the CRD repository is required.

ADNJSPAU is located in FEK.#CUST.JCL, unless you specified a different location when you customized and submitted the FEK.SFEKSAMP (FEKSETUP) job. For more details, see “Customization setup” on page 15.

More information is available in "CICSTS considerations" in the *Host Configuration Reference* (SC14-7290).

---

## RESTful versus Web Service

CICS Transaction Server version 4.1 and later have an HTTP interface that is designed by using Representational State Transfer (RESTful) principles. This RESTful interface is now the strategic CICSTS interface for use by client applications. The older Web Service interface has been stabilized, and enhancements is for the RESTful interface only.

Application Deployment Manager follows this statement of direction and requires the RESTful CRD server for all services that are new to Developer for System version 7.6 or later.

The RESTful and Web Service interfaces can be active concurrently in a single CICS region, if needed. In this case, two CRD servers are active in the region. Both servers share the same CRD repository. CICS issues some warnings about duplicate definitions when the second interface is defined to the region.

---

## CRD server that uses the RESTful interface

The information in this section describes how to define the CRD server that uses the RESTful interface to communicate with the Developer for System z client.

The RESTful and Web Service interfaces can be active concurrently in a single CICS region, if needed. In this case, two CRD servers are active in the region. Both servers share the same CRD repository. CICS issues some warnings about duplicate definitions when the second interface is defined to the region.

## CICS primary connection region

The CRD server must be defined to the primary connection region. This is the Web Owning Region (WOR) that processes Web Service requests from Developer for System z.

- Place the FEK.SFEKLOAD(ADNCRD\*, ADNANAL and ADNREST) load modules in the CICS RPL concatenation (DD statement DFHRPL) of the CICS primary connection region. You should do this by adding the installation data set to the concatenation so that applied maintenance is automatically available to CICS.
- Customize and submit the ADNCSDRS job to update the CICS System Definition (CSD) for the CICS primary connection region. For customization instructions, see the documentation within the member.

ADNCSDRS is located in FEK.#CUST.JCL, unless you specified a different location when you customized and submitted the FEK.SFEKSAMP(FEKSETUP) job. For more details, see "Customization setup" on page 15.

- Use the appropriate CEDA command to install the Application Deployment Manager group for this region, for example:  
CEDA INSTALL GROUP(ADNPCRGP)

## CICS non-primary connection regions

The CRD server can also be used with one or more additional non-primary connection regions, which are usually Application Owning Regions (AOR).

**Note:** It is not necessary to perform these steps if CICSplex® SM Business Application Services (BAS) is used to manage your CICS resource definitions.

- Place the FEK.SFEKLOAD(ADNCRD\*) Application Deployment Manager load module in the CICS RPL concatenation (DD statement DFHRPL) of these non-primary connection regions. Do this by adding the installation data set to the concatenation so that applied maintenance is automatically available to CICS.
- Customize and submit the ADNCSDAR job to update the CSD for these non-primary, connection regions. For customization instructions, see the documentation within the member.  
ADNCSDAR is located in FEK.#CUST.JCL, unless you specified a different location when you customized and submitted the FEK.SFEKSAMP(FEKSETUP) job. For more details, see “Customization setup” on page 15.
- Use the appropriate CEDA command to install the Application Deployment Manager group for these regions, for example:  
CEDA INSTALL GROUP(ADNARRGP)

## (Optional) Customize the CRD server transaction IDs

Developer for System z supplies multiple transactions that are used by the CRD server when defining and inquiring CICS resources.

Table 14. Default CRD server transaction IDs

Transaction	Description
ADMS	For requests from the Manifest Processing tool to change CICS resources. Typically, this is intended for CICS administrators.
ADMI	For requests that define, install, or uninstall CICS resources.
ADMR	For all other requests that retrieve CICS environmental or resource information.

You can change the transaction IDs to match your site standards by following these steps:

1. Customize and submit ADNTXNC to create load module ADNRCUST. For customization instructions, see the documentation within the member.
2. Place the resulting ADNRCUST load module in the CICS RPL concatenation (DD statement DFHRPL) of the CICS regions where the CRD server is defined.
3. Customize and submit ADNCSDTX to define ADNRCUST as program to the CICS regions where the CRD server is defined. For customization instructions, see the documentation within the member.

**Note:** The RESTful CRD server always tries to load the ADNRCUST load module. Therefore, you can get a small performance benefit by creating and defining the ADNRCUST load module even if you do not change the transaction IDs.

---

## CRD server that uses the Web Service interface

The information in this section describes how to define the CRD server that uses the Web Service interface to communicate with the Developer for System z client.

The RESTful and Web Service interfaces can be active concurrently in a single CICS region, if needed. In this case, two CRD servers are active in the region. Both servers share the same CRD repository. CICS issues some warnings about duplicate definitions when the second interface is defined to the region.

## Pipeline message handler

The pipeline message handler (ADNTMSGH) is used for security by processing the user ID and password in the SOAP header. ADNTMSGH is referenced by the sample pipeline configuration file and must therefore be placed into the CICS RPL concatenation. To learn more about the pipeline message handler and the required security setup, see "CICSTS considerations" in the *Host Configuration Reference* (SC14-7290).

Developer for System z supplies multiple transactions that are used by the CRD server when defining and inquiring CICS resources. These transaction IDs are set by ADNTMSGH, depending on the requested operation. Sample COBOL source code is provided to allow site-specific customizations to ADNTMSGH:

Table 15. Default CRD server transaction IDs

Transaction	Description
ADMS	For requests from the Manifest Processing tool to change CICS resources. Typically, this is intended for CICS administrators.
ADMI	For requests that define, install or uninstall CICS resources.
ADMR	For all other requests that retrieve CICS environmental or resource information.

Using the default:

- Place the FEK.SFEKLOAD(ADNTMSGH) load module in the CICS RPL concatenation (DD statement DFHRPL) of the CICS primary connection region. Do this by adding the installation data set to the concatenation so that applied maintenance is automatically available to CICS.

Customizing ADNTMSGH:

Sample members ADNMSGH\* are located in FEK.#CUST.JCL and FEK.#CUST.COBOL, unless you specified a different location when you customized and submitted the FEK.SFEKSAMP(FEKSETUP) job. For more details, see "Customization setup" on page 15.

- Customize the sample Pipeline Message Handler (COBOL) source code, FEK.#CUST.COBOL(ADNMSGHS), to match your site's standards.
- Customize and submit the FEK.#CUST.JCL(ADNMSGHC) job to compile the customized ADNMSGHS source. For customization instructions, see the documentation within ADNMSGHC. The resulting load module must be named ADNTMSGH.
- Place the resulting ADNTMSGH load module in the CICS RPL concatenation (DD statement DFHRPL) of the CICS primary connection region.

**Note:** Ensure that the customized ADNTMSGH load module is located before any reference to FEK.SFEKLOAD; otherwise, the default one is used.

## CICS primary connection region

The CRD server must be defined to the primary connection region. This is the region that processes service requests from Developer for System z.

- Place the FEK.SFEKLOAD(ADNCRD\*, ADNANAL and ADNREST) load modules in the CICS RPL concatenation (DD statement DFHRPL) of the CICS primary connection region. Do this by adding the installation data set to the concatenation so that applied maintenance is automatically available to CICS.

The pipeline message handler load module, ADNTMSGH, must also be placed in the RPL concatenation, as described in “Pipeline message handler” on page 101.

- Customize and submit the ADNCSDWS job to update the CICS System Definition (CSD) for the CICS primary connection region. For customization instructions, see the documentation within the member. The transaction IDs used in this job must match the ones that are used by the Pipeline message handler, which might have been customized.

ADNCSDWS is located in FEK.#CUST.JCL, unless you specified a different location when you customized and submitted the FEK.SFEKSAMP(FEKSETUP) job. For more details, see “Customization setup” on page 15.

- Use the appropriate CEDA command to install the Application Deployment Manager group for this region, for example:

```
CEDA INSTALL GROUP(ADNPCRGP)
```

## CICS non-primary connection regions

The CRD server can also be used with one or more additional non-primary connection regions, which are usually Application Owning Regions (AOR).

**Note:** It is not necessary to perform these steps if CICSplex SM Business Application Services (BAS) is used to manage the CICS resource definitions.

- Place the FEK.SFEKLOAD(ADNCRD\*) Application Deployment Manager load modules in the CICS RPL concatenation (DD statement DFHRPL) of these non-primary connection regions. You should do this by adding the installation data set to the concatenation so that applied maintenance is automatically available to CICS.
- Customize and submit the ADNCSDAR job to update the CSD for these non-primary, connection regions. For customization instructions, see the documentation within the member.

ADNCSDAR is located in FEK.#CUST.JCL, unless you specified a different location when you customized and submitted the FEK.SFEKSAMP(FEKSETUP) job. For more details, see “Customization setup” on page 15.

- Use the appropriate CEDA command to install the Application Deployment Manager group for these regions, for example:

```
CEDA INSTALL GROUP(ADNARRGP)
```

---

## (Optional) Manifest repository

Developer for System z allows clients to browse and optionally change manifests that describe selected CICS resources. Depending on permissions set by the CICS administrator, changes can be done directly or exported to the manifest repository for further processing by a CICS administrator.

**Note:**

- This step is required only if manifests are exported from Developer for System z to be processed by the Manifest Processing tool.
- The Manifest Processing tool is a plug-in for IBM CICS Explorer.

To allocate and initialize the manifest repository VSAM data set, and to define it to the CICS primary connection region, customize and submit the ADNVMFST job. For customization instructions, see the documentation within the member. A separate manifest repository must be created for each CICS primary connection region. All users need update access to the manifest repository.

ADNVMFST is located in FEK.#CUST.JCL, unless you specified a different location when you customized and submitted the FEK.SFEKSAMP (FEKSETUP) job. For more details, see “Customization setup” on page 15.



---

## Chapter 6. (Optional) Host-based code analysis

Similar to the Developer for System z client, the Developer for System z host supports running code analysis tools, which are provided as a separate product, Rational Developer for System z Host Utilities. A benefit of doing code analysis on the host is that it can be integrated in your daily batch processing.

The following code analysis tools are available on the host:

- Code review: Using rules with different severity levels, code review scans source code and reports rule violations.
- Code coverage: Analyze a running program and generate a report of lines that are executed, compared to the total number of executable lines.

---

### Requirements and checklist

You do not need assistance of other administrators to start using host-based code analysis tools at your site, but you must perform the following tasks. Unless otherwise indicated, all tasks are mandatory.

1. Install Rational Developer for System z Host Utilities, as documented in *Program Directory for IBM Rational Developer for System z Host Utilities* (GI13-2864). When using the provided defaults, the product is installed using high-level qualifier AKG and z/OS UNIX path `/usr/lpp/rdzutil`.
2. Create customizable copies of the provided samples by customizing and submitting `AKG.SAKGSAMP(AKGSETUP)`. This job performs the following tasks:
  - Create `AKG.#CUST.PROCLIB` and populate it with sample `SYS1.PROCLIB` members.
  - Create `AKG.#CUST.JCL` and populate it with sample configuration `JCL`.

---

### Code review

Code review scans source code and reports rule violations, using rules with different severity levels. The tool comes with rule providers for Cobol and PL/I, but other rule providers can be added.

Developer for System z Host Utilities provides a sample procedure, `AKGCR`, to simplify the calling of code review services in batch mode. `AKGCR` is found in `AKG.#CUST.PROCLIB`, unless you specified a different location when you customized and submitted the `AKG.SAKGSAMP(AKGSETUP)` job.

Customize the sample procedure, `AKG.#CUST.PROCLIB(AKGCR)`, as described within the member, and copy it to `SYS1.PROCLIB`.

If the `AKGCR` procedure 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 their calling job.

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

## Modify code review processing

Developer for System z Code Review allows for third-party code to be part of the review process. For example, you can provide a rule provider to analyze C/C++ code, or you can enhance the Cobol rule provider to recognize site-specific coding conventions.

Host-based code review is an Eclipse process, just like the Developer for System z client. Therefore, the enhancements done by your development support team for code review on the client can be reused on the host.

The enhancements will consist of Eclipse plugins or Eclipse features. In order to activate them, you must make them available to the existing code, as documented in the AKGCRADD configuration job. AKGCRADD is in AKG.#CUST.JCL, unless you specified a different location when you customized and submitted the AKG.SAKGSAMP(AKGSETUP) job.

---

## Code coverage

| Code coverage analyzes a running program and generates a report of lines that are  
| executed, compared to the total number of executable lines. Note that code  
| coverage sets up a TCP/IP connection, using an ephemeral port, with the  
| Integrated Debugger to gather the required data. Optionally, IBM Debug Tool for  
| z/OS can be used instead of the Integrated Debugger.

Developer for System z Host Utilities provides a sample procedure, AKGCC, to simplify the calling of code coverage services in batch mode. AKGCC is in AKG.#CUST.PROCLIB, unless you specified a different location when you customized and submitted the AKG.SAKGSAMP(AKGSETUP) job.

Customize the sample procedure, AKG.#CUST.PROCLIB(AKGCC), as described within the member, and copy it to SYS1.PROCLIB.

If the AKGCC procedure 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 their calling job.

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

## Code coverage output

The output of code coverage is intended to be imported into a Developer for System z client, and is therefore written to a z/OS UNIX file. Code coverage is also able to use the results of a previous run and combine them with the results of the current run, resulting in a single report that covers multiple code paths.

For these reasons, Developer for System z does not attempt to remove the output of a code coverage run, and the output will thus accumulate over time.

z/OS UNIX provides a shell script, skulker, that deletes files based on the directory they are in and their age. Combined with the z/OS UNIX cron daemon, which runs commands at specified dates and times, you can set up an automated tool that periodically cleans out targeted directories. Refer to *UNIX System Services Command Reference* (SA22-7802) for more information about the skulker script and the cron daemon.

---

## Chapter 7. (Optional) Other customization tasks

This section combines various optional customization tasks. To configure the required service, follow the instructions in the appropriate section.

Customizations to Developer for System z configuration files:

- “(Optional) pushtoclient.properties, the host-based client control”
- “(Optional) ssl.properties, the RSE SSL encryption” on page 110
- “(Optional) rsecomm.properties, the RSE tracing” on page 112
- “(Optional) include.conf, Forced includes for C/C++ content assist” on page 113

Developer for System z related customizations to or for other products:

- “(Optional) z/OS UNIX subprojects” on page 115
- “(Optional) Include preprocessor support” on page 115
- “(Optional) xUnit support for Enterprise COBOL and PL/I” on page 116
- “(Optional) Enterprise Service Tools support” on page 117
- “(Optional) CICS bidirectional language support” on page 118
- “(Optional) Diagnostic IRZ messages for the generated code” on page 118
- “(Optional) Integrated debugger” on page 120
- “(Optional) Problem Determination Tools support” on page 122
- “(Optional) DB2 and IMS debug support” on page 123
- “(Optional) File Manager support” on page 123
- “(Optional) WORKAREA and /tmp cleanup” on page 124

---

### (Optional) pushtoclient.properties, the host-based client control

This customization task does not require assistance, special resources, or special customization tasks for a basic setup.

If you enable group support, you need the assistance of a security administrator or an LDAP administrator to complete this customization task, which requires the following resources or special customization tasks:

- Security rule to allow users access to FEK.PTC.\* profiles
- Or define user membership of FEK.PTC.\* LDAP groups

---

Developer for System z clients version 8.0.1 and later can pull client configuration files and product update information from the host system 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 system and propagated to the client on an each-user basis. These host-based projects look and function exactly like the projects that are defined on the client except that their structure, members, and properties cannot be modified by the client, and they are accessible only when connected to the host system.

`pushtoclient.properties` tells the client if these functions are enabled, and where the related data is stored. The data is maintained by a Developer for System z client administrator or a development project manager.

`pushtoclient.properties` is located in `/etc/rdz/`, unless you specified a different location when you customized and submitted the FEK.SFEKSAMP(FEKSETUP) job. For more details, see “Customization setup” on page 15. You can edit the file with the TSO **OEDIT** command. For the changes to take effect, restart the RSED started task.

Since version 8.0.3, the client administrator can create multiple client configuration sets and multiple client update scenarios to fit the needs of different developer groups. These multiple sets and scenarios can be used to provide users with a customized setup, based on criteria such as membership of an LDAP group or permit to a security profile. For more information about supporting multiple groups, see “Push-to-client considerations” in *Host Configuration Reference* (SC14-7290).

The following code sample shows the `pushtoclient.properties` file, which must be customized to match your system environment. Comment lines start with a number sign (#) when using a US code page. Data lines can have only a directive and its assigned value. Comments are not allowed on the same line. Line continuations are not supported.

```
#
# host-based client control
#
config.enabled=false
product.enabled=false
reject.config.updates=false
reject.product.updates=false
accept.product.license=false
primary.system=false
pushtoclient.folder=/var/rdz/pushtoclient
default.store=com.ibm.ftt.configurations.USS
file.permission=RWX.RWX.RX
```

Figure 33. `pushtoclient.properties`: Host-based client control configuration file

#### **config.enabled**

Indicates whether host-based client control is used for configuration files. The default is `false`. The valid values are `true`, `false`, `LDAP`, or `SAF`. For the meaning of these values, see Table 16 on page 109.

#### **product.enabled**

Indicates whether host-based client control is used for product updates. The default is `false`. The valid values are `true`, `false`, `LDAP`, or `SAF`. For the meaning of these values, see Table 16 on page 109.

#### **reject.config.updates**

Indicates whether a user can reject configuration updates that are pushed to the client. The default is `false`. The valid values are `true`, `false`, `LDAP`, or `SAF`. For the meaning of these values, see Table 16 on page 109.

#### **reject.product.updates**

Indicates whether a user can reject product updates that are pushed to the client. The default is `false`. The valid values are `true`, `false`, `LDAP`, or `SAF`. For the meaning of these values, see Table 16 on page 109.

#### **accept.product.license**

Indicates whether the product license is automatically accepted during

updates that are initiated by push-to-client. If enabled, IBM Installation Manager does not ask to accept the license during client update. The default is false. The only valid values are true and false.

#### **primary.system**

Host-based client control supports storing system-specific data for each system, while maintaining common data on a single system to reduce management effort. This directive indicates whether this is the system that stores global, non-system specific, client definitions. The default is false. The only valid values are true and false.

**Note:** Ensure that you have one, and only one, system that is defined as primary system. Developer for System z client administrators cannot export global configuration data unless the target system is a primary system. Developer for System z clients might show erratic behavior when connecting to multiple primary systems with out-of-sync configurations.

#### **pushtoclient.folder**

The base directory for the host-based client control definitions. The default is `/var/rdz/pushtoclient`.

#### **default.store**

Host-based client control supports different methods for storing the data that is pushed to the client. This directive identifies the driver, or store, that is used to access the data. The default is `com.ibm.ftt.configurations.USS`, which supports the data being stored in z/OS UNIX flat files.

Developer for System z only provides the `com.ibm.ftt.configurations.USS` store. A third-party store is needed when the data is located somewhere else.

#### **file.permission**

The `com.ibm.ftt.configurations.USS` store uses `file.permission` to determine the required access permissions for files that are created by the store. The default is `RWX.RWX.RX`, which allows the owner and the owner's default group read and write access to the directory structure and the files within. Everyone else has only read access to the directory structure and the files within.

According to the UNIX standards, permissions can be set for three types of users: owner, group, and other. The fields in the `file.permission` mask match this order, and the fields are separated by a period (.). Each field can either be empty, or have R, W, RW, X, RX, WX, or RWX as value (where R = read, W = write, X = execute or list directory content).

*Table 16. Push-to-client group support*

Key value	Is the related push-to-client function enabled?
False	No, disabled
True	Yes, enabled for all
LDAP	Yes, but availability is controlled by membership of LDAP groups
SAF	Yes, but availability is controlled by permit to security profiles

#### **Note:**

- To activate host-based client control, a `keymapping.xml` file must exist in `/var/rdz/pushtoclient`. This file is created and maintained by a Developer for System z client administrator.

- For more information about host-based projects, host-based client configuration, and upgrade control, see “Push-to-client considerations” in the *Host Configuration Reference* (SC14-7290).
- When a file is created, z/OS UNIX uses by default the effective UID (user ID) of the creating thread and the GID (group ID) of the owning directory, not the effective GID of the creating thread. For more information on how to change this behavior or how to adjust your host-based client control setup to get the required GID assignment, see “z/OS UNIX directory structure” in *Host Configuration Reference* (SC14-7290).

---

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

You 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 (Secure Socket Layer). This feature is disabled by default and is controlled by the settings in `ssl.properties`.

**Note:** Client authentication with an X.509 certificate requires the use of SSL encrypted communication.

`ssl.properties` is located in `/etc/rdz/`, unless you specified a different location when you customized and submitted the FEK.SFEKSAMP (FEKSETUP) job. For more details, see “Customization setup” on page 15. You can edit the file with the TSO **EDIT** command. RSE must be restarted for the changes to take effect.

The client communicates with RSE daemon during connection setup and with RSE server during the actual session. Both data streams are encrypted when SSL is enabled.

RSE daemon and RSE server support different mechanisms to store certificates due to architectural differences between the two. This implies that SSL definitions are required for both RSE daemon and RSE server. A shared certificate can be used if RSE daemon and RSE server use the same certificate management method.

*Table 17. SSL certificate storage mechanisms*

Certificate storage	Created and managed by	RSE daemon	RSE server
Key ring	SAF-compliant security product	Supported	Supported
Key database	z/OS UNIX's gskkyman	Supported	/
Key store	Java's keytool	/	Supported

**Note:**

- SAF-compliant key rings are the preferred method for managing certificates.

- SAF-compliant key rings can store the certificate's private key either in the security database or by using ICSF, the interface to System z cryptographic hardware. Access to ICSF is protected by profiles in the CSFSERV security class.

RSE daemon uses System SSL functions to manage SSL. This implies that SYS1.SIEALNKE must be program controlled by the security software and be available to RSE when using LINKLIST or the STEPLIB directive in rsed.envvars.

The following code sample shows the sample `ssl.properties` file, which must be customized to match your system environment. Comment lines start with a number sign (#) when using a US code page. Data lines can have only a directive and its assigned value; comments are not allowed on the same line. Line continuations are not supported.

```
# ssl.properties - SSL configuration file
enable_ssl=false

# Daemon Properties

#daemon_keydb_file=
#daemon_keydb_password=
#daemon_key_label=

# Server Properties

#server_keystore_file=
#server_keystore_password=
#server_keystore_label=
#server_keystore_type=JCERACFKS
```

*Figure 34. ssl.properties – SSL configuration file*

The daemon and server properties must be set only if you enable SSL. For more information about SSL setup, see "Setting up SSL and X.509 authentication" in the *Developer for System z Host Configuration Reference*.

#### **enable\_ssl**

Enable or disable SSL communication. The default is false. The only valid options are true and false.

#### **daemon\_keydb\_file**

RACF or similar security product key ring name. Provide the key database name if you used **gskkyman** to create a key database instead of using a key ring. If SSL is enabled, uncomment and customize this directive.

#### **daemon\_keydb\_password**

Leave commented out or blank if you use a key ring. Otherwise, provide the key database password. If SSL is enabled and you are using a **gskkyman** key database, uncomment and customize this directive.

#### **daemon\_key\_label**

The certificate label used in the key ring or key database, if it is not defined as the default one. Must be commented out if the default is used. If SSL is enabled and you are not using the default security certificate, uncomment and customize this directive. Key labels are case-sensitive.

#### **server\_keystore\_file**

Name of the key store created by Java's **keytool** command, or the RACF or similar security product key ring name if `server_keystore_type=JCERACFKS`. If SSL is enabled, uncomment and customize this directive.

**server\_keystore\_password**

Leave commented out or blank if you use a key ring. Otherwise, provide the key store password. If SSL is enabled and you are using a **keytool** key store, uncomment and customize this directive.

**server\_keystore\_label**

The certificate label used in the key ring or key store. The default is the first valid certificate encountered. If SSL is enabled and you are not using the default security certificate, uncomment and customize this directive. Key labels are case-sensitive.

**server\_keystore\_type**

Key store type. The default is JKS. Valid values are these:

*Table 18. Valid keystore types*

Keyword	Key store type
JKS	Java key store
JCERACFKS	SAF-compliant key ring, where the certificate's private key is stored in the security database.
JCECCARACFKS	SAF-compliant key ring, where the certificate's private key is stored using ICSE, the interface to System z cryptographic hardware.

**Note:** At the time of publication, IBM z/OS Java requires an update of the `/usr/lpp/java/J6.0/lib/security/java.security` file to support JCECCARACFKS. The following line must be added:

```
security.provider.1=com.ibm.crypto.hdwrCCA.provider.IBMJCECCA
```

The resulting file looks like this:

```
security.provider.1=com.ibm.crypto.hdwrCCA.provider.IBMJCECCA
security.provider.2=com.ibm.jsse2.IBMJSSEProvider2
security.provider.3=com.ibm.crypto.provider.IBMJCE
security.provider.4=com.ibm.security.jgss.IBMJGSSProvider
security.provider.5=com.ibm.security.cert.IBMCertPath
security.provider.6=com.ibm.security.sasl.IBMSASL
```

---

## (Optional) rsecomm.properties, the 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 required initial detail level in the output logs.

**Attention:** Changing these settings can cause performance degradations and should be done only under the direction of the IBM support center.

rsecomm.properties is located in /etc/rdz/, unless you specified a different location when you customized and submitted FEK.SFEKSAMP(FEKSETUP) job. For more details, see “Customization setup” on page 15. You can edit the file with the TSO **OEDIT** command.

The following code sample shows the rsecomm.properties file, which can be customized to match your tracing needs. Comment lines start with a number sign (#) when using a US code page. Data lines can have only a directive and its assigned value; comments are not allowed on the same line. Line continuations are not supported.

```
# server.version - DO NOT MODIFY!
server.version=5.0.0

# Logging level
# 0 - Log error messages
# 1 - Log error and warning messages
# 2 - Log error, warning and info messages
debug_level=1
#USER=userid
#USER=(userid,userid,...)
```

Figure 35. rsecomm.properties – Logging configuration file

#### server.version

Logging server version. The default is 5.0.0. Do not modify.

#### debug\_level

Detail level for output logs. The default is 1, which logs error and warning messages. The debug\_level detail controls the detail level of multiple services and, thus, multiple output files. Increasing the detail level will cause performance degradations and should be done under only the direction of the IBM support center. For more information about which logs are controlled by this directive, see “RSE tracing” in the *Host Configuration Reference* (SC14-7290).

The valid values are these:

0	Log error messages only.
1	Log error and warning messages.
2	Log error, warning, and informational messages.

**Note:** debug\_level can be changed dynamically for specific log files with the **modify rsecommlog**, **modify rseserverlog**, and **modify rsedaemonlog** operator commands, as described in Chapter 11, “Operator commands,” on page 177.

**USER** Set debug level 2 (log error, warning, and informational messages) for the specified user IDs during server startup. The debug level for all other users is the default as specified in the debug\_level directive. The USER directive alters the trace detail level for RSE server (rsecomm.log) and the MVS data set services (lock.log and ffs\*.log), and is equivalent to issuing the **modify trace user** operator command.

---

## (Optional) include.conf, Forced includes for C/C++ content assist

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

---

Content assist for C/C++ can use the definitions in `include.conf` to do forced includes of specified files or members. A forced include consists of a file or directory, data set, or data set member which will be parsed when a content assist operation is performed, regardless of whether that file or member was included in the source code using a pre-processor directive.

The file must be referenced in `rzed.envvars` by the `include.c` or `include.cpp` variables before it is used. This reference in `rzed.envvars` implies that you can specify a different file for usage by C and C++. The variables in `rzed.envvars` are disabled by default.

The sample `include.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 15 for more details. You can edit the file with the TSO **OEDIT** command.

Definitions must start in column 1. Comment lines start with a pound sign (#) when using a US code page. Data lines can only have the name of a directory, file, data set or member. Comments are not allowed on the same line. Line continuations are not supported.

```
# To include the stdio.h file from the /usr/include directory, input:
# /usr/include/stdio.h
#
# To include all files of the /usr/include directory and all of it's
# sub-directories, input:
# /usr/include
#
# Uncomment and customize variable FILETYPES to limit the z/OS UNIX
# wildcard include to selected (case sensitive) file types:
# The file types are specified in a comma-delimited list (no blanks)
# FILETYPES=H,h,hpp,C,c,cpp,cxx

# To include all members of the CBC.SCLBH.H data set, input:
# //CBC.SCLBH.H
#
# To include the STDIOSTR member of the CBC.SCLBH.H data set, input:
# //CBC.SCLBH.H(STDIOSTR)
# The sample list contains some commonly used C standard library files
/usr/include/assert.h
/usr/include/ctype.h
/usr/include/errno.h
/usr/include/float.h
/usr/include/limits.h
/usr/include/locale.h
/usr/include/math.h
/usr/include/setjmp.h
/usr/include/signal.h
/usr/include/stdarg.h
/usr/include/stddef.h
/usr/include/stdio.h
/usr/include/stdlib.h
/usr/include/string.h
/usr/include/time.h
```

*Figure 36. include.conf - Forced includes for C/C++ content assist*

---

## (Optional) z/OS UNIX subprojects

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

---

REXEC (Remote Execution) is a TCP/IP service that enables clients to execute a command on the host system. SSH (Secure Shell) is a similar service, but 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.

### Note:

- Developer for System z uses the z/OS UNIX version of REXEC, not the TSO version.
- If REXEC/SSH is not configured to use the default port, the Developer for System z client must define the correct port for use by z/OS UNIX subprojects. This configuration can be done by selecting the **Window > Preferences > z/OS Solutions > USS Subprojects > Remote Action Options** preference page. To know which port is used, see “REXEC or SSH setup.”

## REXEC or SSH setup

REXEC and SSH rely on services provided by INETD (Internet Daemon), which is another TCP/IP service. *Communications Server IP Configuration Guide* (SC31-8775) describes the steps required to set up INETD, REXEC, and SSH. For more details and alternate setup methods, see the white paper *Using INETD, REXEC and SSH with Rational Developer for System z* (SC14-7301), available in the Developer for System z library, <http://www-01.ibm.com/support/docview.wss?uid=swg27038517>.

A common port used by REXEC is 512. To verify the port being used, check `/etc/inetd.conf` and `/etc/services`.

- Find the service name (1st word, `exec` in this example) of the `rexecd` server (7th word) in `/etc/inetd.conf`.

```
exec stream tcp nowait OMVSKERN /usr/sbin/orexecd rexecd -LV
```

- Find the port (2nd word, `512` in this example) attached to this service name (1st word) in `/etc/services`.

```
exec      512/tcp      #REXEC    Command Server
```

The same principle applies to SSH. Its common port is 22, and the server name is `sshd`.

---

## (Optional) Include preprocessor support

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

---

Developer for System z supports the interpreting and expanding COBOL and PL/I include statements, including select third-party include statements. Developer for System z also provides a sample REXX `exec`, `FEKRNPLI`, that can be called by the Developer for System z client to expand PL/I source by invoking the PL/I compiler.

FEKRNPLI is located in FEK.#CUST.CNTL, unless you specified a different location when you customized and submitted the FEK.SFEKSAMP(FEKSETUP) job. For more details, see Chapter 2, “Basic customization,” on page 15.

Customize the sample FEK.#CUST.CNTL(FEKRNPLI) exec, as described within the member. You must provide the following information:

- `compiler_hlq`: The high-level qualifier for the PL/I compiler

The Developer for System z client uses the TSO Command Service to execute the exec. This implies that if the FEKRNPLI exec is placed in the SYSPROC or SYSEXEC concatenation for the TSO Command Service, the user does not need to know the exact location of the exec. The user only needs to know the name. By default, the TSO Command Service uses the ISPF Client Gateway to create a TSO environment, but APPC is also supported, as documented in the white paper, *Using APPC to provide TSO command services* (SC14-7291). When using the ISPF Client Gateway, the SYSPROC or SYSEXEC concatenation is defined in ISPF.conf. For more details on customizing this file, see “ISPF.conf, the ISPF’s TSO/ISPF Client Gateway configuration file” on page 52.

---

## (Optional) xUnit support for Enterprise COBOL and PL/I

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

- LINKLIST update
- 

Frameworks that assist developers in writing code to perform repeatable, self-checking unit tests are collectively known as xUnit. Developer for System z provides such a framework for unit testing of Enterprise COBOL and PL/I code, called zUnit.

To use the zUnit framework, developers need access to the AZU\* and IAZU\* load modules in the FEK.SFEKLOAD load library, either through STEPLIB or LINKLIST. The zUnit test runner, AZUTSTRN, in turn needs access to various system libraries, either through STEPLIB or LINKLIST:

- CEE.SCEERUN and CEE.SCEERUN2 (LE runtime)
- SYS1.CSSLIB (callable system services)
- SYS1.SIXML0D1 (XML toolkit)

The zUnit test runner also needs access to a load library that holds the different test cases. This library is likely to be unique to a developer.

The zUnit test runner, AZUTSTRN, can be called by the Developer for System z client in batch mode, from the TSO command line, and from the z/OS UNIX command line.

- Developer for System z provides a sample procedure, AZUZUNIT, to simplify the calling of the zUnit test runner in batch mode. AZUZUNIT is located in FEK.#CUST.PROCLIB, unless you specified a different location when you customized and submitted the FEK.SFEKSAMP(FEKSETUP) job. For more details, see Chapter 2, “Basic customization,” on page 15.

Customize the sample procedure, FEK.#CUST.PROCLIB(AZUZUNIT), as described within the member, and copy it to SYS1.PROCLIB.

The name of the procedure and the names of the steps in the procedure match the default properties that are included with the Developer for System z client. If

the name of a procedure or the name of a step in a procedure is changed, the corresponding properties file on all of the clients must be updated. You should not change the procedure and step names.

If the AZUZUNIT procedure 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 their calling job.

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

- For calling the zUnit test runner from z/OS UNIX (using the /usr/lpp/rdz/bin/zunit script), you can specify the required non-LINKLIST data sets in the STEPLIB directive of rsed.envvars, thus simplifying the setup for the developer.

rsed.envvars is located in /etc/rdz/, unless you specified a different location when you customized and submitted FEK.SFEKSAMP (FEKSETUP) job. For more details, see “Customization setup” on page 15. You can edit the file with the TSO **EDIT** command.

The *zunit* script allows the user to specify data sets that will be added to the STEPLIB directive used by the script.

- For calling the zUnit test runner from the TSO command line by using the FEK.SFEKPROC (FEKZUNIT) exec, the system libraries must exist in LINKLIST. If they do not, developers must specify the system data set names on every call instance of the zUnit test runner. You can also write a wrapper exec that does the **TSOLIB** allocations of these data sets for them. You can use FEKZUNIT itself as an example of how to code this wrapper exec.

The zUnit test runner allows for automatic reformatting of test reports. Developer for System z provides sample conversions (for example, conversion to Ant or jUnit format), which are located in /usr/lpp/rdz/samples/zunit/xsd and /usr/lpp/rdz/samples/zunit/xsl, if you installed Developer for System z in the default /usr/lpp/rdz location.

---

## (Optional) Enterprise Service Tools support

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

---

The Developer for System z client has a code generation component called Enterprise Service Tools. Depending on the type of code being generated, this code relies on functions provided by the Developer for System z host system installation. Making these host system functions available is described in the following sections:

- Chapter 5, “(Optional) Application Deployment Manager (deprecated),” on page 97
- “(Optional) CICS bidirectional language support” on page 118
- “(Optional) Diagnostic IRZ messages for the generated code” on page 118

**Note:** Enterprise Service Tools encompasses multiple tools, such as the Service Flow Modeler (SFM) and XML Services for the Enterprise.

---

## (Optional) CICS bidirectional language support

You 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 component supports different formats of Arabic and Hebrew interface messages, and bidirectional data presentation and editing in all editors and views. In terminal applications, both left-to-right and right-to-left screens are supported, and numeric fields and fields with opposite-to-screen orientation.

Additional bidirectional features and functionality include the following:

- The Enterprise Service Tools service requestor dynamically specifies bidirectional attributes of interface messages.
- Bidirectional data processing in service flows is based on bidirectional attributes such as 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.
- Enterprise Service Tools-generated runtime code includes conversion of data between fields in messages that have different bidirectional attributes.

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

To activate CICS Bidirectional language support, perform the following tasks:

1. Place the FEK.SFEKLOAD load modules FEJBDCMP and FEJBDTRX in the CICS RPL concatenation (DD statement DFHRPL). You should do this by adding the installation data set to the concatenation so that applied maintenance is automatically available to CICS.

**Important:** If you do not concatenate the installation data set but copy the modules into a new or existing data set, keep in mind that those modules are DLLs and must reside in a PDSE library.

2. Define FEJBDCMP and FEJBDTRX as programs to CICS by using the appropriate CEDA command, for example:

```
CEDA DEF PROG(FEJBDCMP) LANG(LE) G(XXX)
CEDA DEF PROG(FEJBDTRX) LANG(LE) G(XXX)
```

---

## (Optional) Diagnostic IRZ messages for the generated code

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

- LINKLIST update
  - CICS region JCL update
-

The Developer for System z client has a code generation component called Enterprise Service Tools. For the code generated by Enterprise Service Tools to issue diagnostic error messages, all IRZM\* and IIRZ\* modules in the FEK.SFEKLMOD load library must be made available to the generated code. Enterprise Service Tools can generate code for the following environments:

- CICS
- IMS
- MVS batch

When the generated code is executed in a CICS transaction, add all IRZM\* and IIRZ\* modules in FEK.SFEKLMOD to the DFHRPL DD of the CICS region. You should do this by adding the installation data set to the concatenation so that applied maintenance is automatically available.

In all other situations, make all IRZM\* and IIRZ\* modules in FEK.SFEKLMOD available either through STEPLIB or LINKLIST. You should do this by adding the installation data set to the concatenation so that applied maintenance is automatically available.

If you use STEPLIB, define the modules not available through LINKLIST in the STEPLIB directive of the task that executes the code.

If the load modules are not available and the generated code encounters an error, the following message is issued:

```
IRZ9999S Failed to retrieve the text of a Language Environment runtime
message. Check that the Language Environment runtime message module for
facility IRZ is installed in DFHRPL or STEPLIB.
```

**Note:**

- Module FEK.SFEKLMOD(IRZPWSIO) is statically linked during top-down IMS MPP code generation. Therefore, the module must not be available during run time of the generated code. It should be available only during compile time.
- In version 8.5, the IRZ\* and IIRZ\* load modules and diagnostic messages have moved from the FEK.SFEKLOAD load library to FEK.SFEKLMOD.
- In version 9.0.1, FEK.SFEKLMOD(IRZPWSIO) and the related FEK.SFEKSAMP(IRZPWSH) sample PL/I include member moved from Developer for System z to IMS Version 12. The parts are renamed to IMS.SDFSRESL(DFSPWSIO) and IMS.SDFSSMPL(DFSPWSH), respectively.

|  
|  
|  
|

---

## (Optional) Integrated debugger

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

- LINKLIST update
  - APF authorization
  - Define started task
  - Define security profiles and access lists
  - Reserve TCP/IP ports for client-host and host confined communication
  - (Optional) Add an SVC (requires IPL)
  - (Optional) LPA updated for SVC
  - (Optional) Update CICS region JCL
  - (Optional) Update CICS CSD
- 

The Developer for System z Integrated Debugger host component allows version 9.0.1 and higher clients to debug various Language Environment (LE) based applications, including CICS transactions loaded into read-only memory.

See the “Integrated Debugger” section of the “Understanding Developer for System z” chapter in the *Host Configuration Reference (SC14-7290)* for an overview of the Integrated Debugger data flow.

To start using the Integrated Debugger at your site, you must perform the following tasks. Unless otherwise indicated, all tasks are mandatory.

1. Integrated Debugger requires that the optional DBGMGR started task is active (together with the mandatory RSED started task). See “DBGMGR, Debug manager started task” on page 24 for the DBGMGR startup JCL.
2. Integrated Debugger configuration is managed by startup arguments of the DBGMGR started task. See “Integrated debugger configuration parameters” on page 121 for details.
3. The DBGMGR started task requires that the FEK.SFEKAUTH library is APF-authorized. See “Integrated debugger parmlib updates” on page 121 for details.
4. The Integrated Debugger must be accessible in your application and requires STEPLIB or LINKLIST updates. See “Integrated debugger parmlib updates” on page 121 for details.
5. Integrated Debugger requires that the user ID of the application being debugged has a valid OMVS segment. See “Integrated debugger security updates” on page 122 for details.
6. The DBGMGR started task requires some security permits. See “Integrated debugger security updates” on page 122 for details.

The following steps are only required for debugging CICS transactions:

1. The Integrated Debugger is capable of debugging CICS transactions. This requires that the Integrated Debugger is defined to CICS. See “Integrated debugger CICS updates” on page 122 for details.
2. (Optional) The Integrated Debugger is capable of debugging CICS transactions loaded into read-only memory. This requires that a Developer for System z supervisor call (SVC) is defined to your system. The related load module must be loaded in LPA at IPL time. See “Integrated debugger parmlib updates” on page 121 for details.

The SVC requires that users are permitted to a security profile if used in a problem-state (non-authorized) environment. See “Integrated debugger security updates” on page 122 for details.

Note that only one Language Environment (LE) based debugger, such as the Integrated Debugger, can be active in a given application or CICS region. A good indication that a debugger is an LE-based debugger is that it provides a CEEEVDBG load module or alias that must be available to the application.

## Integrated debugger configuration parameters

The Integrated Debugger allows for configuring the following variables in the DBGMR startup JCL. See “DBGMR, Debug manager started task” on page 24 for the DBGMR startup JCL.

- The time-zone offset, default EST5DST
- The port used for external (client-host) communication, default 5335
- The port used for internal (host-confined) communication, default 5336
- The SVC number used for debugging read-only CICS transactions, default 251
- The high-level qualifier of the load library, default FEK

## Integrated debugger parmlib updates

- The DBGMR started task requires that the FEK.SFEKAUTH library is APF-authorized. See “APF authorizations in PROGxx” on page 19 for details.
- Language Environment (LE) must be able to invoke the Integrated Debugger. Therefore, the FEK.SFEKAUTH library must be placed either in LINKLIST, or in STEPLIB of the application to be debugged. See “LINKLIST definitions in PROGxx” on page 20 for details.

**Note:** When using LINKLIST, ensure that FEK.SFEKAUTH is located before the libraries of other LE-based debuggers holding the CEEEVDBG or CEEEV006 load modules. For example, IBM Debug Tool for z/OS uses hlq.SEQA\* libraries.

- The Integrated Debugger uses the z/OS Binder API. This API is available since z/OS 1.10 as /usr/lib/iwbndd.so, and since z/OS 1.13 also as SYS1.SIEAMIGE(IEWBND). This implies that for z/OS 1.13 and higher, SYS1.SIEAMIGE should be in LINKLIST (or STEPLIB). See “Requisite LINKLIST and LPA definitions” on page 22 for details.

**Note:** If SYS1.SIEAMIGE is not in LINKLIST or STEPLIB on z/OS 1.13 and higher systems, the Integrated Debugger will issue the following message and attempt to use /usr/lib/iwbndd.so:

```
CEE3501S The module //IEWBND was not found
```

- The Integrated Debugger is capable of debugging CICS transactions loaded into read-only memory. This requires that a Developer for System z supervisor call (SVC) is defined to your system. The default SVC number is 251. The related load module, FEK.SFEKLPA(AQESVC01), must be loaded in LPA at IPL time. See “SVC definitions in IEASVCxx” on page 18 and “LPA definitions in LPALSTxx” on page 19 for details.

## Integrated debugger TCP/IP updates

The Integrated Debugger uses 2 TCP/IP ports. Refer to “TCP/IP considerations” in the *Host Configuration Reference Guide* (SC14-7290) for details.

- Port for client-host communication (default 5335). Communication on this port can be encrypted.

- Port for host-confined communication (default 5336).

## Integrated debugger security updates

The Integrated Debugger requires the following security definitions. See “Security definitions” on page 29 for details.

- OMVS segment for the user ID that is running the application being debugged
- DBGMR started task
- BPX.SERVER permit for started task user ID
- Program control for started task load library

The Integrated Debugger optionally uses the following security definitions. See “Security considerations” in the *Host Configuration Reference Guide* (SC14-7290) for details.

- AQE.AUTHDEBUG.WRITEBUFFER permit for users to debug CICS transactions loaded into read-only memory in a problem-state (non-authorized) environment.

**Note:** To simplify migration from an existing Developer for System z setup without the Integrated Debugger, sample JCL FEK.SFEKSAMP(AQERACF) with RACF commands is provided to define just the Integrated Debugger related security definitions.

## Integrated debugger CICS updates

To debug CICS transactions, the Integrated Debugger requires the following CICS updates.

- CICS JCL update:
  - Define the FEK,SFEKAUTH load library in the region’s DFHRPL DD statement if the library is not in LINKLIST.
  - For z/OS 1.13 and higher, define the SYS1.SIEAMIGE load library in the region’s STEPLIB DD statement if the library is not in LINKLIST. See the z/OS Binder API information in “Integrated debugger parmlib updates” on page 121 for more details.
- CICS CSD updates:

Define the debugger to a CICS region, as documented in the AQECSD sample CSD update job. AQECSD is located in FEK.#CUST.JCL, unless you specified a different location when you customized and submitted job FEK.SFEKSAMP(FEKSETUP). See “Customization setup” on page 15 for more details.

---

## (Optional) Problem Determination Tools support

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

---

Developer for System z can integrate with various IBM z/OS Problem Determination Tools. The following sections describe how to make these tools available to the Developer for System z client:

- IBM Debug Tool for z/OS: See “(Optional) DB2 and IMS debug support” on page 123. Note that since version 9.0.1, Developer for System z provides the Integrated Debugger, which can be used instead of Debug Tool.
- IBM File Manager for z/OS: See “(Optional) File Manager support” on page 123.

- IBM Fault Analyzer for z/OS: No Developer for System z host system configuration required. Note that since version 9.0, Developer for System z no longer supports Fault Analyzer Integration (FAI). Older clients who still have this support should uninstall the function and install the IBM Fault Analyzer plug-in for Eclipse. This plug-in is available from the IBM Problem Determination Tools Plug-ins web page, <http://www-01.ibm.com/software/awdtools/deployment/pdplugins/>.

---

## (Optional) DB2 and IMS debug support

This customization task does not require assistance, special resources, or special customization tasks for the Developer for System z configuration. However, there are requirements for the IBM Debug Tool for z/OS configuration.

---

IBM Debug Tool for z/OS provides a customized Language Environment (LE) user exit, CEEBXITA, which returns the TEST runtime options when called by the LE initialization logic in IMS and DB2 Stored Procedures. IBM Debug Tool for z/OS also provides the Debug Tool extension for the Problem Determination Tools Common Components server, to create and manage the TEST runtime options data set on the z/OS system. Developer for System z can use and enhance IBM Debug Tool for z/OS's support for managing debug profiles for the IMS and DB2 Stored Procedure run-times.

The IBM Debug Tool for z/OS documentation describes in detail the required setup, which is briefly mentioned here.

- Specifying the TEST runtime options through the Language Environment user exit, h1q.SEQA\*
- Adding support for the DTSP Profile view
  - Install the Problem Determination Tools Common Components Server (h1q.SIPV\*, job IPVGSVRJ)
  - Install and configure the Debug Tool extension for the h1q.SEQA\*Problem Determination Tools Common Components

### Note:

- The IBM Debug Tool for z/OS product must be obtained, installed, and configured separately. The installation and customization of this product is not described in this manual.
- The Developer for System z client does not use the DTSP Profile view plug-in for Eclipse.
- The Developer for System z client does not use the Language Environment user exit for regular batch-mode debugging.
- The Developer for System z client communicates directly with the Problem Determination Tools Common Components server, which implies that the user must know this port number, and that the port used by this server must be opened on your firewall protecting the z/OS host system.

---

## (Optional) File Manager support

This customization task does not require assistance, special resources, or special customization tasks for the Developer for System z configuration. However, there are requirements for the IBM File Manager for z/OS configuration.

---

Developer for System z's initial integration with IBM File Manager for z/OS has been marked as deprecated in Developer for System z version 8.0.3 and is no longer supported in version 8.5. The services provided by this function have been moved to different areas. Some functions, such as unformatted QSAM editing, are now part of regular data set handling by Developer for System z. More advanced functions, such as formatted data editing using copybooks or include files, require that the IBM File Manager plug-in for Eclipse is installed on the Developer for System z client. This plug-in is available from the IBM Problem Determination Tools Plug-ins web page, <http://www-01.ibm.com/software/awdtools/deployment/pdtpugins/>.

The IBM File Manager plug-in for Eclipse uses the Problem Determination Tools Server to access the File Manager services. This server is not used by the File Manager ISPF panel interface. Therefore, you have additional File Manager setup tasks, specific for the Problem Determination Tools. For more details, see your File Manager documentation.

The port number used by the Problem Determination Tools server must be specified in the `rsed.envvars` directive `PD_SERVER_PORT`.

`rsed.envvars` is located in `/etc/rdz/`, unless you specified a different location when you customized and submitted the `FEK.SFEKSAMP(FEKSETUP)` job. For more details, see "Customization setup" on page 15. You can edit the file with the TSO **EDIT** command.

**Note:**

- The IBM File Manager for z/OS product must be obtained, installed, and configured separately. The installation and customization of this product is not described in this manual.
- The Developer for System z client communicates directly with the Problem Determination Tools server, which implies that the port used by this server must be opened on the firewall that protects the z/OS host system.

---

## (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. Therefore, clear the `WORKAREA` and `/tmp` directories from time to time.

z/OS UNIX provides a shell script, `skulker`, that deletes files according to the directory they are in and their age. Combined with the z/OS UNIX cron daemon, which runs commands at specified dates and times, you can set up an automated tool that periodically cleans targeted directories. For more information about the `skulker` script and the cron daemon, see *UNIX System Services Command Reference* (SA22-7802).

**Note:** WORKAREA is located in /var/rdz/, unless you specified a different location when you customized and submitted the FEK.SFEKSAMP(FEKSETUP) job. For more details, see “Customization setup” on page 15.



---

## Chapter 8. Installation verification

After completing the product customization, you can use the Installation Verification Programs (IVPs) described in this chapter to verify the successful setup of key product components.

---

### Verify the started tasks

#### JMON, the JES Job Monitor

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

```
FEJ211I Server ready to accept connections.
```

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

**Note:** Start JES Job Monitor before continuing with the other IVP tests.

#### RSED, the RSE daemon

Start the RSED started task or user job with the IVP=IVP parameter. With this parameter, the server ends after doing some installation verification tests. The output of these tests is available in DD STDOUT. In case of certain errors, data is also available in DD STDERR. Check DD STDOUT for messages that indicate that the following IVPs were successful:

- Java startup
- JES Job Monitor connection
- TCP/IP setup

The STDOUT data should look like the following sample:

```
-----  
RSE daemon startup script  
-----
```

```
arguments: IVP -C/etc/rdz -P
```

```
RSE daemon IVP test
```

```
CDFMVS08 -- Fri Mar 23 17:50:52 2012 UTC  
uid=8(STCRSE) gid=1(STCGROUP)
```

```
started from /usr/lpp/rdz/bin/rsed.sh  
startup script version Aug09,2012
```

```
configuration files located in /etc/rdz -- startup argument  
daemon port is 4035 -- set in rsed.envvars  
debug level is 1 -- set in rsecomm.properties  
TMPDIR=/tmp -- default
```

```
-----  
current environment variables  
-----
```

```
@="/usr/lpp/rdz/bin/rsed.sh" @[1]="-C/etc/rdz" @[2]="-P"  
ANT_HOME="/usr/lpp/Apache/Ant/apache-ant-1.7.1"  
CGI_DTWORK="/var/rdz"  
CGI_ISPCONF="/etc/rdz"
```

```

CGI_ISPHOME="/usr/lpp/ispf"
CGI_ISPWORK="/var/rdz"
CGI_TRANTABLE="FEK.#CUST.LSTRANS.FILE"
CLASSPATH=".:usr/lpp/rdz/lib:usr/lpp/rdz/lib/dstore_core.jar:usr/lpp/
ERRNO="0"
HOME="/tmp"
IFS="
"
JAVA_HOME="/usr/lpp/java/J6.0"
JAVA_PROPAGATE="NO"
LANG="C"
LIBPATH=".:usr/lib:usr/lpp/java/J6.0/bin:usr/lpp/java/J6.0/bin/classi
LINENO="66"
LOGNAME="STCRSE"
MAILCHECK="600"
OLDPWD="/tmp"
OPTIND="1"
PATH=".:usr/lpp/java/J6.0/bin:usr/lpp/rdz/bin:usr/lpp/ispf/bin:bin:/
PPID="33554711"
PS1="\$ "
PS2 "> "
PS3="#? "
PS4="+ "
PWD="/etc/rdz"
RANDOM="27298"
RSE_CFG="/etc/rdz"
RSE_HOME="/usr/lpp/rdz"
RSE_LIB="/usr/lpp/rdz/lib"
SECONDS="0"
SHELL="/bin/sh"
STEPLIB="NONE"
TMPDIR="/tmp"
TZ="EST5EDT"
X_ARG="-T"
X_C="-- startup argument"
X_KEY="-T"
X_L="-- set in rsecomm.properties"
X_LOG="1"
X_P="-- set in rsed.envvars"
X_PORT="4035"
X_VAL=""
="-----"
_BPX_SHAREAS="YES"
_BPX_SPAWN_SCRIPT="YES"
_CEE_DMPTARG="/tmp"
_CEE_RUNOPTS="ALL31(ON) HEAP(32M,32K,ANYWHERE,KEEP,,) TRAP(ON)"
_CMDSERV_BASE_HOME="/usr/lpp/ispf"
_CMDSERV_CONF_HOME="/etc/rdz"
_CMDSERV_WORK_HOME="/var/rdz"
_EDC_ADD_ERRNO2="1"
_RSE_ISPF_OPTS="&SESSION=SPAWN"
_RSE_DAEMON_CLASS="com.ibm.etools.zos.server.RseDaemon"
_RSE_DAEMON_IVP_TEST="1"
_RSE_HOST_CODEPAGE="IBM-1047"
_RSE_JAVA_OPTS=" -DISPF_OPTS='&SESSION=SPAWN' -DA_PLUGIN_PATH=
_RSE_JMON_PORT="6715"
_RSE_LOG_LEVEL="1"
_RSE_POOL_SERVER_CLASS="com.ibm.etools.zos.server.ThreadPoolProcess"
_RSE_RSED_PORT="4035"
_RSE_SAF_CLASS="/usr/include/java_classes/IRRRacf.jar"
_RSE_SCRIPT_VERSION="Jan09,2012"
_RSE_SERVER_CLASS="org.eclipse.dstore.core.server.Server"
_RSE_SERVER_TIMEOUT="120000"
_SCLMDT_BASE_HOME="/usr/lpp/rdz"
_SCLMDT_CONF_HOME="/var/rdz/scldmt"
_SCLMDT_TRANTABLE="FEK.#CUST.LSTRANS.FILE"
_SCLMDT_WORK_HOME="/var/rdz"

```

```
debug_level="1"
```

```
-----  
Address Space size limits  
-----
```

```
current address space size limit is 1913626624 (1825.0 MB)  
maximum address space size limit is 2147483647 (2048.0 MB)  
-----
```

```
service history  
-----
```

```
Fri Jun 14 13:47:39 2013 -- COPY -- HHOP900 v9000 created 14 Jun 2013  
-----
```

```
java service level  
-----
```

```
java full version "J2RE 1.6.0 IBM z/OS build pmz3160sr13-20130207_01(SR13)  
-----
```

```
LE runtime options  
-----
```

```
Options Report for Enclave main 05/23/12 1:50:52 PM  
Language Environment V01 R11.00
```

LAST WHERE SET	OPTION
Installation default	ABPERC(NONE)
Programmer default	ABTERMENC(RETCODE)
Installation default	NOAIXBLD
Invocation command	ALL31(ON)
Programmer default	ANYHEAP(32768,16384,ANYWHERE,FREE)
Installation default	NOAUTOTASK
Programmer default	BELOWHEAP(32768,16384,FREE)
Installation default	CBLOPTS(ON)
Installation default	CBLPSTHPOP(ON)
Installation default	CBLQDA(OFF)
Installation default	CEEDUMP(60,SYSOUT=*,FREE=END,SPIN=UNALL
Installation default	CHECK(ON)
Installation default	COUNTRY(US)
Installation default	NODEBUG
Installation default	DEPTHCONDLMT(10)
Installation default	DYNDUMP(*USERID,NODYNAMIC,TDUMP)
Installation default	ENVAR("")
Installation default	ERRCOUNT(0)
Installation default	ERRUNIT(6)
Installation default	FILEHIST
Installation default	FILETAG(NOAUTOCVT,NOAUTOTAG)
Default setting	NOFLOW
Invocation command	HEAP(33554432,32768,ANYWHERE,KEEP,16384
Installation default	HEAPCHK(OFF,1,0,0,0)
Installation default	HEAPPOLS(OFF,8,10,32,10,128,10,256,10,
Installation default	INFMSGFILTER(OFF,,,) )
Installation default	INQPCOPN
Installation default	INTERRUPT(OFF)
Programmer default	LIBSTACK(32768,16384,FREE)
Installation default	MSGFILE(SYSOUT,FBA,121,0,NOENQ)
Installation default	MSGQ(15)
Installation default	NATLANG(ENU)
Ignored	NONONIPSTACK(See THREADSTACK)
Installation default	OCSTATUS
Installation default	NOPC
Installation default	PLITASKCOUNT(20)
Programmer default	POSIX(ON)
Installation default	PROFILE(OFF,"")
Installation default	PRTUNIT(6)
Installation default	PUNUNIT(7)

```

Installation default      RDRUNIT(5)
Installation default      RECPAD(OFF)
Invocation command        RPTOPTS(ON)
Installation default      RPTSTG(OFF)
Installation default      NORTEREUS
Installation default      NOSIMVRD
Programmer default
STACK(65536,65536,ANYWHERE,KEEP,524288,131072)
Installation default      STORAGE(NONE,NONE,NONE,0)
Installation default      TERMTHDACT(TRACE,,96)
Installation default      NOTEST(ALL,"*","PROMPT","INSPREF")
Installation default      THREADHEAP(4096,4096,ANYWHERE,KEEP)
Installation default
THREADSTACK(OFF,4096,4096,ANYWHERE,KEEP,131072,
Installation default      TRACE(OFF,4096,DUMP,LE=0)
Invocation command        TRAP(ON,SPIE)
Installation default      UPSI(00000000)
Installation default      NOUSRHDLR(,)
Installation default      VCTRSAVE(OFF)
Installation default      XPLINK(OFF)
Installation default      XUFLOW(AUTO)

```

```
-----
java startup test...
```

```
-----
java full version "JRE 1.6.0 IBM z/OS build pmz3160sr13-20130207_01
(SR13)"
java version "1.6.0"
Java(TM) SE Runtime Environment (build pmz3160sr13-20130207_01(SR13))
IBM J9 VM (build 2.4, JRE 1.6.0 IBM J9 2.4 z/OS s390-31 jvmmz3160sr13-
20130114_1
J9VM - 20130114_134867
JIT - r9_20130108_31100
GC - 20121212_AA)
JCL - 20130204_01

```

```
-----
JES Job Monitor test...
```

```
-----
executed on CDFMVS08 -- Fri Mar 23 17:50:52 EDT 2012
executed by uid=8(STCRSE) gid=1(STCGROUP)
using /etc/rdz/rsed.envvars

```

```
current address space size limit is 1913626624 (1825.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)

```

```
testing JES Job Monitor on port 6715...
hostName=CDFMVS08
hostAddr=9.42.112.75
IPv4 is supported
Waiting for JES Job Monitor response...
ACKNOWLEDGE01v03
Success

```

```
-----
TCP/IP IVP test...
```

```
-----
executed on CDFMVS08 -- Fri Mar 23 17:50:53 EDT 2012
executed by uid=8(STCRSE) gid=1(STCGROUP)
using /etc/rdz/rsed.envvars

```

```
current address space size limit is 1913626624 (1825.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)

```

```
-----
TCP/IP resolver configuration (z/OS UNIX search order):

```

```

-----
Resolver Trace Initialization Complete -> 2012/05/23 17:50:54.208378

res_init Resolver values:
Global Tcp/Ip Dataset = None
Default Tcp/Ip Dataset = None
Local Tcp/Ip Dataset = /etc/resolv.conf
Translation Table = Default
UserId/JobName = STCRSE
Caller API = LE C Sockets
Caller Mode = EBCDIC
(L) DataSetPrefix = TCPIP
(L) HostName = CDFMVS08
(L) TcpIpJobName = TCPIP
(L) DomainOrigin = RALEIGH.IBM.COM
(L) NameServer = 9.42.206.2
                9.42.206.3
(L) NsPortAddr = 53 (L) ResolverTimeout = 10
(L) ResolveVia = UDP (L) ResolverUdpRetries = 1
(*) Options NDots = 1
(*) SockNoTestStor
(*) AlwaysWto = NO (L) MessageCase = MIXED
(*) LookUp = DNS LOCAL
res_init Succeeded
res_init Started: 2012/05/23 17:50:54.229888
res_init Ended: 2012/05/23 17:50:54.229898
*****
MVS TCP/IP NETSTAT CS V1R11 TCPIP Name: TCPIP 17:50:54
Tcpiip started at 11:31:40 on 05/23/2012 with IPv6 enabled

```

```

-----
host IP address:
-----
hostName=CDFMVS08
hostAddr=9.42.112.75
bindAddr=9.42.112.75
localAddr=9.42.112.75

```

Success, addresses match

```

-----
RSE daemon IVP ended -- return code 0 -- Fri Mar 23 17:50:55 EDT 2012
-----

```

**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)
```

## DBGMGR, the debug manager

Start the optional DBGMGR started task or user job. The server issues the following console message upon successful startup, where clientport is the number of the port used for external (client-host) communication, and hostport is the port number used for internal (host-confined) communication.

```
AQECM001I Debug Manager startup complete (clientport/hostport)
```

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

**Note:** Start DBGMGR before continuing with the other debug-related IVP tests.

---

## IVP operator commands

An active RSE daemon supports the **IVP** modify command, which you can use to do selected IVPs from the console.

### PassTicket reusability

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

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

The command should return an output like that in the following sample:

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

```
+FEK900I PASSTICKET IVP: start: serverid=STCRSE userid=IBMUSER
+FEK900I PASSTICKET IVP: the default applid=FEKAPPL
+FEK900I PASSTICKET IVP: Success, PassTicket IVP finished normally
+FEK901I PASSTICKET IVP Exit code = 0
```

### RSE daemon connection

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

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

This command is functionally identical to the fekfivpd IVP described in “Verify the services” on page 133 but with the benefit that no password is required. RSE generates a PassTicket and uses this as password. The command should return an output like that in the following sample:

```
F RSED,APPL=IVP DAEMON,IBMUSER
```

```
+FEK900I DAEMON IVP: SSL is disabled
+FEK900I DAEMON IVP: connected
+FEK900I DAEMON IVP: 1343
+FEK900I DAEMON IVP: 8878350
+FEK900I DAEMON IVP: Success
+FEK901I DAEMON IVP Exit code = 0
```

### ISPF Client Gateway

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

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

This command is functionally identical to the fekfivpi IVP described in “Verify the services” on page 133. The command should return an output like that in the following sample:

```
F RSED,APPL=IVP ISPF,IBMUSER
```

```
+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.SISPLOAD
+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 TSO/ISPF session initializati
on check only
+FEK900I ISPF IVP: *** CHECK : ENVIRONMENT VARIABLES - key variables
displayed below :
+FEK900I ISPF IVP: Server PATH          = ./usr/lpp/java/J6.0/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: CGI_ISPHOME         = /usr/lpp/ispf
+FEK900I ISPF IVP: CGI_ISPCONF        = /etc/rdz
+FEK900I ISPF IVP: CGI_ISPWORK       = /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

```

---

## Verify the services

The Developer for System z installation provides several Installation Verification Programs (IVP) for the basic and optional services. The IVP scripts are located in the installation directory which, by default, is /usr/lpp/rdz/bin/.

*Table 19. IVPs for services*

fekfivpc	“(Optional) CARMA connection” on page 139
fekfivpd	“RSE daemon connection” on page 136

Table 19. IVPs for services (continued)

fekfivpi	"ISPF's TSO/ISPF Client Gateway connection" on page 137
fekfivpj	"JES Job Monitor connection" on page 137
fekfivps	"(Optional) SCLMDT connection" on page 140
fekfivpt	"TCP/IP setup" on page 135

The tasks described in the following sections require you to be active in z/OS UNIX. This can be done by issuing the **OMVS** TSO command. To return to TSO, use the **exit** command.

A large region size is required for the user ID that executes the IVPs because functions such as Java, which require a lot of memory, are executed. You should set the region size to 131072 kilobytes (128 megabytes) or more.

The following sample error is a clear indication of an insufficient region size, but other errors can occur, too. For example, Java might fail to start.

```
CEE5213S The signal SIGPIPE was received.
%z/OS UNIX command%: command was killed by signal number 13
  %line-number% *-* %REXX command%
    +++ RC(137) +++
```

**Note:** The Developer for System z started tasks must be active before starting the IVP test.

## IVP initialization

All sample commands in this section require certain environment variables to be set. This way, the IVP scripts are available through the PATH statement and the location of the customized configuration files is known. Use the **pwd** and **cd** commands to verify and change your current directory to the directory with the customized configuration files. The `ivpinit` shell script can then be used to set the RSE environment variables, such as in the following sample, where \$ is the z/OS UNIX prompt:

```
$ pwd
/u/userid
$ cd /etc/rdz
$ ./ivpinit
RSE configuration files located in /etc/rdz --default
added /usr/lpp/rdz/bin to PATH
```

The first period (.) in `./ivpinit` is a z/OS UNIX command to run the shell in the current environment, so that the environment variables set in the shell are effective even after exiting the shell. The second period (.) is referring to the current directory.

**Note:**

- If `./ivpinit` is not executed before the `fekfivp*` scripts, the path to these scripts must be specified when calling them, as in the following sample:  
`/usr/lpp/rdz/bin/fekfivpr 512 USERID`

Also, if `./ivpinit` is not executed first, all `fekfivp*` scripts ask for the location of the customized `rsed.envvars`.

- Some IVP tests use the TCP/IP REXX socket API, which requires that the TCP/IP load library, which, by default, is TCPIP.SEZALOAD, is in LINKLIST or STEPLIB. The following command might be necessary to be able to execute these IVP tests :

```
$ EXPORT STEPLIB=$STEPLIB:TCPIP.SEZALOAD
```

Adding a non-APF authorized library to an existing STEPLIB removes the APF authorization for existing STEPLIB data sets.

If CEE.SCEELKED is in LINKLIST or STEPLIB, TCPIP.SEZALOAD must be placed before CEE.SCEELKED. Failure to do so will result in a 0C1 system abend for the TCP/IP REXX socket calls.

For information about diagnosing host system connection problems, see "Troubleshooting configuration problems" in the *Host Configuration Reference* (SC14-7290) and the Technotes in the Support section of the Developer for System z website, <http://www-03.ibm.com/software/products/us/en/developerforsystemz/> .

## Port availability

The JES Job Monitor and RSE daemon port availability can be verified by issuing the **netstat** command. The result should show the ports used by these services, as in the following samples:

### IPv4

```
$ netstat
MVS TCP/IP NETSTAT CS VxRy   TCPIP Name: TCPIP      13:57:36
User Id Conn      Local Socket      Foreign Socket      State
----- ----      -
RSED      0000004B 0.0.0.0..4035    0.0.0.0..0         Listen
JMON      00000037 0.0.0.0..6715    0.0.0.0..0         Listen
```

### IPv6

```
$ netstat
MVS TCP/IP NETSTAT CS VxRy   TCPIP Name: TCPIP      14:03:35
User Id Conn      State
----- ----      -
RSED      0000004B Listen
Local Socket: 0.0.0.0..4035
Foreign Socket: 0.0.0.0..0
JMON      00000037 Listen
Local Socket: 0.0.0.0..6715
Foreign Socket: 0.0.0.0..0
```

## TCP/IP setup

Developer for System z is dependent upon TCP/IP having the correct host name when it is initialized. This implies that the different TCP/IP and Resolver configuration files must be set up correctly. For more information about TCP/IP and Resolver setup, see "Setting up TCP/IP" in the *Host Configuration Reference* (SC14-7290). Verify the current settings by executing the following command:

```
fekfivpt
```

**Note:** This IVP issues the TCPIP **netstat -u** command, which might be protected against execution by your security software. See the EZB.NETSTAT.mvsname.tcprocname.UP profile in the SERVAUTH class.

The command should return an output that is similar to this sample:

```

$ fekfivpt

executed on CDFMVS08 -- Wed Jul  2 13:11:54 EDT 2008
executed by uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars

current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)

-----
TCP/IP resolver configuration (z/OS UNIX search order):
-----
Resolver Trace Initialization Complete -> 2008/07/02 13:11:54.745964

res_init Resolver values:
Global Tcp/Ip Dataset = None
Default Tcp/Ip Dataset = None
Local Tcp/Ip Dataset = /etc/resolv.conf
Translation Table = Default
UserId/JobName = USERID
Caller API = LE C Sockets
Caller Mode = EBCDIC
(L) DataSetPrefix = TCPIP
(L) HostName = CDFMVS08
(L) TcpIpJobName = TCPIP
(L) DomainOrigin = RALEIGH.IBM.COM
(L) NameServer = 9.42.206.2
                9.42.206.3
(L) NsPortAddr = 53          (L) ResolverTimeout = 10
(L) ResolveVia = UDP        (L) ResolverUdpRetries = 1
(*) Options NDots = 1
(*) SockNoTestStor
(*) AlwaysWto = NO          (L) MessageCase = MIXED
(*) LookUp = DNS LOCAL

res_init Succeeded
res_init Started: 2008/07/02 13:11:54.755363
res_init Ended: 2008/07/02 13:11:54.755371
*****
MVS TCP/IP NETSTAT CS V1R9      TCPIP Name: TCPIP      13:11:54
Tcpip started at 01:28:36 on 06/23/2008 with IPv6 enabled

-----
host IP address:
-----
hostName=CDFMVS08
hostAddr=9.42.112.75
bindAddr=9.42.112.75
localAddr=9.42.112.75

Success, addresses match

```

## RSE daemon connection

Verify the RSE daemon connection by executing the following command.

```
fekfivpd
```

After prompting for a password, the command should return an output that is similar to the following sample:

```

$ fekfivpd

executed on CDFMVS08 -- Wed Jul  2 15:00:27 EDT 2008
executed by uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars

current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)

```

```
attempting to connect userid USERID using port 4035 ...
```

```
Password:  
SSL is disabled  
connected  
8108  
570655399  
Success
```

When testing an SSL-enabled connection, ensure that the user ID executing the IVP has access to all of the required certificates, including the CA certificates that are used to sign the Developer for System z certificate. The operator command version of this IVP, `F RSED,APPL=IVP DAEMON,userid`, uses the SSL setup that is done for RSE host system, and is therefore less error prone. Some common SSL-related errors are given in the following list:

- Verify that the user ID executing the IVP has access to all of the required certificates if you get this error message: `gsk_environment_init() failed: Error detected while opening the certificate data base`
- Verify that the signing CA certificates are also on the keyring if you get this error message: `gsk_secure_socket_init() failed: Certificate validation error`

## JES Job Monitor connection

Verify the JES Job Monitor connection by executing the following command.

```
fekfivpj
```

The command should return the JES Job Monitor acknowledge message, similar to what appears in the following sample (\$ is the z/OS UNIX prompt):

```
$ fekfivpj
```

```
executed on CDFMVS08 -- Wed Jul  2 15:00:27 EDT 2008  
executed by uid=1(USERID) gid=0(GROUP)  
using /etc/rdz/rsed.envvars
```

```
current address space size limit is 1914675200 (1826.0 MB)  
maximum address space size limit is 2147483647 (2048.0 MB)
```

```
testing JES Job Monitor on port 6715...  
hostName=CDFMVS08  
hostAddr=9.42.112.75  
IPv4 is supported  
Waiting for JES Job Monitor response...  
ACKNOWLEDGE01v03
```

```
Success
```

## ISPF's TSO/ISPF Client Gateway connection

Verify the connection to ISPF's TSO/ISPF client Gateway by executing the following command:

```
fekfivpi
```

The command should return the result of ISPF's TSO/ISPF client Gateway-related checks, such as variables, HFS modules, starting and stopping TSO/ISPF session. The output should be similar to what is given in the following sample:

```
$ fekfivpi
```

```
executed on CDFMVS08 -- Wed Jul  2 15:00:27 EDT 2008
```

executed by uid=1(USERID) gid=0(GROUP)  
using /etc/rdz/rsed.envvars

current address space size limit is 1914675200 (1826.0 MB)  
maximum address space size limit is 2147483647 (2048.0 MB)

-----  
/etc/rdz/ISPF.conf content:  
-----

```
ispmlib=ISP.SISPMENU
isptlib=ISP.SISPTENU
ispplib=ISP.SISPPENU
ispplib=ISP.SISPSLIB
ispllib=ISP.SISPLOAD
sysproc=ISP.SISPCLIB,FEK.SFEKPROC
```

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

RSE connection and base TSO/ISPF session initialization check only  
\*\*\* CHECK : ENVIRONMENT VARIABLES - key variables displayed below :

```
Server PATH          =
/usr/lpp/java/J6.0/bin:/usr/lpp/rdz/lib:/usr/lpp/ispf/bin:
/bin:/usr/sbin:.
```

```
STEPLIB              = FEK.SFEKAUTH:FEK.SFEKLOAD
```

```
CGI_ISPHOME          = /usr/lpp/ispf
CGI_ISPCONF           = /etc/rdz
CGI_ISPWORK           = /var/rdz
```

-----  
\*\*\* CHECK : USS MODULES  
Checking ISPF Directory : /usr/lpp/ispf  
Checking modules in /usr/lpp/ispf/bin directory  
Checking for ISPF configuration file ISPF.conf  
RC=0  
MSG: SUCCESSFUL

-----  
\*\*\* CHECK : TSO/ISPF INITIALIZATION  
( TSO/ISPF session will be initialized )  
RC=0  
MSG: SUCCESSFUL

-----  
\*\*\* CHECK: Shutting down TSO/ISPF IVP session  
RC=0  
MSG: SUCCESSFUL

-----  
Host installation verification completed successfully  
-----

**Note:** If any of the ISPF checks fail, more detailed information is shown.

fekfivpi has the following optional, non-positional, parameters:

**-file** fekfivpi can produce large amounts of output, running into hundreds of lines. The **-file** parameter sends this output to a file, \$TMPDIR/fekfivpi.log, where \$TMPDIR is the value of the TMPDIR directive in rsed.envvars which, by default, is /tmp.

### **-debug**

The -debug parameter creates detailed test output. Do not use this option unless directed by the IBM support center.

## **(Optional) CARMA connection**

Verify the connection to CARMA by executing the following command:

```
fekfivpc
```

The command should return the result of CARMA-related checks, as shown in the following sample:

```
$ fekfivpc
```

```
executed on CDFMVS08 -- Fri Aug 20 14:15:46 EDT 2010
executed by uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars
```

```
current address space size limit is 140484608 (134.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)
```

```
*** /etc/rdz/CRASRV.properties content:
port.start = 5227
port.range = 100
startup.script.name = /usr/lpp/rdz/bin/carma.startup.rex
clist.dsname = *CRASTART
crastart.stub = /usr/lpp/rdz/bin/CRASTART
crastart.configuration.file = /etc/rdz/crastart.endevor.conf
crastart.syslog = Partial
crastart.timeout = 420
```

```
*** Creating /tmp/fekfivpc.log
```

```
*** Verifying CARMA installation...
```

```
1. Creating CARMA connection (timeout after 60 seconds)
2. Initializing CARMA
3. Retrieving RAM list
   The following RAMs were found
      00 CA Endevor SCM      Unique ID: COM.IBM.CARMA.ENDEVORRAM
4. Getting customization data for RAM 00
5. Initializing RAM 00
6. Retrieving Repository Instance List
   Found 6 Repository Instance(s)
7. Terminating RAM 00
8. Terminating CARMA
```

```
*** IVP Successful!!!!
```

**Note:** If the IVP fails, verify the content of /tmp/fekfivpc.log. This log documents the communication between RSE and CARMA and might contain information that helps to find the root cause of the failure.

fekfivpc has the following optional, non-positional, parameters:

### **-noram**

By default, fekfivpc starts the first RAM that is defined in the CRADEF VSAM data set. There might be instances when you do not want to test the RAM; for example, a third-party RAM is listed first, and it requires unexpected input. In such cases, you can use the -noram startup argument to omit the RAM-specific steps (step 4 through 7) of the IVP test.

## (Optional) SCLMDT connection

Verify the connection to SCLM Developer Toolkit by executing the following command:

```
fekfivps
```

The command should return the result of SCLM Developer Toolkit related checks, such as variables, HFS modules, REXX runtime, starting and stopping TSO/ISPF session, and show an output similar to the following sample:

```
$ fekfivps
```

```
executed on CDFMVS08 -- Wed Jul 2 15:00:27 EDT 2008
```

```
executed by uid=1(USERID) gid=0(GROUP)
```

```
using /etc/rdz/rsed.envvars
```

```
current address space size limit is 1914675200 (1826.0 MB)
```

```
maximum address space size limit is 2147483647 (2048.0 MB)
```

```
-----  
/etc/rdz/ISPF.conf content:  
-----
```

```
isplib=ISP.SISPMENU  
isptlib=ISP.SISPTENU  
ispslib=ISP.SISPLIB  
ispllib=ISP.SISPLIB  
ispllib=ISP.SISPLIB  
sysproc=ISP.SISPCLIB,FEK.SFEKPROC  
-----
```

```
Host install verification for RSE  
Review IVP log messages from HOST below :  
-----
```

```
*** CHECK : ENVIRONMENT VARIABLES - key variables displayed below :
```

```
Server PATH      = /usr/lpp/java/J6.0/bin:/usr/lpp/rdz/lib:/usr/lpp/ispf/bin:  
/bin:/usr/sbin:.
```

```
STEPLIB          = FEK.SFEKAUTH:FEK.SFEKLOAD
```

```
CGI_ISPHOME     = /usr/lpp/ispf  
CGI_ISPCONF     = /etc/rdz  
CGI_ISPWORK     = /var/rdz  
_SCLMDT_CONF_HOME = /var/rdz/sclmdt  
_SCLMDT_WORK_HOME = /var/rdz  
_SCLMDT_TRANSTABLE = FEK.#CUST.LSTRANS.FILE  
-----
```

```
*** CHECK : JAVA PATH SETUP VERIFICATION
```

```
RC=0
```

```
MSG: SUCCESSFUL  
-----
```

```
*** CHECK : USS MODULES
```

```
Checking ISPF Directory : /usr/lpp/ispf  
Checking modules in /usr/lpp/ispf/bin directory  
Checking for ISPF configuration file ISPF.conf  
Checking install bin Directory : /usr/lpp/rdz/bin  
RC=0  
MSG: SUCCESSFUL  
-----
```

```
*** CHECK : REXX RUNTIME ENVIRONMENT
```

```
RC=0
```

```
MSG: SUCCESSFUL
```

```
-----  
*** CHECK : TSO/ISPF INITIALIZATION  
( TSO/ISPF session will be initialized )  
RC=0  
MSG: SUCCESSFUL
```

```
-----  
*** CHECK: Shutting down TSO/ISPF IVP session  
RC=0  
MSG: SUCCESSFUL
```

```
-----  
Host installation verification completed successfully  
-----
```

**Note:** If any of the SCLMDT checks fail, more detailed information is shown.

fekfivps has the following optional, non-positional, parameters:

**-file** fekfivps can produce large amounts of output, running into hundreds of lines. The **-file** parameter sends this output to a file, \$TMPDIR/fekfivps.log, where \$TMPDIR is the value of the TMPDIR directive in rsed.envvars, which, by default, is /tmp.

**-debug**

The **-debug** parameter creates detailed test output. Do not use this option unless directed by the IBM support center.

## (Optional) Integrated Debugger connection

Verify the connection to Integrated Debugger by executing the following command:

```
fekfivpe
```

The command should return the result of Integrated Debugger related checks, and show an output similar to the following sample:



---

## Chapter 9. Security definitions

Customize and submit the sample FEKRACF member, which has sample RACF and z/OS UNIX commands to create the basic security definitions for Developer for System z.

FEKRACF is located in FEK.#CUST.JCL, unless you specified a different location when you customized and submitted the FEK.SFEKSAMP(FEKSETUP) job. For more details, see “Customization setup” on page 15.

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

### Note:

- For those sites that use CA ACF2™ for z/OS, see the product page on the CA support site (<https://support.ca.com>) and check for the related Developer for System z Knowledge Document, TEC492389. This Knowledge Document has details on the security commands that are necessary to properly configure Developer for System z.
- For those sites that use CA Top Secret® for z/OS, see the product page on the CA support site (<https://support.ca.com>) and check for the related Developer for System z Knowledge Document, TEC492091. This Knowledge Document has details on the security commands that are necessary to properly configure Developer for System z.

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

---

## Requirements and checklist

To complete the security setup, the security administrator must know the values that are listed in Table 20. These values were defined during previous steps of the installation and customization of Developer for System z.

Table 20. Security setup variables

Description	• Default value • Where to find the answer	Value
Developer for System z product high-level qualifier	<ul style="list-style-type: none"><li>• FEK</li><li>• SMP/E installation</li></ul>	
Developer for System z customization high-level qualifier	<ul style="list-style-type: none"><li>• FEK.#CUST</li><li>• FEK.SFEKSAMP(FEKSETUP), as described in “Customization setup” on page 15.</li></ul>	
Integrated Debugger started task name	<ul style="list-style-type: none"><li>• DBGMGR</li><li>• FEK.#CUST.PROCLIB(DBGMGR), as described in “PROCLIB changes” on page 23</li></ul>	

Table 20. Security setup variables (continued)

Description	<ul style="list-style-type: none"> <li>• Default value</li> <li>• Where to find the answer</li> </ul>	Value
JES Job Monitor started task name	<ul style="list-style-type: none"> <li>• JMON</li> <li>• FEK.#CUST.PROCLIB(JMON), as described in "PROCLIB changes" on page 23.</li> </ul>	
RSE daemon started task name	<ul style="list-style-type: none"> <li>• RSED</li> <li>• FEK.#CUST.PROCLIB(RSED), as described in "PROCLIB changes" on page 23.</li> </ul>	
Application ID	<ul style="list-style-type: none"> <li>• FEKAPPL</li> <li>• /etc/rdz/rsed.envvars, as described in "Defining extra Java startup parameters with _RSE_JAVAOPTS" on page 45</li> </ul>	

The following list is an overview of the actions that are required to complete the basic security setup of Developer for System z. As documented in the following sections, different methods can be used to fulfill these requirements, depending on the required security level. For information about the security setup of optional Developer for System z services, see the previous sections.

- "Activate the security settings and classes"
- "Define an OMVS segment for Developer for System z users" on page 145
- "Define the Developer for System z started tasks" on page 146
- "Define RSE as a secure z/OS UNIX server" on page 147
- "Define the MVS program controlled libraries for RSE" on page 147
- "Define the PassTicket support for RSE" on page 148
- "Define the application protection for RSE" on page 149
- "Define the JES command security" on page 150
- "Define the data set profiles" on page 151
- "Define the z/OS UNIX program controlled files for RSE" on page 155
- "Verify the security settings" on page 156

---

## Activate the security settings and classes

Developer for System z uses a variety of security mechanisms to ensure a secure and controlled host system environment for the client. 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(OPERCMD5)
  - SETROPTS CLASSACT(OPERCMD5) RACLIST(OPERCMD5)
- Activate application protection for RSE
  - SETROPTS GENERIC(APPL)
  - SETROPTS CLASSACT(APPL) RACLIST(APPL)
- Activate secured signon using PassTickets for RSE
  - SETROPTS GENERIC(PKTDATA)
  - SETROPTS CLASSACT(PKTDATA) RACLIST(PKTDATA)
- 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 the security software. In this case, you must merge the existing \* and the new \*\* definitions. 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 program control before activating it on a production system.

- (Optional) Activate X.509 HostIdMappings and extended Port Of Entry (POE) support
  - SETROPTS GENERIC(SERVAUTH)
  - SETROPTS CLASSACT(SERVAUTH) RACLIST(SERVAUTH)

---

## Define an OMVS segment for Developer for System z users

A RACF OMVS segment or equivalent that specifies a valid non-zero z/OS UNIX user ID (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.

When using the optional Integrated Debugger, the user ID under which the application being debugged is active, and its default group also requires a valid RACF OMVS segment or equivalent.

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

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

---

## Define the Developer for System z started tasks

The following sample RACF commands create the DBGMR, JMON, and RSED started tasks, with protected user IDs ( STCDBM, STCJMON, and STCRSE) and the STCGROUP group 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 STCDBM DFLTGRP(STCGROUP) NOPASSWORD
  NAME('RDZ - DEBUG MANAGER')
  OMVS(UID(#user-id-debug) HOME(/tmp) PROGRAM(/bin/sh) )
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
•
  ADDUSER STCJMON DFLTGRP(STCGROUP) NOPASSWORD NAME('RDZ - JES JOBMONITOR')
  OMVS(UID(#user-id-jmon) HOME(/tmp) PROGRAM(/bin/sh) )
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
•
  ADDUSER STCRSE DFLTGRP(STCGROUP) NOPASSWORD NAME('RDZ - RSE DAEMON')
  OMVS(UID(#user-id-rse) HOME(/tmp) PROGRAM(/bin/sh) ASSIZEMAX(2147483647)
  )
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
•
  RDEFINE STARTED DBGMR.* DATA('RDZ - DEBUG MANAGER')
  STDATA(USER(STCDBM) GROUP(STCGROUP) TRUSTED(NO))
•
  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))
•
  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.
- RSE daemon requires a large address space size (2GB) for proper operation. Set this value in the ASSIZEMAX variable of the OMVS segment for user ID STCRSE. Setting this value ensures that RSE daemon gets the required region size, regardless of changes to MAXASSIZE in SYS1.PARMLIB(BPXPRMxx).
- RSE also requires a large number of threads for proper operation. You can set the limit in the THREADSMAX variable of the OMVS segment for user ID STCRSE. Setting the limit ensures that RSE gets the required thread limit, regardless of changes to MAXTHREADS or MAXTHREADTASKS in SYS1.PARMLIB(BPXPRMxx). To determine the correct value for the thread limit, see "Tuning considerations" in the *Host Configuration Reference* (SC14-7290).
- User ID STCJMON is another good candidate for setting THREADSMAX in the OMVS segment, because JES Job Monitor uses a thread per client connection.
- The Integrated Debugger started task (DBGMR) is used only by the optional Integrated Debugger feature.

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

```
ALTUSER STCRSE RESTRICTED
```

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

**Attention:** If you use restricted user IDs, explicitly add the permission to access a resource by using the TSO **PERMIT** or the z/OS UNIX **setfac1** commands. The resources include those resources where the Developer for System z documentation uses UACC, such as the \*\* profile in the PROGRAM class, or where it relies on common z/OS UNIX conventions, such as everyone having read and execute permission for Java libraries. Test the access before activating it on a production system.

---

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

RSE requires UPDATE access to the BPX.SERVER profile to create or delete the security environment for the client’s thread. If this profile is not defined, UID(0) is required for RSE. This step is required for clients to be able to connect.

Integrated Debugger requires UPDATE access to the BPX.SERVER profile to create or delete the security environment for the debug thread. If this profile is not defined, UID(0) is required for the STCDBM started task user ID. This permit is only required when the optional Integrated Debugger feature is used.

- RDEFINE FACILITY BPX.SERVER UACC(NONE)
- PERMIT BPX.SERVER CLASS(FACILITY) ACCESS(UPDATE) ID(STCRSE)
- PERMIT BPX.SERVER CLASS(FACILITY) ACCESS(UPDATE) ID(STCDBM)
- 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 switch might impact other z/OS UNIX applications and operations. Test the security before activating it on a production system. For more information about the different security levels, see *UNIX System Services Planning* (GA22-7800).

---

## Define the MVS program controlled libraries for RSE

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

RSE uses system (SYS1.LINKLIB), Language Environment’s runtime (CEE.SCEERUN\*) and ISPF’s TSO/ISPF Client Gateway (ISP.SISPLOAD) load library.

- 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. The profile obscures and complicates the search path used by your security software. In this case, you must merge the existing \* and the new \*\* definitions. Use the \*\* profile, as documented in *Security Server RACF Security Administrator's Guide* (SA22-7683).

The following additional prerequisite libraries must be made program controlled to support the use of optional services. This list does not include data sets that are specific to a product that Developer for System z interacts with, such as IBM File Manager.

- Alternate REXX runtime library, for SCLM Developer Toolkit
  - REXX.\*.SEAGALT
- System load library, for SSL encryption
  - SYS1.SIEALNKE
- Developer for System z library, for Integrated Debugger
  - FEK.SFEKAUTH

**Note:** Libraries that are designed for LPA placement also require program control authorizations if they are accessed through LINKLIST or STEPLIB. This publication documents the usage of the following LPA libraries:

- ISPF, for TSO/ISPF Client Gateway
  - ISP.SISPLPA
- REXX runtime library, for SCLM Developer Toolkit
  - REXX.\*.SEAGLPA
- Developer for System z, for CARMA
  - FEK.SFEKLPA

---

## Define the PassTicket support for RSE

The client's password or other means of identification, such as an X.509 certificate is used only to verify the identity upon connection. Afterward, PassTickets are used to maintain thread security. This step is required for clients to be able to connect.

PassTickets are system-generated passwords with a lifespan of about 10 minutes. The generated PassTickets are based on a secret key. This key is a 64-bit number (16 hexadecimal characters). In the following sample RACF commands, replace the key16 placeholder with a user-supplied 16-character hexadecimal string that has 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)
```
-

```
SETRPTS RACLIST(PTKTDATA) REFRESH
```

RSE supports the using of an application ID other than FEKAPPL. Uncomment and customize the "APPLID=FEKAPPL" option in `rsed.envvars` to activate this, as documented in "Defining extra Java startup parameters with `_RSE_JAVAOPTS`" on page 45. The PTKTDATA class definitions must match the actual application ID used by RSE.

You should not use OMVSAPPL as application ID, because it will open the secret key to most z/OS UNIX applications. You should also not use the default MVS application ID, which is MVS followed by the system's SMF ID, because this will open the secret key to most MVS applications, including user batch jobs.

**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.
- Substitute the wildcard (\*) in the IRRPTAUTH.FEKAPPL.\* definition with a valid user ID mask to limit the user IDs for which RSE can generate a PassTicket.
- Depending on your RACF settings, the user defining a profile might also be on the access list of the profile. Remove this permission for the PTKTDATA profiles.
- JES Job Monitor and RSE must have the same application ID to allow JES Job Monitor to evaluate the PassTickets presented by RSE. For JES Job Monitor, the application ID is set in the FEJJCENFG configuration file with the APPLID directive.
- If the system has a cryptographic product installed and available, you can encrypt the secured signon application key for added protection. To do so, use the KEYENCRYPTED keyword instead of KEYMASKED. For more information, see *Security Server RACF Security Administrator's Guide (SA22-7683)*.

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

---

## Define the 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')
```
- ```
SETRPTS RACLIST(APPL) REFRESH
```

**Note:**

- As described in more detail in "Define the PassTicket support for RSE" on page 148, RSE supports the using of an application ID other than FEKAPPL. The APPL class definition must match the actual application ID used by RSE.
- The client connection request succeeds if the application ID is not defined in the APPL class.
- The client connection request will fail only if the application ID is defined and the user lacks READ access to the profile.

## Define the 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 “FEJJCNFG, the JES Job Monitor configuration file” on page 30.

The following sample RACF commands give Developer for System z users conditional access to a limited set of JES commands, which are Hold, Release, Cancel, and Purge. Users have only 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 the security before activating it on a production system.

Table 21 and Table 22 on page 151 show the operator commands issued for JES2 and JES3, and the discrete security profiles that can be used to protect them.

*Table 21. JES2 Job Monitor operator commands*

| Action  | Command                               | OPERCMDs profile                                                                    | Required access |
|---------|---------------------------------------|-------------------------------------------------------------------------------------|-----------------|
| Hold    | \$Hx(jobid)<br>with x = {J, S or T}   | jesname.MODIFYHOLD.BAT<br>jesname.MODIFYHOLD.STC<br>jesname.MODIFYHOLD.TSU          | UPDATE          |
| Release | \$Ax(jobid)<br>with x = {J, S or T}   | jesname.MODIFYRELEASE.BAT<br>jesname.MODIFYRELEASE.STC<br>jesname.MODIFYRELEASE.TSU | UPDATE          |
| Cancel  | \$Cx(jobid)<br>with x = {J, S or T}   | jesname.CANCEL.BAT<br>jesname.CANCEL.STC<br>jesname.CANCEL.TSU                      | UPDATE          |
| Purge   | \$Cx(jobid),P<br>with x = {J, S or T} | jesname.CANCEL.BAT<br>jesname.CANCEL.STC<br>jesname.CANCEL.TSU                      | UPDATE          |

Table 22. JES3 Job Monitor operator commands

| Action  | Command      | OPERCMD5 profile   | Required access |
|---------|--------------|--------------------|-----------------|
| Hold    | *F,J=jobid,H | jesname.MODIFY.JOB | UPDATE          |
| Release | *F,J=jobid,R | jesname.MODIFY.JOB | UPDATE          |
| Cancel  | *F,J=jobid,C | jesname.MODIFY.JOB | UPDATE          |
| Purge   | *F,J=jobid,C | jesname.MODIFY.JOB | UPDATE          |

**Note:**

- The Hold, Release, Cancel, and Purge JES operator commands, and the Show JCL command, can be executed only against spool files owned by the client user ID, unless LIMIT\_COMMANDS= with value LIMITED or NOLIMIT is specified in the JES Job Monitor configuration file. For more information, see "Actions against jobs - target limitations" in the *Host Configuration Reference* (SC14-7290).
- Users can browse any spool file, unless LIMIT\_VIEW=USERID is defined in the JES Job Monitor configuration file. For more information, see "Access to spool files" in *Host Configuration Reference* (SC14-7290).
- Even if users are not authorized for these operator commands, they will still be able to submit jobs and read job output through JES Job Monitor if they have sufficient authority to possible profiles that protect these resources, such as those in the JESINPUT, JESJOBS and JESSPOOL classes.

Assuming the identity of the JES Job Monitor server by creating a JMON console from a TSO session is prevented by your security software. Even though the console can be created, the point of entry is different; for example, JES Job Monitor versus TSO. JES commands issued from this console will fail the security check if your security is set up as documented in this publication and the user does not have authority to the JES commands through other means.

---

## Define the data set profiles

READ access for users and ALTER for system programmers is sufficient for most Developer for System z data sets. Replace the #sysprog placeholder with valid user IDs or RACF group names. Also, ask the system programmer who installed and configured the product for the correct data set names. FEK is the default high-level qualifier used during installation and FEK.#CUST is the default high-level qualifier for data sets created during the customization process.

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

**Note:**

- Protect FEK.SFEKAUTH against updates because this data set is APF-authorized. The same is true for FEK.SFEKLOAD and FEK.SFEKLPA, but here because these data sets are program controlled.
- The sample commands in this publication and in the FEKRACF job assume that Enhanced Generic Naming (EGN) is active. When EGN is active, the \*\* qualifier can be used to represent any number of qualifiers in the DATASET class. Substitute \*\* with \* if EGN is not active on your system. For more information about EGN, see *Security Server RACF Security Administrator's Guide (SA22-7683)*.

Some of the optional Developer for System z components require additional security data set profiles. Replace the #sysprog, #ram-developer and #cicsadmin placeholders with valid user ID's or RACF group names:

- If SCLM Developer Toolkit's long/short name translation is used, users require UPDATE access to the mapping VSAM, FEK.#CUST.LSTRANS.FILE.

```

-
  ADDSD 'FEK.#CUST.LSTRANS*.**' UACC(UPDATE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z - SCLMDT')
-
  PERMIT 'FEK.#CUST.LSTRANS*.**' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
  SETROPTS GENERIC(DATASET) REFRESH

```

- CARMA RAM (Repository Access Manager) developers require UPDATE access to the CARMA VSAMs, FEK.#CUST.CRA\*.

```

-
  ADDSD 'FEK.#CUST.CRA*.**' UACC(READ)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z - CARMA')
-
  PERMIT 'FEK.#CUST.CRA*.**' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
  PERMIT 'FEK.#CUST.CRA*.**' CLASS(DATASET) ACCESS(UPDATE) ID(#ram-developer)
-
  SETROPTS GENERIC(DATASET) REFRESH

```

- If Application Deployment Manager's CICS Resource Definition (CRD) server is used, CICS administrators require UPDATE access to the CRD repository VSAM.

```

-
  ADDSD 'FEK.#CUST.ADNREP*.**' UACC(READ)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z - ADN')
-
  PERMIT 'FEK.#CUST.ADNREP*.**' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
  PERMIT 'FEK.#CUST.ADNREP*.**' CLASS(DATASET) ACCESS(UPDATE) ID(#cicsadmin)
-
  SETROPTS GENERIC(DATASET) REFRESH

```

- If Application Deployment Manager's manifest repository is defined, all of the CICS Transaction Server users require UPDATE access to the manifest repository VSAM.

```

-
  ADDSD 'FEK.#CUST.ADNMAN*.**' UACC(UPDATE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z - ADN')
-

```

```

PERMIT 'FEK.#CUST.ADNMAN*.*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
SETROPTS GENERIC(DATASET) REFRESH

```

Use the following sample RACF commands for a more secure setup where READ access is also controlled.

- uacc (none) data set protection

```

-
ADDGROUP (FEK)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z - HLQ STUB')
OWNER(IBMUSER) SUPGROUP(SYS1)"
-
ADDSO 'FEK.*.*' UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
-
ADDSO 'FEK.SFEKAUTH' UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
-
ADDSO 'FEK.SFEKLOAD' UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
-
ADDSO 'FEK.SFEKMOD' UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
-
ADDSO 'FEK.SFEKPROC' UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
-
ADDSO 'FEK.#CUST.PARMLIB' UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
-
ADDSO 'FEK.#CUST.CNTL' UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
-
ADDSO 'FEK.#CUST.SQL' UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
-
ADDSO 'FEK.#CUST.LSTRANS*.*' UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z - SCLMDT')
-
ADDSO 'FEK.#CUST.CRA*.*' UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z - CARMA')
-
ADDSO 'FEK.#CUST.ADNREP*.*' UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z - ADN')
-
ADDSO 'FEK.#CUST.ADNMAN*.*' UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z - ADN')

```

- Permit system programmer to manage all libraries

```

-
PERMIT 'FEK.*.*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
PERMIT 'FEK.SFEKAUTH' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)

```

```

-
PERMIT 'FEK.SFEKLOAD CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
PERMIT 'FEK.SFEKLMOD CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
PERMIT 'FEK.SFEKPROC CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
PERMIT 'FEK.#CUST.PARMLIB CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
PERMIT 'FEK.#CUST.CNTL CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
PERMIT 'FEK.#CUST.SQL CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
PERMIT 'FEK.#CUST.LSTRANS.*.**' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
PERMIT 'FEK.#CUST.CRA*.**' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
PERMIT 'FEK.#CUST.ADNREP*.**' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
PERMIT 'FEK.#CUST.ADNMAN*.**' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)

```

- Permit clients to access the load and exec libraries

```

-
PERMIT 'FEK.SFEKAUTH' CLASS(DATASET) ACCESS(READ) ID(*)
-
PERMIT 'FEK.SFEKLOAD' CLASS(DATASET) ACCESS(READ) ID(*)
-
PERMIT 'FEK.SFEKPROC' CLASS(DATASET) ACCESS(READ) ID(*)
-
PERMIT 'FEK.#CUST.CNTL' CLASS(DATASET) ACCESS(READ) ID(*)
-
PERMIT 'FEK.#CUST.SQL' CLASS(DATASET) ACCESS(READ) ID(*)

```

**Note:** No permits are needed for FEK.SFEKLPA because all code that resides in LPA is accessible by everyone.

- Permit Integrated Debugger to access the load library.
  - PERMIT 'FEK.SFEKAUTH' CLASS(DATASET) ACCESS(READ) ID(STCDBM)
- Permit JES Job Monitor to access the load and parameter library
  - PERMIT 'FEK.SFEKAUTH' CLASS(DATASET) ACCESS(READ) ID(STCJMON)
  - PERMIT 'FEK.#CUST.PARMLIB' CLASS(DATASET) ACCESS(READ) ID(STCJMON)
- (Optional) Permit clients to update the long/short name translation VSAM for SCLMDT
  - PERMIT 'FEK.#CUST.LSTRANS.\*.\*\*' CLASS(DATASET) ACCESS(UPDATE) ID(\*)
- (Optional) Permit RAM developers to update the CARMA VSAMs for CARMA
  -

- ```

        PERMIT 'FEK.#CUST.CRA*.*' CLASS(DATASET) ACCESS(UPDATE) ID(#ram-developer)
    
```
- (Optional) Permit CICS users to read the CRD repository VSAM for Application Deployment Manager
- ```

    -
        PERMIT 'FEK.#CUST.ADNREP*.*' CLASS(DATASET) ACCESS(READ) ID(*)
    
```
- (Optional) Permit CICS administrators to update the CRD repository VSAM for Application Deployment Manager
- ```

    -
        PERMIT 'FEK.#CUST.ADNREP*.*' CLASS(DATASET) ACCESS(UPDATE) ID(#cicsadmin)
    
```
- (Optional) Permit CICS users to update the manifest repository VSAM for Application Deployment Manager
- ```

    -
        PERMIT 'FEK.#CUST.ADNMAN*.*' CLASS(DATASET) ACCESS(UPDATE) ID(*)
    
```
- (Optional) Permit CICS TS server to access the load library for bidi and Application Deployment Manager
- ```

    -
        PERMIT 'FEK.SFEKLOAD' CLASS(DATASET) ACCESS(READ) ID(#cicsts)
    
```
- (Optional) Permit CICS TS server, IMS regions, and MVS batch jobs to access the load library for IRZ messages
- ```

    -
        PERMIT 'FEK.SFEKLMOD' CLASS(DATASET) ACCESS(READ) ID(#cicsts)
        PERMIT 'FEK.SFEKLMOD' CLASS(DATASET) ACCESS(READ) ID(#ims)
        PERMIT 'FEK.SFEKLMOD' CLASS(DATASET) ACCESS(READ) ID(#batch)
    
```
- Activate security profiles
- ```

    -
        SETROPTS GENERIC(DATASET) REFRESH
    
```

When controlling READ access to system data sets, you must provide Developer for System z servers and users permission to READ the following data sets:

- CEE.SCEERUN
- CEE.SCEERUN2
- CBC.SCLBDLL
- ISP.SISPLoad
- ISP.SISPLPA
- SYS1.LINKLIB
- SYS1.SIEALNKE
- SYS1.SIEAMIGE
- REXX.V1R4M0.SEAGLPA

**Note:** When you use the Alternate Library for REXX product package, the default REXX runtime library name is REXX.\*.SEAGALT instead of REXX.\*.SEAGLPA, as used in the sample above.

---

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

Servers with authority to BPX.SERVER must run in a clean, program-controlled environment. This requirement implies that all programs called by RSE 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).

RSE server uses RACF's Java shared library (/usr/lib/libIRRracf\*.so).

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

**Note:**

- Since z/OS 1.9, /usr/lib/libIRRracf\*.so is installed in program controlled mode during SMP/E RACF installation.
- Since z/OS 1.10, /usr/lib/libIRRracf\*.so is part of SAF, which is provided 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. For more information, consult the documentation of your security product.
- The SMP/E installation of Developer for System z sets the program control bit for internal RSE 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** is displayed 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 the 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
- Started tasks
  - LISTGRP STCGROUP OMVS
  - LISTUSER STCDBM OMVS
  - LISTUSER STCJMON OMVS
  - LISTUSER STCRSE OMVS
  - RLIST STARTED DBGMGR.\* ALL STDATA
  - RLIST STARTED JMON.\* ALL STDATA
  - RLIST STARTED RSED.\* ALL STDATA
- RSE as a secure z/OS UNIX server
  - RLIST FACILITY BPX.SERVER ALL
- MVS program controlled libraries for RSE
  - RLIST PROGRAM \*\* ALL
- PassTicket support for RSE
  - RLIST PTKTDATA FEKAPPL ALL SSIGNON
  - RLIST PTKTDATA IRRPTAUTH.FEKAPPL.\* ALL
- Application protection for RSE
  - RLIST APPL FEKAPPL ALL
- JES command security
  - RLIST CONSOLE JMON ALL
  - RLIST OPERCMDS MVS.MCSOPER.JMON ALL

- RLIST OPERCMDS JES%.\*\* ALL
- Data set profiles
  - LISTGRP FEK
  - LISTDSD PREFIX(FEK) ALL
- z/OS UNIX program controlled files for RSE
  - ls -E /usr/lib/libIRRRacf\*.so

Optionally, profiles can exist that direct the Developer for System z behavior for a specific user. These profiles match the FEK.\*\* filter and are by default located in the FACILITY class. See the `_RSE_FEK_SAF_CLASS` directive in `rzed.envvars`. You can use the **SEARCH** command to list the profile names. Use the **RLIST** command to show the details for a profile.

- SEARCH CLASS(FACILITY) FILTER(FEK.\*\*)
- RLIST FACILITY #profile-name ALL



---

## Chapter 10. Migration guide

---

### Migration considerations

This section highlights installation and configuration changes compared to previous releases of the product. It also gives some general guidelines to migrate to this release. For more information, see the related sections in this manual.

- If you are a previous user of IBM Rational Developer for System z, IBM WebSphere Developer for System z, IBM WebSphere Developer for zSeries or IBM WebSphere® Studio Enterprise Developer, save the related customized files before upgrading to this version of IBM Rational Developer for System z Version 9.0.
- If you plan on running multiple instances of Developer for System z, read "Running multiple instances" in the *Host Configuration Reference* (SC14-7290).
- If your migration scenario spans more than two releases, you should do the customizations again, as if there is no older release present.

### Backing up the previously configured files

If you are a previous user of Developer for System z, save the related customized files before installing this version of IBM Developer for System z.

Customizable Developer for System z files can be found at the following locations:

- Version and 8.0.1 and 8.5
  - FEK.#CUST.RDZ\*.\*, Configuration Utility work files
  - FEK.SFEKSAMP, some members are copied to FEK.#CUST.\* by the FEKSETUP sample job, where \* equals PARMLIB, PROCLIB, JCL, CNTL, ASM and COBOL
  - FEK.SFEKSAMV
  - /usr/lpp/rdz/samples/, some files are copied to /etc/rdz/ and /var/rdz/sclmdt/\* by the FEKSETUP sample job, where \* equals CONFIG/, CONFIG/PROJECT/ and CONFIG/script/

Previous Developer for System z setups also document changes to configuration files owned by other products.

- Version 8.0.1 and 8.5
  - SYS1.PARMLIB(BPXPRMxx)  
Set z/OS UNIX system defaults.
  - SYS1.PARMLIB(COMMNDxx)  
Start servers at IPL time.
  - SYS1.PARMLIB(LPALSTxx)  
Add FEK.SFEKLPA to LPA.
  - SYS1.PARMLIB(PROGxx)  
Add FEK.SFEKAUTH and FEK.SFEKLOAD to LINKLIST.
  - (WLM)  
Assign an application environment for a DB2 stored procedure.

---

## Version 9.0 migration notes

The following migration notes are specific to IBM Rational Developer for System z version 9.0. These notes are valid for migration from IBM Rational Developer for System z version 9.0.0 to version 9.0.1, and they are additions to the existing version 9.0.0 migration notes.

All of the listed changes are valid since version 9.0.1.

### IBM Rational Developer for System z, FMID HHOP900

- CARMA: The CRADEF VSAM file for the CA Endeavor<sup>®</sup> SCM RAM has been updated.
- CARMA: The CRASTART load module, which resides in LPA, has been updated, requiring an LPA update.
- CARMA: Added support to execute a user exit during CARMA startup.
- CARMA: Added support for RAMs processing startup arguments.
- CARMA: New customizable members have been added:
  - CRAEXIT: Sample CARMA user exit.
  - CRAALLOC: Allocation exec for custom RAM CARMA invocations.
  - CRACFG: CA Endeavor<sup>®</sup> SCM RAM usage configuration file.
- CARMA: The following customizable members have changed:
  - CRASRV.properties
  - crastart.conf
  - crastart.endevor.conf
  - CRASUBMT
  - CRASUBCA
  - CRANDVRA
- CARMA: Additional DD statements are added for the CA Endeavor<sup>®</sup> SCM RAM in crastart.endevor.conf and CRASUBCA:
  - CRAPARM, which is allocated by CRANDVRA
  - CRACFG
- CARMA: Additional DD statements are added for the non-“CA Endeavor<sup>®</sup> SCM RAM” in crastart.conf and CRASUBMT:
  - CRAPARM, which is allocated by CRAALLOC
- Customization: The FEKSETUP JCL now processes the new members:
  - CRACFG: copied to FEK.#CUST.PARMLIB(CRACFG)
  - AQESTC: copied to FEK.#CUST.PROCLIB(DBGMGR)
  - AQECSD: copied to FEK.#CUST.JCL(AQECSD)
- Integrated Debugger: New optional service
  - IEASVCxx, LPALSTxx and PROGxx (APF and LINKLIST) parmlib updates
  - DBGMGR: started task JCL
  - AQECSD: sample JCL to update CICS CSD
  - AQERACF: sample JCL to do security setup for just Integrated Debugger
- RSE: Updated PROCLIB members
  - ELAXFGO
- RSE: New optional directives have been added to rsecomm.properties:
  - USER

- RSE: New operator commands
  - F rsed,APPL=TRACE {USER | SERVER | CLEAR}
- RSE: New optional directives have been added to rsed.envvars:
  - (\_RSE\_JAVAOPTS) -Dsearch.server.limit.timeout
  - (\_RSE\_JAVAOPTS) -Dkeep.all.logs
  - (\_RSE\_JAVAOPTS) -Daudit.users
  - RSE\_UBLD\_DD
  - RSE\_UBLD\_STEPLIB
- zUnit: New optional startup arguments have been added:
  - CLOCALE / -l

**Note:** To simplify migration from an existing Developer for System z setup without the Integrated Debugger, sample JCL FEK.SFEKSAMP(AQERACF) with RACF commands is provided to define just the Integrated Debugger related security definitions.

## IBM Rational Developer for System z Host Utilities, FMID HAKG900

There are no version 9.0 specific migration notes for this product.

---

### Migrate from version 8.5 to version 9.0

These notes are for a migration from a base version 8.5 to version 9.0. It includes changes that are already documented as part of version 8.5 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 HHOP900

- The default SMP/E install location for MVS and z/OS UNIX components did not change and thus remain FEK.\* and /usr/lpp/rdz/\*.
- CARMA: The CRADEF and CRASTRS VSAM files for the CA Endeavor® SCM RAM must be updated to use the new support for customizable CA Endeavor® SCM batch-actions (since version 8.5.1).
- CARMA: Added support to disable a RAM during CRADEF VSAM creation (since version 8.5.1).
- CARMA: Added support for non-absolute file references in CRASRV.properties (since version 8.5.1).
- CARMA: New sample members have been added:
  - CRABJOB: Default JOB card for CA Endeavor® SCM batch-actions (since version 8.5.1).
- CARMA: The following customizable members have changed:
  - CRASRV.properties (since version 8.5.1)
  - carma.startup.rex (since version 8.5.1)
  - CRA\$VCAD (since version 8.5.1)
  - CRA\$VDEF (since version 8.5.1)
  - CRABATCA (since version 8.5.1)
  - CRABCFG (since version 8.5.1)
  - CRANDVRA (since version 8.5.1)

- CARMA: Additional DD statements are added for the CA Endeavor<sup>®</sup> SCM RAM in crastart.endevor.conf and CRASUBCA:
  - CRABJCLO, which is allocated by CRANDVRA (since version 8.5.1)
  - ENHCEDIT, which is allocated by CRANDVRA (since version 8.5.1)
- Customization: The FEKSETUP JCL now processes the new members:
  - CRABJOB: copied to FEK.#CUST.CNTL(CRABJOB) (since version 8.5.1)
  - ELAXFSP: copied to FEK.#CUST.PROCLIB(ELAXFSP) (since version 9.0)
  - ELAXFSQL: copied to FEK.#CUST.PROCLIB(ELAXFSQL) (since version 9.0)
  - FEKTEP2: copied to FEK.#CUST.SQL(FEKTEP2) (since version 9.0)
  - FEKTIAD: copied to FEK.#CUST.SQL(FEKTEP2) (since version 9.0)
- Fault Analyzer Integration: support for FAI has been discontinued. This change is incompatible with older clients still using FAI.
- JES Job Monitor - New operator commands have been added to the JMON started task:
  - MODIFY USERS (since version 8.5.1)
  - MODIFY -T{N | E | I | V} (since version 8.5.1)
  - MODIFY -M{N | E | W | I | V} (since version 8.5.1)
  - MODIFY TRACE {N | E | I | V} (since version 9.0)
  - MODIFY MESSAGE {N | E | W | I | V} (since version 9.0)
- JES Job Monitor - New optional directives have been added to FEJJCENFG:
  - LOOPBACK\_ONLY (since version 9.0)
- JES Job Monitor - Optional directives have been removed from FEJJCENFG:
  - \_BPXK\_SETIBMOPT\_TRANSPORT (since version 9.0)
- Problem determination: The FEKLOGS JCL now supports specifying multiple user IDs for gathering user logs (since version 8.5.1).
- Problem determination: The FEKLOGS JCL now uses DD REFORMAT to collect logs reformatted for quicker problem determination (since version 8.5.1).
- Problem determination : The following customizable members have changed:
  - FEKLOGS (since version 8.5.1)
- RSE - New operator commands have been added to the RSED started task:
  - MODIFY DISPLAY OWNER,DATASET=dataset (since version 9.0)
  - MODIFY DEBUG GC,PID=pid (since version 9.0)
- RSE: New non-customizable directives have been added to rsed.envvars:
  - \_CMDSERV\_BASE\_HOME (since version 8.5.1)
  - \_CMDSERV\_CONF\_HOME (since version 8.5.1)
  - \_CMDSERV\_WORK\_HOME (since version 8.5.1)
  - RSE\_DSN\_SFELKLOAD (since version 9.0)
  - (\_RSE\_JAVAOPTS) -Dlock.info.timeout (since version 9.0)
  - (\_RSE\_JAVAOPTS) -DDSTORE\_INITIAL\_SIZE (since version 9.0)
  - (\_RSE\_JAVAOPTS) -DDSTORE\_MAX\_FREE (since version 9.0)
- RSE: New required directives have been added to rsed.envvars:
  - RSE\_HLQ (since version 9.0)
- RSE: New optional directives have been added to rsed.envvars:
  - (\_RSE\_JAVAOPTS) -DRSE\_DSICALL (since version 8.5.1)
  - (\_RSE\_JAVAOPTS) -DDISABLE\_REMOTE\_INDEX\_SEARCH (since version 8.5.1)

- (\_RSE\_JAVAOPTS) -DDISABLE\_TEXT\_SEARCH (since version 9.0)
- (\_RSE\_JAVAOPTS) -Dsearch.server.limit.hits (since version 9.0)
- (\_RSE\_JAVAOPTS) -Dsearch.server.limit.datasets (since version 9.0)
- (\_RSE\_JAVAOPTS) -Dsearch.server.limit.lines (since version 9.0)
- (\_RSE\_JAVAOPTS) -DDSTORE\_SSL\_ALGORITHM (since version 9.0)
- RSE: The default value for non-customizable directives in rsed.envvars has changed:
  - (\_RSE\_JAVAOPTS) -DSPIRIT\_EXPIRY\_TIME (since version 9.0)
- RSE: The default value for optional directives in rsed.envvars has changed:
  - (\_RSE\_JAVAOPTS) -Xms (since version 8.5.1)
  - (\_RSE\_JAVAOPTS) -Xmx (since version 8.5.1)
  - (\_RSE\_JAVAOPTS) -Dmaximum.clients (since version 8.5.1)
  - (\_RSE\_JAVAOPTS) -Dmaximum.threads (since version 8.5.1)
  - CGI\_ISPPREF (since version 9.0)
- Security: Support for new security profiles has been added:
  - FEK.USR.\*\* (since version 8.5.1)

## Configurable files

Table 23 shows an overview of files that are customized in version 9.0. The Developer for System z sample libraries, FEK.SFEKSAMP, FEK.SFEKSAMV and /usr/lpp/rdz/samples/, contain more customizable members than listed here, such as sample CARMA source code and jobs to compile them. The Developer for System z Host Utilities sample libraries, AKG.SAKGSAMP and /usr/lpp/rdzutil/samples, contain more customizable members than listed here, such as the sample code review post-processing script.

The following members and files are no longer customizable or no longer used:

- LOCKD started task
- ELAXMSAM sample DB2 stored procedure
- ELAXMJCL sample JCL for DB2 stored procedure

**Note:** Sample job FEKSETUP copies all listed members to different data sets and directories, default FEK.#CUST.\* and /etc/rdz/\*.

Table 23. Version 9.0 customizations

Member/File	Default location	Purpose	Migration notes
FEKSETUP	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to create data sets and directories, and populate them with customizable files	Updated to remove actions for files that are no longer used and add actions for new files
JMON	FEK.SFEKSAMP(FEJJJCL) [FEK.#CUST.PROCLIB]	JCL for JES Job Monitor	None
FEJJJCL	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(JMON)]	Name for JMON member	See JMON member
RSED	FEK.SFEKSAMP(FEKRSED) [FEK.#CUST.PROCLIB]	JCL for RSE daemon	None

Table 23. Version 9.0 customizations (continued)

Member/File	Default location	Purpose	Migration notes
FEKRSED	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(RSED)]	Name for RSED member	See RSED member
ELAXF*	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	JCL for remote project builds, and so on	ELAXFSP and ELAXFSQL are new, ELAXFCOC and ELAXFCP1 are updated for Cobol Version 5 support
FEKRACF	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL for security definitions	None
FEJJCNFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	JES Job Monitor configuration file	New optional directives have been added. Existing optional directives have been removed.
FEJTSO	FEK.SFEKSAMP [FEK.#CUST.CNTL]	JCL for TSO submits	None
CRA\$VMMSG	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to create the CARMA message VSAM	None
CRA\$VDEF	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to create the CARMA configuration VSAM	Added support to exclude RAMs
CRA\$VSTR	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to create the CARMA custom information VSAM	None
CRA\$VCAD	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to create the CARMA configuration VSAM for CA Endeavor <sup>®</sup> SCM RAM	Added support to exclude RAMs and VSAM input has changed.
CRA\$VCAS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to create the CARMA custom information VSAM for CA Endeavor <sup>®</sup> SCM RAM	VSAM input has changed

Table 23. Version 9.0 customizations (continued)

Member/File	Default location	Purpose	Migration notes
CRASUBMT	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CARMA batch startup CLIST	None
CRASUBCA	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CARMA batch startup CLIST for CA Endeavor® SCM RAM	None
CRABCFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CARMA batch actions configuration for CA Endeavor® SCM RAM	New directives added
CRABATCA	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CARMA batch action JCL for CA Endeavor® SCM RAM	Added support for variable JOB card
CRASHOW	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CARMA configuration for CA Endeavor® SCM RAM	None
CRATMAP	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CARMA configuration for CA Endeavor® SCM RAM	None
CRANDVRA	FEK.SFEKPROC	CARMA allocation REXX for CA Endeavor® SCM RAM	Added new DD allocations
CRA#VSLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to create the SCLM RAM's message VSAM	None
CRA#ASLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to create the SCLM RAM's data sets	None
CRA#VPDS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to create the PDS RAM's message VSAM	None
CRA#UADD	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to merge RAM definitions	None

Table 23. Version 9.0 customizations (continued)

Member/File	Default location	Purpose	Migration notes
CRA#UQRY	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to extract RAM definitions	None
CRAXJCL	FEK.SFEKSAMP [FEK.#CUST.ASM]	Sample source code for IRXJCL replacement	None
CRA#CIRX	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to compile CRAXJCL	None
ADNCSDRS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to define the RESTful CRD server to primary CICS region	None
ADNCSDTX	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to define alternate transaction IDs to CICS region	None
ADNTXNC	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to create alternate transaction IDs	None
ADNMSGHC	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to compile ADNMSGHS	None
ADNMSGHS	FEK.SFEKSAMP [FEK.#CUST.COBOL]	Sample source code for the Pipeline Message Handler	None
ADNVCRD	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to create the CRD repository	None
ADNCSDWS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to define the Web Service CRD server to primary CICS region	None
ADNCSDAR	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to define the CRD server to non-primary CICS regions	None
ADNJSPAU	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to update the CRD defaults	None

Table 23. Version 9.0 customizations (continued)

Member/File	Default location	Purpose	Migration notes
ADNMFST	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to create and define the Manifest repository	None
FEKTEP2	FEK.SFEKSAMP [FEK.#CUST.SQL]	SQL command input used by ELAXF*	New, customization is optional
FEKTIAD	FEK.SFEKSAMP [FEK.#CUST.JCL]	SQL command input used by ELAXF*	New, customization is optional
AZUZUNIT	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	JCL procedure for zUnit	None
FEKRPLI	FEK.SFEKSAMP [FEK.#CUST.CNTL]	REXX to call the PL/I compiler from within the preprocessor framework	None
FEKLOGS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to collect log files	Added additional checks. Any customization to older files must be done again.
rseed.envvars	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE environment variables	Older copies must be replaced by this one and the customizations done again.
ISPF.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	TSO/ISPF Client Gateway configuration file	None
CRASRV.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	CARMA configuration file	Added support for default values
crastart.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	CARMA configuration file for CRASTART usage	None

Table 23. Version 9.0 customizations (continued)

Member/File	Default location	Purpose	Migration notes
crastart.endevor.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	CARMA configuration file for CRASTART usage for CA Endeavor® SCM RAM	None
include.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	Forced includes for C/C++ content assist	None
ssl.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE SSL configuration file	None
rsecomm.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE trace configuration file	None
pushtoclient.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	Push information to the client configuration file	None

## IBM Rational Developer for System z Host Utilities, FMID HAKG900

There are no migration notes, because there is no equivalent function in version 8.5.

### Configuration files

Table 24 shows an overview of files that are customized in version 9.0. The Developer for System z Host Utilities sample libraries, AKG.SAKGSAMP and /usr/lpp/rdzut il/samples, contain more customizable members than listed here, such as sample code review post-processing script.

**Note:** Sample job AKGSETUP copies all listed members to different data sets, default AKG.#CUST.\*.

Table 24. Host Utilities version 9.0 customizations. Host Utilities version 9.0 customizations

Member or File	Default location	Purpose	Migration notes
AKGSETUP	AKG.SAKGSAMP [AKG.#CUST.JCL]	JCL to create data sets, and populate them with customizable files	New
AKGCC	AKG.SAKGSAMP [AKG.#CUST.PROCLIB]	JCL for code coverage	New
AKGCR	AKG.SAKGSAMP [AKG.#CUST.PROCLIB]	JCL for code review	New
AKGCRADD	AKG.SAKGSAMP [AKG.#CUST.JCL]	JCL to add third-party code to code review	New

---

## Version 8.5 migration notes

The following migration notes are specific to version 8.5. These notes are valid for migration from IBM Rational Developer for System z version 8.5.0 to version 8.5.1, and they are additions to the existing version 8.5.0 migration notes.

All of these listed changes are valid since version 8.5.1.

- CARMA: The CRADEF and CRASTRS VSAM files for the CA Endeavor<sup>®</sup> SCM RAM must be updated to use the new support for customizable CA Endeavor<sup>®</sup> SCM batch-actions.
- CARMA: Added support to disable a RAM during CRADEF VSAM creation.
- CARMA: Added support for non-absolute file references in CRASRV.properties.
- CARMA: New sample members have been added:
  - CRABJOB: Default JOB card for CA Endeavor<sup>®</sup> SCM batch-actions.
- CARMA: The following customizable members have changed:
  - CRASRV.properties
  - carma.startup.rex
  - CRA\$VCAD
  - CRA\$VDEF
  - CRABATCA
  - CRABCFG
  - CRANDVRA
- CARMA: Additional DD statements are added for the CA Endeavor<sup>®</sup> SCM RAM in crastart.endevor.conf and CRASUBCA:
  - CRABJCLO, which is allocated by CRANDVRA
  - ENHCEDIT, which is allocated by CRANDVRA
- Customization: The FEKSETUP JCL now processes the new members:
  - CRABJOB: copied to FEK.#CUST.CNTL(CRABJOB).
- JES Job Monitor - New operator commands have been added to the JMON started task (since version 8.0.3.2):
  - MODIFY USERS
  - MODIFY -T{N | E | I | V}
  - MODIFY -M{N | E | W | I | V}
- Problem determination: The FEKLOGS JCL now supports specifying multiple user IDs for gathering user logs.
- Problem determination: The FEKLOGS JCL now uses DD REFORMAT to collect logs reformatted for quicker problem determination.
- Problem determination : The following customizable members have changed:
  - FEKLOGS
- RSE: New non-customizable directives have been added to rsed.envvars:
  - \_CMDSERV\_BASE\_HOME
  - \_CMDSERV\_CONF\_HOME
  - \_CMDSERV\_WORK\_HOME
- RSE: New optional directives have been added to rsed.envvars:
  - (\_RSE\_JVAOPTS) -DRSE\_DSICALL
  - (\_RSE\_JVAOPTS) -DDISABLE\_REMOTE\_INDEX\_SEARCH

- RSE: The default value for optional directives in `rseed.envvars` has changed:
  - `(_RSE_JAVAOPTS) -Xms`
  - `(_RSE_JAVAOPTS) -Xmx`
  - `(_RSE_JAVAOPTS) -Dmaximum.clients`
  - `(_RSE_JAVAOPTS) -Dmaximum.threads`
- RSE: The default value for non-customizable directives in `rseed.envvars` has changed:
  - `(_RSE_JAVAOPTS) -DDSTORE_SPIRIT_ON`
- Security: Support for new security profiles has been added:
  - `FEK.USR.**`

---

## Migrate from version 8.0.1 to version 8.5

These notes are for a migration from a base version 8.0.1 to version 8.5. It includes changes that are already documented as part of version 8.0.1 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 HHOP850

- The default SMP/E install location for MVS and z/OS UNIX components did not change and thus remain `FEK.*` and `/usr/lpp/rdz/*`.
- CARMA - The CRASTART load module, which resides in LPA, has been updated, requiring an LPA update (since version 8.0.3.2).
- CARMA - The CRAMSG VSAM must be updated (since version 8.0.3 and 8.5).
- CARMA - The CRADEF and CRASTRS VSAM files for the CA Endeavor<sup>®</sup> SCM RAM must be updated to use the new support for CA Endeavor<sup>®</sup> SCM batch-actions (since version 8.0.3) and CA Endeavor<sup>®</sup> SCM packages (since version 8.0.3).
- CARMA - New CRADEF and CRASTRS VSAM input has been added to allow restoring CA Endeavor<sup>®</sup> SCM package actions from CA Endeavor<sup>®</sup> SCM element menus.
  - `CRA0VPKD` - to be merged in CRADEF.
  - `CRA0VPKS` - to be merged in CRASTRS.
- CARMA - New sample members have been added (since version 8.0.3):
  - `CRABCFG` - configuration file for CA Endeavor<sup>®</sup> SCM batch-actions.
  - `CRABATCA` - sample job for CA Endeavor<sup>®</sup> SCM batch-actions.
- CARMA - The following customizable members have changed (since version 8.0.3, 8.0.3.1, and 8.5):
  - `CRANDVRA`
  - `CRASHOW`
  - `CRASRV.properties`
  - `CRABCFG`
- CARMA - Additional DD statements are added for the CA Endeavor<sup>®</sup> SCM RAM in `crastart.endeavor.conf` and `CRASUBCA` (since version 8.0.3):
  - `CRABCFG`
  - `CRABSKEL`
  - `PKGSCLS` (allocated by `CRANDVRA`)
- Enterprise Service Tools - IRZ load modules and message modules moved to a new library (since version 8.5):

- FEK.SFEKLMOD(IRZ\* IIRZ\*)
- File Manager Integration is removed (since version 8.5). Some functions, such as unformatted QSAM editing, are now part of regular data set handling by Developer for System z. More advanced functions, such as formatted data editing using copybooks or include files, require the IBM File Manager plug-in for Eclipse.
- Include pre-processor - New sample members have been added (since version 8.0.3.1):
  - FEKRNPLI
- Host Configuration Utility - Added migration option (since version 8.0.2)
- JES Job Monitor - New operator commands have been added to the JMON started task (since version 8.0.3.2):
  - MODIFY STORAGE
- JES Job Monitor - New optional directives have been added to FEJCNFG (since version 8.0.3.1 and 8.0.3.2):
  - LIMIT\_CONSOLE
  - SEARCHALL
  - TRACE\_STORAGE
- PROCLIB - The following PROCLIB members have changed (since version 8.0.3):
  - ELAXFUOP
- RSE - The option to specify TMPDIR as startup argument for the RSED and LOCKD started tasks has been removed. It is replaced by a non-customizable function that defines the home directory of the started task user ID to TMPDIR if /tmp is not available for write actions (since version 8.0.3.1).
- 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, 8.0.3, and 8.0.3.2):
  - MODIFY IVP ISPF,userid
  - MODIFY IVP PASSTICKET,userid
  - MODIFY DEBUG HEAPDUMP,PID=pid
  - MODIFY DEBUG JAVACORE,PID=pid
- RSE - RSED started task operator commands have been enhanced (since version 8.0.2 and 8.0.3.1):
  - MODIFY DISPLAY CLIENT[,{LOGON | ,ID | ,USER}]
  - MODIFY DISPLAY PROCESS,CPU[,PID=pid]
- RSE - The following console messages are new (since version 8.0.3 and 8.0.3.1):
  - FEK910I = {0} IVP Exit code = {1}
  - FEK211W User, {0}, not logged on
- 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, 8.0.3.1, and 8.5):

- (\_RSE\_JAVAOPTS) -Daudit.action
- (\_RSE\_JAVAOPTS) -Daudit.action.id
- (\_RSE\_JAVAOPTS) -Dlogon.action
- (\_RSE\_JAVAOPTS) -Dlogon.action.id
- (\_RSE\_JAVAOPTS) -Dreject.logon.threshold
- (\_RSE\_JAVAOPTS) -Dinclude.c
- (\_RSE\_JAVAOPTS) -Dinclude.cpp
- (\_RSE\_JAVAOPTS) -DCPP\_CLEANUP\_INTERVAL
- (\_RSE\_JAVAOPTS) -DRIS\_BUFFER
- (\_RSE\_JAVAOPTS) -DDSTORE\_TCP\_NO\_DELAY
- \_RSE\_FEK\_SAF\_CLASS
- \_RSE\_LDAP\_SERVER
- \_RSE\_LDAP\_PORT
- \_RSE\_LDAP\_PTC\_GROUP\_SUFFIX
- CGI\_ISPPREF
- RSE - Existing required directives have been renamed (since version 8.5):
  - \_CMDSERV\_BASE\_HOME -> CGI\_ISPHOME
  - \_CMDSERV\_CONF\_HOME -> CGI\_ISPCONF
  - \_CMDSERV\_WORK\_HOME -> CGI\_ISPWORK
  - \_RSE\_CMDSERV\_OPTS -> \_RSE\_ISPF\_OPTS
- RSE - Existing optional directives have been extended with more values (since version 8.5):
  - STEPLIB
- RSE - Interpretation of the following optional directives in rsed.envvars has changed (since version 8.0.3):
  - (\_RSE\_JAVAOPTS) -Dprocess.cleanup.interval
- RSE - The following optional configuration files are new (since version 8.5):
  - include.conf
- 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 and 8.0.3.1):
  - process\_audit.rex
  - process\_logon.sh
- Security - support for new security profiles has been added (since version 8.0.3):
  - FEK.PTC.\*\*
- zUnit - New optional PROCLIB member has been added (since version 8.5):
  - AZUZUNIT
- New publication, *Rational Developer for System z Messages and Codes* (SC14-7497).

- New publication, *Rational Developer for System z Answers to common host configuration and maintenance issues* (SC14-7373).

## Configurable files

Table 25 shows an overview of files that are customized in version 8.5. The Developer for System z sample libraries, FEK.SFEKSAMP, FEK.SFEKSAMV and /usr/lpp/rdz/samples/, contain more customizable members than listed here, such as sample CARMA source code and jobs to compile them.

The following members and files are no longer customizable or no longer used.

- FMIEXT.properties is no longer used

**Note:** Sample job FEKSETUP copies all listed members to different data sets and directories, default FEK.#CUST.\* and /etc/rdz/\*.

Table 25. Version 8.5 customizations

Member/File	Default location	Purpose	Migration notes
FEKSETUP	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to create data sets and directories, and populate them with customizable files	Updated to include new customizable members, create new directory structure, and remove actions for files that are no longer used
JMON	FEK.SFEKSAMP(FEJJJCL) [FEK.#CUST.PROCLIB]	JCL for JES Job Monitor	None
FEJJJCL	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(JMON)]	Name for JMON member	See JMON member
RSED	FEK.SFEKSAMP(FEKRSERD) [FEK.#CUST.PROCLIB]	JCL for RSE daemon	Changed TMPDIR support
FEKRSERD	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(RSED)]	Name for RSED member	See RSED member
LOCKD	FEK.SFEKSAMP(FEKLOCKD) [FEK.#CUST.PROCLIB]	JCL for lock daemon	Changed TMPDIR support
FEKLOCKD	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(LOCKD)]	Name for LOCKD member	See LOCKD member
ELAXF*	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	JCL for remote project builds, and so on	Member ELAXFUOP has changed
FEKRACF	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL for security definitions	None
FEJJCNFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	JES Job Monitor configuration file	New optional directives have been added

Table 25. Version 8.5 customizations (continued)

Member/File	Default location	Purpose	Migration notes
FEJTSO	FEK.SFEKSAMP [FEK.#CUST.CNTL]	JCL for TSO submits	None
CRA\$VMSG	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to create the CARMA message VSAM	VSAM input has changed
CRA\$VDEF	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to create the CARMA configuration VSAM	None
CRA\$VSTR	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to create the CARMA custom information VSAM	None
CRA\$VCAD	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to create the CARMA configuration VSAM for CA Endeavor® SCM RAM	VSAM input has changed
CRA\$VCAS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to create the CARMA custom information VSAM for CA Endeavor® SCM RAM	VSAM input has changed
CRASUBMT	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CARMA batch startup CLIST	None
CRASUBCA	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CARMA batch startup CLIST for CA Endeavor® SCM RAM	Additional DD statements added
CRABCFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CARMA batch actions configuration for CA Endeavor® SCM RAM	NEW, customization is optional
CRABATCA	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CARMA batch action JCL for CA Endeavor® SCM RAM	NEW, customization is optional
CRASHOW	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CARMA configuration for CA Endeavor® SCM RAM	New filters are added
CRATMAP	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CARMA configuration for CA Endeavor® SCM RAM	None
CRANDVRA	FEK.SFEKPROC	CARMA allocation REXX for CA Endeavor® SCM RAM	Additional DD statements added
CRA#VSLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to create the SCLM RAM's message VSAM	None
CRA#ASLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to create the SCLM RAM's data sets	None
CRA#VPDS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to create the PDS RAM's message VSAM	None
CRA#UADD	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to merge RAM definitions	None

Table 25. Version 8.5 customizations (continued)

Member/File	Default location	Purpose	Migration notes
CRA#UQRY	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to extract RAM definitions	None
CRAXJCL	FEK.SFEKSAMP [FEK.#CUST.ASM]	Sample source code for IRXJCL replacement	None
CRA#CIRX	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to compile CRAXJCL	None
ADNCSDRS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to define the RESTful CRD server to primary CICS region	None
ADNCSDTX	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to define alternate transaction IDs to CICS region	None
ADNTXNC	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to create alternate transaction IDs	None
ADNMSGHC	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to compile ADNMSGHS	None
ADNMSGHS	FEK.SFEKSAMP [FEK.#CUST.COBOLE]	Sample source code for the Pipeline Message Handler	None
ADNVCRD	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to create the CRD repository	None
ADNCSDWS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to define the Web Service CRD server to primary CICS region	None
ADNCSDAR	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to define the CRD server to non-primary CICS regions	None
ADNJSPAU	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to update the CRD defaults	None
ADNVMFST	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to create and define the Manifest repository	None
ELAXMSAM	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	JCL procedure of the WLM address space	None
ELAXMJCL	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to define the PL/I and COBOL Stored Procedure Builder to DB2	None
AZUZUNIT	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	JCL procedure for zUnit	NEW, customization is optional
FEKRNPLI	FEK.SFEKSAMP [FEK.#CUST.CNTL]	Rexx to call the PL/I compiler from within the preprocessor framework	NEW, customization is optional

Table 25. Version 8.5 customizations (continued)

Member/File	Default location	Purpose	Migration notes
FEKLOGS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL to collect log files	Added additional checks. Any customization to older files must be done again.
rzed.envvars	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE environment variables	Older copies must be replaced by this one and the customizations done again.
ISPF.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	TSO/ISPF Client Gateway configuration file	None
CRASRV.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	CARMA configuration file	Added support for ephemeral ports
crastart.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	CARMA configuration file for CRASTART usage	None
crastart.endevor.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	CARMA configuration file for CRASTART usage for CA Endeavor® SCM RAM	Additional DD statements added
include.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	Forced includes for C/C++ content assist	NEW, customization is optional
ssl.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE SSL configuration file	None
rsecomm.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE trace configuration file	Some directives became optional
pushtoclient.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	Push information to the client configuration file	Additional directives added and existing directives enhanced

---

## Chapter 11. Operator commands

This chapter provides an overview of the available operator (or console) commands for Developer for System z.

---

### Start (S)

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

#### Integrated debugger

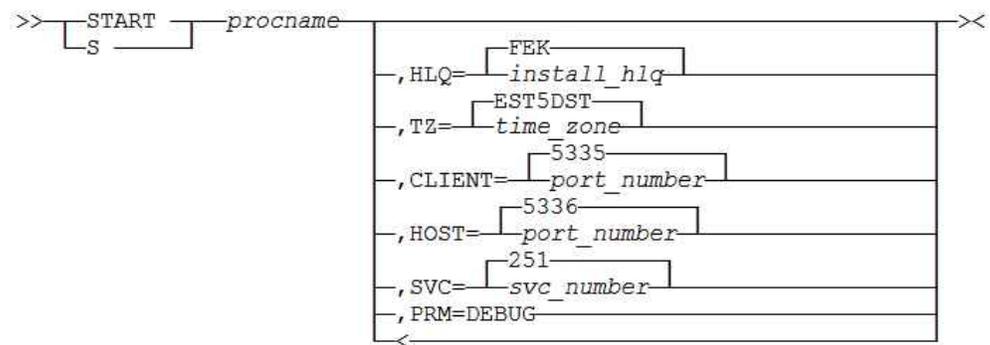


Figure 37. START DBGMGR operator command

#### procname

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

#### HLQ=install\_hlq

High-level qualifier used to install Developer for System z. The default is FEK.

#### TZ=time\_zone

Time zone offset. The default is EST5DST.

#### CLIENT=port\_number

The port used for external (client-host) communication, default 5335.

#### HOST=port\_number

The port used for internal (host-confined) communication, default 5336.

#### SVC=svc\_number

The SVC number used for debugging read-only CICS transactions, default 251.

#### PRM=DEBUG

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

## JES Job Monitor

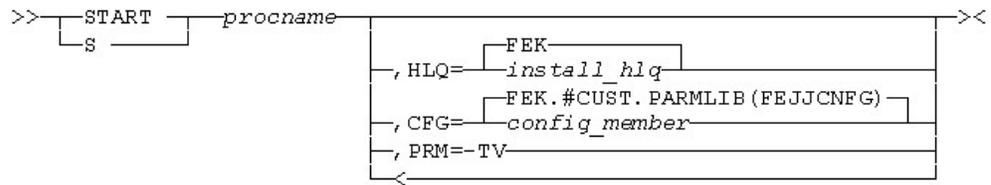


Figure 38. START JMON operator command

### procname

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

### HLQ=install\_hlq

High-level qualifier used to install Developer for System z. The default is FEK.

### CFG=config\_member

Absolute data set and member name of the JES Job Monitor configuration file. The default is FEK.#CUST.PARMLIB(FEJJCNFG). If this variable is set to NULLFILE, JES Job Monitor will use default configuration values.

### PRM=-TV

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

## RSE daemon

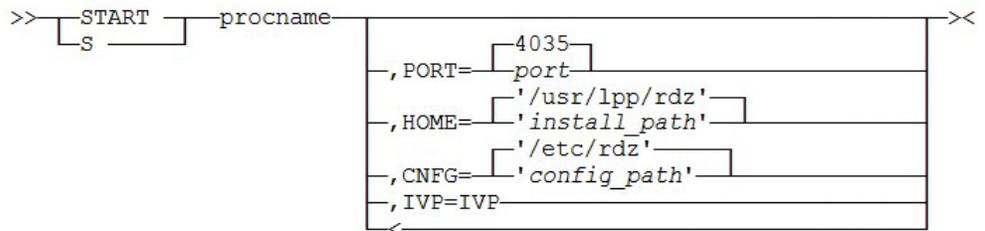


Figure 39. START RSED operator command

### procname

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

### PORT=port

The port used by the RSE daemon for the clients to connect. If not specified, the port defined in /etc/rdz/rsed.envvars in the variable `_RSE_RSED_PORT` is used. The default is 4035.

### IVP=IVP

Do not start the server but run the RSE daemon installation verification program (IVP).

**CNFG='config\_path'**

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

**HOME='install\_path'**

Path prefix and the mandatory /usr/lpp/rdz used to install Developer for System z. The default is '/usr/lpp/rdz'. The z/OS UNIX path is case-sensitive and must be enclosed in single quotation marks (') to preserve lowercase characters.

---

## Modify (F)

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

### Integrated debugger

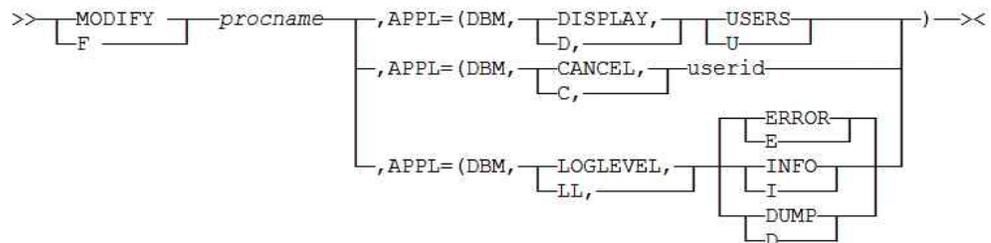


Figure 40. MODIFY DBGMGR operator command

**procname**

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

**DISPLAY,USERS**

Display the active users with AQECM104I console messages. Message AQECM103I is issued if there are no active users. The user list shows the state of that user in the server. Refer to the "Integrated Debugger" section of the "Understanding Developer for System z" chapter in the *Host Configuration Reference* (SC14-7290) for an overview of the Integrated Debugger data flow.

```

AQECM104I User:IBMUSER  RegisterSocket(2) waits for probe
connection
AQECM104I User:IBMUSR2  ProbeSocket(3) waits for register
connection
AQECM104I User:IBMUSR3  EngineSocket(4) connected to
ProbeSocket(8)
AQECM104I User:IBMUSR4  ProbeSocket(5) waits for engine
connection
AQECM103I There is no active user

```

**CANCEL,userid**

Cancel all debug sessions for the specified user ID. Results are shown with a AQECM110I or AQECM111I console message.

```

AQECM110I user(IBMUSER) canceled
AQECM111I user(IBMUSER) not connected

```

## LOGLEVEL,{ERROR | INFO | DUMP}

Control the detail level of the Debug Manager message log (DD SYSPRINT). The default is E (Error). A message "LOGLEVEL command processed normally" is written to the console with message ID AQECM101I.

E or ERROR	Error messages only (default)
I or INFO	Error and Informational messages
D or DUMP	Error, Informational, and Debug/dump messages

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

## JES Job Monitor

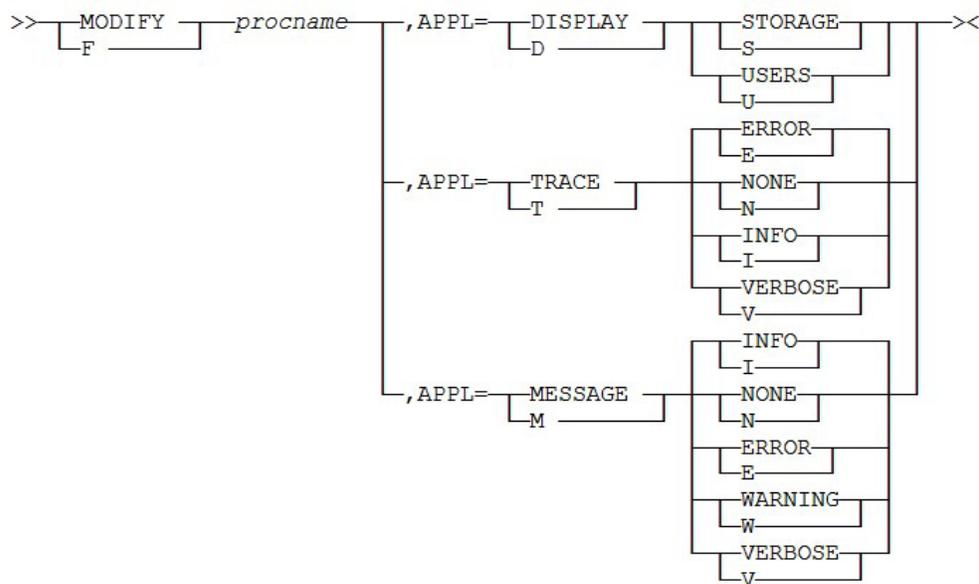


Figure 41. MODIFY JMON operator command

### procname

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

### DISPLAY STORAGE

Write a storage usage report to DD SYSOUT. A message "JMON storage information written to SYSOUT" is written to the console with message ID BPXM023I. The storage usage report shows various storage-related fields with sizes in bytes, kilobytes, and megabytes.

```
>>>STORAGE TRACE (console request)<<<
LDAREGRQ  0000000000 00000000K 00000M requested region size
          below 16M line
LDASIZA   00006266880 00006120K 00005M maximum region size
LDALIMIT  00006266880 00006120K 00005M limit
LDAVVRG   00006266880 00006120K 00005M getmain limit
LDALOAL   00000061440 00000060K 00000M in use
LDAHIAL   00000266240 00000260K 00000M LSQA/SWA/private subpools
_GAP      00000000000 00000000K 00000M gaps in allocation
_AVAIL    00005939200 00005800K 00005M available (including gaps)
_MAX      00006000640 00005860K 00005M current limit
```

```

above 16M line
LDAESIZA 01905262592 01860608K 01817M maximum region size
LDAELIM 01905262592 01860608K 01817M limit
LDAEVVRG 01905262592 01860608K 01817M getmain limit
LDAELOAL 00000937984 00000916K 00000M in use
LDAEHIAL 00012754944 00012456K 00012M ELSQA/ESWA/private subpools
_EGAP 00000000000 00000000K 00000M gaps in allocation
_EAVAIL 01891569664 01847236K 01803M available (including gaps)
_EMAX 01892507648 01848152K 01804M current limit

```

### DISPLAY USERS

Write a list of active users to DD SYSOUT. A message "JMON user list written to SYSOUT" is written to the console with message ID BPXM023I. The user list shows various user-related data, including CPU usage.

```

S0 userid USER 4:04(elapsed) 4:04(idle)
Users: 1

```

### TRACE {NONE | ERROR | INFO | VERBOSE}

Control the detail level of the JES Job Monitor trace log (DD SYSOUT). The default is E (Error). A message "JMON TRACE LEVEL:{NONE | ERROR | INFO | VERBOSE}" is written to the console with message ID BPXM023I.

N or NONE	Startup messages only
E or ERROR	Startup and Error messages only (default)
I or INFO	Startup, Error, and Informational messages
V or VERBOSE	Startup, Error, Informational, and Verbose messages

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

### MESSAGE {NONE | ERROR | WARNING | INFO | VERBOSE}

Control the detail level of the JES Job Monitor message log (DD SYSPRINT). The default is I (Informational). A message "JMON MESSAGE LEVEL:{NONE | ERROR | WARNING | INFO | VERBOSE}" is written to the console with message ID BPXM023I.

N or NONE	No messages.
E or ERROR	Error messages only
W or WARNING	Error and Warning messages
I or INFO	Error, Warning, and Informational messages (default)
V or VERBOSE	Error Warning, Informational, and Verbose messages

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

# RSE daemon

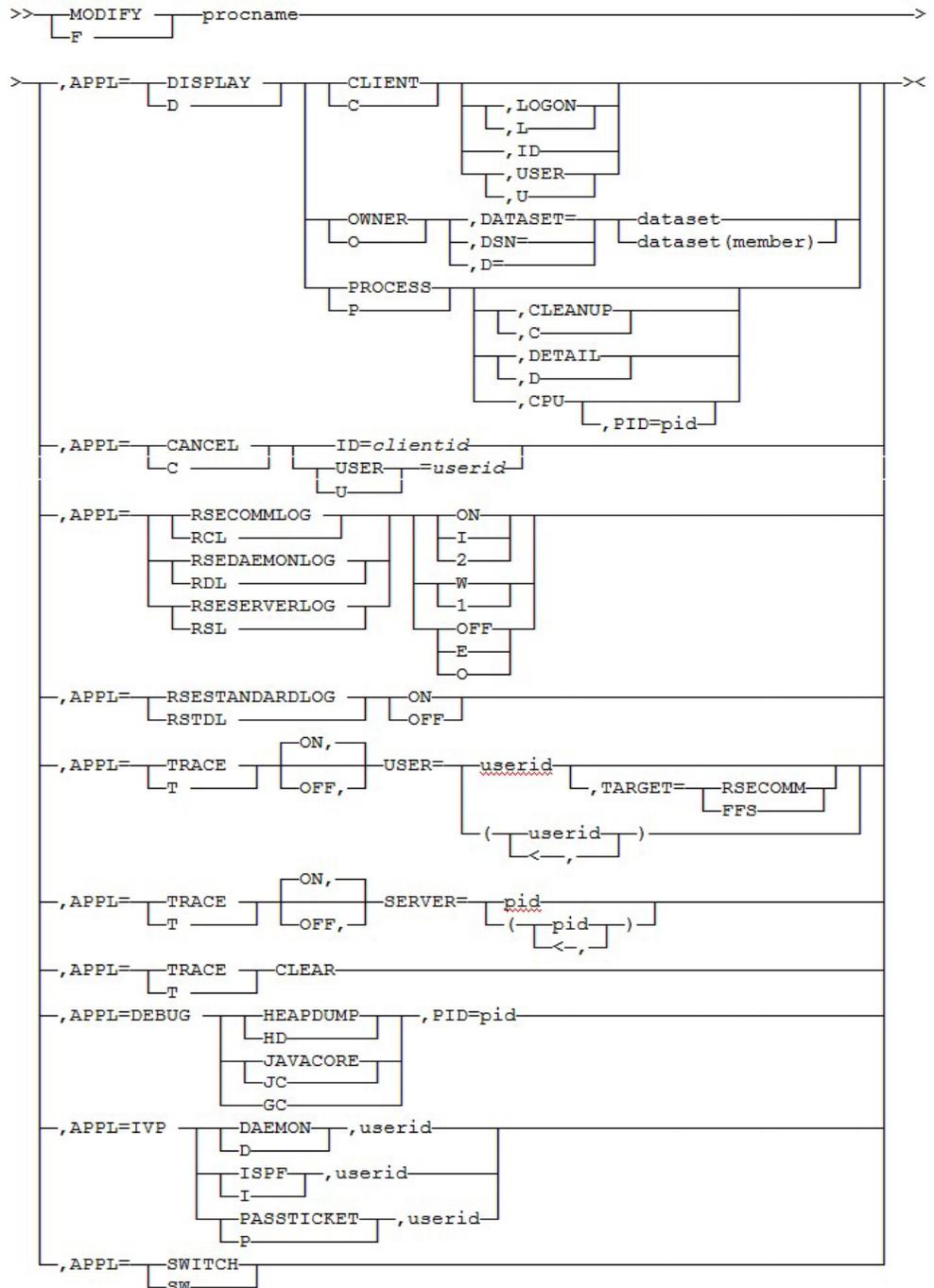


Figure 42. MODIFY RSED operator command

## procname

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

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

Display the active clients in a single BPXM023I message. The result layout

depends on the command option that 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 OWNER,DATASET={dataset | dataset(member)}**

Display the data set enqueue owner in a single BPXM023I message.

```
FEK217I <dataset[(member)]> is locked by <userid>
FEK218I <dataset[(member)]> is not locked
FEK219E Failed to determine lock owner for <dataset[(member)]>
```

- The server also reports the locks held by other products, such as ISPF.
- The **D GRS,RES=(\*,dataset)** operator command is unable to tell which Developer for System z user is the actual enqueue holder, all it can tell you is the threadpool in which the user is active.

#### **DISPLAY PROCESS[{,CLEANUP | ,CPU [,PID=pid] | ,DETAIL}]**

Display the RSE thread pool processes in one or more BPXM023I messages. 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`. The reported Java heap usage includes storage that is released by Developer for System z, but which is not yet freed by the Java garbage collection process.
- <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 26 on page 184 documents the possible non-blank values for <error status>.

Table 26. 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. To remove this entry from the table, use the CLEANUP option of the <b>DISPLAY PROCESS</b> modify command.
*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. To remove this entry from the table, use the CLEANUP option of the <b>DISPLAY PROCESS</b> modify command.
*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      30
  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. Due to other limiting factors, the defined limit might never be reached.

The CPU option of the **DISPLAY PROCESS** modify command shows the accumulated CPU usage, in milliseconds, of each thread in a thread pool. Every thread pool has a BPXM023I message. By default, all thread pools report the CPU usage, but you can limit the scope to a single thread pool by specifying PID=pid on the operator command, where pid is the process ID of the target thread pool.

```
ProcessId(421 ) ASId(007D) JobName(RSED8) Order(1)
USERID  THREAD-ID      TCB@  ACC_TIME TAG
STCRSE  0EDE5400000000 005E6B60 822 1/ThreadPoolProcess
STCRSE  0EDE8700000000 005E69C8 001
STCRSE  0EDE9800000000 005E6518 1814
STCRSE  0EDEBA00000000 005E66B0 2305
STCRSE  0EDECB00000000 005E62F8 001
STCRSE  0EDED000000000 005E60D8 001
STCRSE  0EDF8600000000 005C2BF8 628 6/ThreadPoolMonitor$Memory
UsageMonitor
STCRSE  0EDF9700000000 005C2D90 003 7/ThreadPoolMonitor
STCRSE  0EDFDB00000000 005C29D8 001
STCRSE  0EE22E00000000 005C1BE0 070
IBMUSER 0EE0EB00000000 005C22B8 276 20/ServerReceiver
IBMUSER 0EE25000000000 005C19C0 137 16/ServerUpdateHandler
```

```

IBMUSER 0EE2610000000013 005C17A0      509 15/ServerCommandHandler
IBMUSER 0EE1840000000014 005C1E00      065 21/ZosSystemMiner
STCRSE  0EE1510000000016 005C2098      078
STCRSE  0EE1950000000017 005C1580      001
IBMUSER 0EE23F0000000018 005C1360      021 26/UniversalFileSystemMine
r
IBMUSER 0EE2A5000000001C 005C0CF0      003 27/EnvironmentMiner
IBMUSER 0EE283000000001D 005C1140      002 31/CommandMiner
IBMUSER 0EE272000000001E 005C0E88      081 32/MVSFileSystemMiner
IBMUSER 0EE294000000001F 005C0AD0      002 33/MVSByteStreamHandler$Op
enCloseThread
STCRSE  0EE2E90000000023 005C0470      001
IBMUSER 0EE2C70000000024 005C08B0      050 38/JESMiner
IBMUSER 0EE2B60000000026 005C0690      004 40/FAMiner
IBMUSER 0EE30B0000000027 005C0250      002 41/LuceneMiner
IBMUSER 0EE31C0000000028 005C0030      002 42/CDTParserMiner
IBMUSER 0EE32D0000000029 005BDE00      002 43/MVSLuceneMiner
IBMUSER 0EE33E000000002A 005BDBE0      002 44/CDTMVSParserMiner

```

If the output size exceeds the maximum number of lines for a console message, the output is split over multiple BPXM023I messages. These additional messages have the same header as the first message, but with the CONTINUATION keyword added to the first line.

```

ProcessId(421      ) ASId(007D) JobName(RSED8) Order(1) CONTINUATION
USERID  THREAD-ID      TCBE      ACC_TIME TAG

```

The output is limited to the first 4000 threads for each thread pool.

#### CANCEL ID=clientid

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

When a client connection is cancelled, the host system threads go through normal termination processing to clean up resources used by them. This action implies that some threads can take a few minutes before they end; for example, because they are waiting on the keep-alive mechanism to time out.

#### CANCEL USER=userid

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

When a client connection is cancelled, the host system threads go through normal termination processing to clean up resources used by them. This action implies that some threads can take a few minutes before they end; for example, because they are waiting on the keep-alive mechanism to time out.

#### 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. Three detail levels are 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 | W | E | 2 | 1 | 0}**

Control the trace detail level for RSE daemon (rsedaemon.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.

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

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

**RSESTANDARDLOG {ON | OFF}**

Disable (OFF) or enable (ON) the updating of the log files that hold the stdout and stderr streams of the stdout\*.log and stderr\*.log thread pools. 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.

**TRACE [{ON, | OFF,}]USER=userid[,TARGET={FFS | RSECOMM}]**

Enable (ON) or disable (OFF) tracing for the specified user IDs. The default is ON. This setting overrides the default setting controlled by the **MODIFY RSECOMMLOG** operator command. Two detail levels are available:

OFF	Error messages only
ON (default)	Error, warning, and informational messages.

The command alters the trace detail level for RSE server (rsecomm.log) and the MVS data set services (lock.log and ffs\*.log). This can be limited with the TARGET keyword, which accepts two values:

FFS	Set the specified log level only for MVS data set services (lock.log and ffs*.log)
RSECOMM	Set the specified log level only for RSE server (rsecomm.log)

The command can be issued for users that are currently not logged on. The setting remains active when a user logs off and will be used again when the user logs on.

Use the `USER` directive in `rsecomm.properties` to simulate issuing the **MODIFY TRACE USER** command at server startup. Existing settings from previous **MODIFY TRACE USER** or **MODIFY TRACE SERVER** operator commands or the `USER` directive in `rsecomm.properties` will be replaced by the setting of this command.

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

**TRACE [{ON, | OFF,}]USER=(userid,userid,...)**

Enable (ON) or disable (OFF) tracing for the specified user IDs. The default is ON. This setting overrides the default setting controlled by the **MODIFY RSECOMMLOG** operator command. Two detail levels are available:

OFF	Error messages only.
ON (default)	Error, warning, and informational messages.

The command alters the trace detail level for RSE server (`rsecomm.log`) and the MVS data set services (`lock.log` and `ffs*.log`). The command can be issued for users that are currently not logged on. The setting remains active when a user logs off and will be used again when the user logs on. Use the `USER` directive in `rsecomm.properties` to simulate issuing the **MODIFY TRACE USER** command at server startup. Existing settings from previous **MODIFY TRACE USER** or **MODIFY TRACE SERVER** operator commands or the `USER` directive in `rsecomm.properties` will be replaced by the setting of this command.

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

**TRACE [{ON, | OFF,}] SERVER={pid | (pid,pid,...)}**

Enable (ON) or disable (OFF) tracing for all users in the specified thread pool where `pid` is the process ID of an RSE thread pool. The default is ON. This setting overrides the default setting controlled by the **MODIFY RSECOMMLOG** operator command. Two detail levels are available:

OFF	Error messages only.
ON (default)	Error, warning, and informational messages.

The command alters the trace detail level for RSE server (`rsecomm.log`) and the MVS data set services (`lock.log` and `ffs*.log`). Existing settings from previous **MODIFY TRACE USER** or **MODIFY TRACE SERVER** operator commands or the `USER` directive in `rsecomm.properties` will be replaced by the setting of this command.

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

**TRACE CLEAR**

Remove all trace overrides set by the **MODIFY TRACE USER** and **MODIFY TRACE SERVER** operator commands and the `USER` directive in `rsecomm.properties`.

**DEBUG HEAPDUMP,PID=pid**

Request a Java Heap dump for a specified thread pool, where `pid` is the process ID of an RSE thread pool. The dump is written to the directory specified by `_CEE_DUMPTARG` in `rse.envvars`, where the default value is `/tmp`. Results are shown in a single BPXM023I console message.

JVMDUMP034I User requested Heap dump using '/tmp/heapdump.20120223.211'  
430.16777590.0001.phd' through JVMRI

#### **DEBUG JAVACORE,PID=pid**

Request a Java Core dump for a specified thread pool, where pid is the process ID of an RSE thread pool. The dump is written to the directory specified by \_CEE\_DUMPTARG in rsed.envvars, where the default value is /tmp. Results are shown in a single BPXM023I console message.

JVMDUMP034I User requested Java dump using '/tmp/javacore.20120223.214'  
244.16777590.0002.phd' through JVMRI

#### **DEBUG GC,PID=pid**

Request a Java Garbage Collection for a specified thread pool, where pid is the process ID of an RSE thread pool.

#### **IVP DAEMON,userid**

Log user ID userid on to RSE daemon to do a connection test. Results are shown with one or 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 generates a PassTicket which is used as password for the IVP, so there is no Write To Operator with Reply (WTOR) requesting a password.

#### **IVP ISPF,userid**

Invoke ISPF's Client Gateway as user ID userid. Results are shown with one or 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.SISPLOAD
+FEK900I ISPF IVP: isplib=ISP.SISPMENU
+FEK900I ISPF IVP: isptlib=ISP.SISPTENU
+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 TSO/ISPF session initializati
on check only
+FEK900I ISPF IVP: *** CHECK : ENVIRONMENT VARIABLES - key variables
displayed below :
+FEK900I ISPF IVP: Server PATH           = ./usr/lpp/java/J6.0/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: CGI_ISPHOME          = /usr/lpp/ispf
+FEK900I ISPF IVP: CGI_ISPCONF         = /etc/rdz
+FEK900I ISPF IVP: CGI_ISPWORK        = /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 generates a PassTicket which is used as password for the IVP, so there is no Write To Operator with Reply (WTOR) 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 the RSE daemon with the IVP=IVP argument calls a PassTicket IVP that tests PassTicket generation, but it cannot test PassTicket reusability.

- RSE daemon generates a PassTicket which is used as password for the IVP, so there is no Write To Operator with Reply (WTOR) requesting a password.

## SWITCH

Switch to a new audit log file.

---

## Stop (P)

Use the **STOP** command to stop an active task. The abbreviated version of the command is the letter P.



Figure 43. STOP operator command

### procname

The name of the member in a procedure library that was used to start the server. The default names used during the host system configuration are DBGMR, JMON, and RSED for Integrated Debugger, JES Job Monitor, and the RSE daemon, respectively.

---

## How to read a syntax diagram

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

## Symbols

The following symbols are used in syntax diagrams:

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

## Operands

The following types of operands are used in syntax diagrams:

- Required operands are displayed on the main path line:  
 >>—REQUIRED\_OPERAND—<<
- Optional operands are displayed below the main path line:  
 >>—[OPTIONAL\_OPERAND]—<<
- Default operands are displayed above the main path line:  
 >>—[DEFAULT\_OPERAND]—<<

Operands are classified as keywords or variables:

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

## Syntax example

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

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

## Nonalphanumeric characters and blank spaces

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

```
>>—OPERAND—=(—001— 0.001—)————><
```

## Selecting more than one operand

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

```
>>—┐—————><
    └—REPEATABLE_OPERAND_1—┘
    └—REPEATABLE_OPERAND_2—┘
    <—┘
```

## Longer than one line

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

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

## Syntax fragments

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

```
>>—| Syntax fragment |—————><
```

Syntax fragment:

```
|—1ST_OPERAND—, —2ND_OPERAND—, —3RD_OPERAND—|
```



---

## Chapter 12. Host Configuration Reference

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

---

### Understanding Developer for System z

The Developer for System z host system consists of several components that interact to give the client access to the host system 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 system 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 system 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 configuration makes RSE a prime target for tuning the Developer for System z setup. However, maintaining hundreds of users, each using 17 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.

---

## User exit considerations

This chapter assists you with enhancing Developer for System z by writing exit routines.

---

## 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 when you want multiple instances of Developer for System z to be 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 might 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 might 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 might encounter when setting up TCP/IP, or during checking or modifying an existing setup.



---

## Bibliography

---

### Referenced publications

The following publications are referenced in this document:

Table 27. Referenced publications

Publication title	Order number	Reference	Reference Web site
Program Directory for IBM Rational Developer for System z	GI11-8298	Developer for System z	<a href="http://www-01.ibm.com/support/docview.wss?uid=swg27038517">http://www-01.ibm.com/support/docview.wss?uid=swg27038517</a>
Program Directory for IBM Rational Developer for System z Host Utilities	GI13-2864	Developer for System z	<a href="http://www-01.ibm.com/support/docview.wss?uid=swg27038517">http://www-01.ibm.com/support/docview.wss?uid=swg27038517</a>
Rational Developer for System z Prerequisites	SC23-7659	Developer for System z	<a href="http://www-01.ibm.com/support/docview.wss?uid=swg27038517">http://www-01.ibm.com/support/docview.wss?uid=swg27038517</a>
Rational Developer for System z Host Configuration Quick Start	GI11-9201	Developer for System z	<a href="http://www-01.ibm.com/support/docview.wss?uid=swg27038517">http://www-01.ibm.com/support/docview.wss?uid=swg27038517</a>
Rational Developer for System z Host Configuration Guide	SC23-7658	Developer for System z	<a href="http://www-01.ibm.com/support/docview.wss?uid=swg27038517">http://www-01.ibm.com/support/docview.wss?uid=swg27038517</a>
Rational Developer for System z Host Configuration Reference	SC14-7290	Developer for System z	<a href="http://www-01.ibm.com/support/docview.wss?uid=swg27038517">http://www-01.ibm.com/support/docview.wss?uid=swg27038517</a>
Rational Developer for System z Host Configuration Utility Guide	SC14-7282	Developer for System z	<a href="http://www-01.ibm.com/support/docview.wss?uid=swg27038517">http://www-01.ibm.com/support/docview.wss?uid=swg27038517</a>
Rational Developer for System z Messages and Codes	SC14-7497	Developer for System z	<a href="http://www-01.ibm.com/support/docview.wss?uid=swg27038517">http://www-01.ibm.com/support/docview.wss?uid=swg27038517</a>
Rational Developer for System z Answers to common host configuration and maintenance issues	SC14-7373	Developer for System z	<a href="http://www-01.ibm.com/support/docview.wss?uid=swg27038517">http://www-01.ibm.com/support/docview.wss?uid=swg27038517</a>
Rational Developer for System z Common Access Repository Manager Developer's Guide	SC23-7660	Developer for System z	<a href="http://www-01.ibm.com/support/docview.wss?uid=swg27038517">http://www-01.ibm.com/support/docview.wss?uid=swg27038517</a>
Rational Developer for System z Prerequisites	SC23-7659	Developer for System z	<a href="http://www.ibm.com/software/rational/products/developer/systemz/library/index.html">http://www.ibm.com/software/rational/products/developer/systemz/library/index.html</a>
Rational Developer for System z Host Configuration Quick Start	GI11-9201	Developer for System z	<a href="http://www.ibm.com/software/rational/products/developer/systemz/library/index.html">http://www.ibm.com/software/rational/products/developer/systemz/library/index.html</a>
SCLM Developer Toolkit Administrator's Guide	SC23-9801	Developer for System z	<a href="http://www-01.ibm.com/support/docview.wss?uid=swg27038517">http://www-01.ibm.com/support/docview.wss?uid=swg27038517</a>
Using APPC to provide TSO command services	SC14-7291	White paper	<a href="http://www-01.ibm.com/support/docview.wss?uid=swg27038517">http://www-01.ibm.com/support/docview.wss?uid=swg27038517</a>

Table 27. Referenced publications (continued)

Publication title	Order number	Reference	Reference Web site
Using ISPF Client Gateway to provide CARMA services	SC14-7292	White paper	<a href="http://www-01.ibm.com/support/docview.wss?uid=swg27038517">http://www-01.ibm.com/support/docview.wss?uid=swg27038517</a>
Communications Server IP Configuration Guide	SC31-8775	z/OS 1.13	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
Communications Server IP Configuration Reference	SC31-8776	z/OS 1.13	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
Communications Server IP Diagnosis Guide	GC31-8782	z/OS 1.13	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
Communications Server IP System Administrator's Commands	SC31-8781	z/OS 1.13	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
Communications Server SNA Network Implementation Guide	SC31-8777	z/OS 1.13	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
Communications Server SNA Operations	SC31-8779	z/OS 1.13	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
Cryptographic Services System SSL Programming	SC24-5901	z/OS 1.13	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
DFSMS Macro Instructions for Data Sets	SC26-7408	z/OS 1.13	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
DFSMS Using data sets	SC26-7410	z/OS 1.13	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
Language Environment Customization	SA22-7564	z/OS 1.13	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
Language Environment Debugging Guide	GA22-7560	z/OS 1.13	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
MVS Diagnosis: Tools and Service Aids	GA22-7589	z/OS 1.13	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
MVS Initialization and Tuning Guide	SA22-7591	z/OS 1.13	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
MVS Initialization and Tuning Reference	SA22-7592	z/OS 1.13	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
MVS JCL Reference	SA22-7597	z/OS 1.13	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
MVS Planning APPC/MVS Management	SA22-7599	z/OS 1.13	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
MVS Planning Workload Management	SA22-7602	z/OS 1.13	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
MVS System Commands	SA22-7627	z/OS 1.13	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
Security Server RACF Command Language Reference	SA22-7687	z/OS 1.13	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
Security Server RACF Security Administrator's Guide	SA22-7683	z/OS 1.13	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>

Table 27. Referenced publications (continued)

Publication title	Order number	Reference	Reference Web site
TSO/E Customization	SA22-7783	z/OS 1.13	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
TSO/E REXX Reference	SA22-7790	z/OS 1.13	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
UNIX System Services Command Reference	SA22-7802	z/OS 1.13	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
UNIX System Services Planning	GA22-7800	z/OS 1.13	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
UNIX System Services User's Guide	SA22-7801	z/OS 1.13	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
Using REXX and z/OS UNIX System Services	SA22-7806	z/OS 1.13	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
Java™ Diagnostic Guide	SC34-6650	Java 6.0	<a href="http://www.ibm.com/developerworks/java/jdk/diagnosis/">http://www.ibm.com/developerworks/java/jdk/diagnosis/</a>
Java SDK and Runtime Environment User Guide	/	Java 6.0	<a href="http://www-03.ibm.com/servers/eserver/zseries/software/java/">http://www-03.ibm.com/servers/eserver/zseries/software/java/</a>
Resource Definition Guide	SC34-6430	CICSTS 3.1	<a href="http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html">http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html</a>
Resource Definition Guide	SC34-6815	CICSTS 3.2	<a href="http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html">http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html</a>
Resource Definition Guide	SC34-7000	CICSTS 4.1	<a href="https://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp?topic=/com.ibm.cics.ts.home.doc/library/library_html.html">https://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp?topic=/com.ibm.cics.ts.home.doc/library/library_html.html</a>
Resource Definition Guide	SC34-7181	CICSTS 4.2	<a href="https://publib.boulder.ibm.com/infocenter/cicsts/v4r2/index.jsp?topic=/com.ibm.cics.ts.home.doc/library/library_html.html">https://publib.boulder.ibm.com/infocenter/cicsts/v4r2/index.jsp?topic=/com.ibm.cics.ts.home.doc/library/library_html.html</a>
RACF Security Guide	SC34-6454	CICSTS 3.1	<a href="http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html">http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html</a>
RACF Security Guide	SC34-6835	CICSTS 3.2	<a href="http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html">http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html</a>
RACF Security Guide	SC34-7003	CICSTS 4.1	<a href="https://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp?topic=/com.ibm.cics.ts.home.doc/library/library_html.html">https://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp?topic=/com.ibm.cics.ts.home.doc/library/library_html.html</a>
RACF Security Guide	SC34-7179	CICSTS 4.2	<a href="https://publib.boulder.ibm.com/infocenter/cicsts/v4r2/index.jsp?topic=/com.ibm.cics.ts.home.doc/library/library_html.html">https://publib.boulder.ibm.com/infocenter/cicsts/v4r2/index.jsp?topic=/com.ibm.cics.ts.home.doc/library/library_html.html</a>
Language Reference	SC27-1408	Enterprise COBOL for z/OS	<a href="http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html">http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html</a>

The following Web sites are referenced in this document:

Table 28. Referenced Web sites

Description	Reference Web site
Developer for System z Information Center	<a href="http://pic.dhe.ibm.com/infocenter/ratdevz/v9r0/index.jsp">http://pic.dhe.ibm.com/infocenter/ratdevz/v9r0/index.jsp</a>
Developer for System z Library	<a href="http://www-01.ibm.com/support/docview.wss?uid=swg27038517">http://www-01.ibm.com/support/docview.wss?uid=swg27038517</a>

Table 28. Referenced Web sites (continued)

Description	Reference Web site
Developer for System z home page	<a href="http://www-03.ibm.com/software/products/us/en/developperforsystemz/">http://www-03.ibm.com/software/products/us/en/developperforsystemz/</a>
Developer for System z Recommended service	<a href="http://www-01.ibm.com/support/docview.wss?rs=2294&amp;context=SS2QJ2&amp;uid=swg27006335">http://www-01.ibm.com/support/docview.wss?rs=2294&amp;context=SS2QJ2&amp;uid=swg27006335</a>
Developer for System z enhancement request	<a href="https://www.ibm.com/developerworks/support/rational/rfe/">https://www.ibm.com/developerworks/support/rational/rfe/</a>
z/OS internet library	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
CICSTS Information Center	<a href="https://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp">https://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp</a>
IBM Tivoli® Directory Server	<a href="http://www-01.ibm.com/software/tivoli/products/directory-server/">http://www-01.ibm.com/software/tivoli/products/directory-server/</a>
Problem Determination Tools Plug-ins	<a href="http://www-01.ibm.com/software/awdtools/deployment/pdplugins/">http://www-01.ibm.com/software/awdtools/deployment/pdplugins/</a>
Download Apache Ant	<a href="http://ant.apache.org/">http://ant.apache.org/</a>
Java keytool documentation	<a href="http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html">http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html</a>
CA support home page	<a href="https://support.ca.com/">https://support.ca.com/</a>

## Informational publications

The following publications can be helpful in understanding setup issues for the requisite host system components:

Table 29. Informational publications

Publication title	Order number	Reference	Reference website
ABCs of z/OS System Programming Volume 9 (z/OS UNIX)	SG24-6989	Redbook	<a href="http://www.redbooks.ibm.com/">http://www.redbooks.ibm.com/</a>
System Programmer's Guide to: Workload Manager	SG24-6472	Redbook	<a href="http://www.redbooks.ibm.com/">http://www.redbooks.ibm.com/</a>
TCPIP Implementation Volume 1: Base Functions, Connectivity, and Routing	SG24-7532	Redbook	<a href="http://www.redbooks.ibm.com/">http://www.redbooks.ibm.com/</a>
TCPIP Implementation Volume 3: High Availability, Scalability, and Performance	SG24-7534	Redbook	<a href="http://www.redbooks.ibm.com/">http://www.redbooks.ibm.com/</a>
TCP/IP Implementation Volume 4: Security and Policy-Based Networking	SG24-7535	Redbook	<a href="http://www.redbooks.ibm.com/">http://www.redbooks.ibm.com/</a>
Tivoli Directory Server for z/OS	SG24-7849	Redbook	<a href="http://www.redbooks.ibm.com/">http://www.redbooks.ibm.com/</a>

---

## Documentation notices for IBM Rational Developer for System Z

© Copyright IBM Corporation 2009, 2013.

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

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

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

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

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

For license inquiries regarding double-byte (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.  
1623-14, Shimotsuruma, Yamato-shi  
Kanagawa 242-8502 Japan*

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be

incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

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

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

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

*Intellectual Property Dept. for Rational Software  
IBM Corporation  
5 Technology Park Drive  
Westford, MA 01886  
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 is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the

names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

## **Copyright license**

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

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

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 2009, 2013.

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

## **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).

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

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

Windows is a trademark of Microsoft Corporation in the United States, other countries, or both.

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

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Other product and service names might be trademarks of IBM or other companies.

---

## 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.

---

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

IBM Rational Developer for System z  
Version 9.0.1  
Configuration Guide

Publication No. SC23-7658-10

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



Fold and Tape

Please do not staple

Fold and Tape



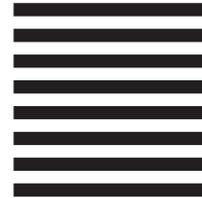
NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES

# 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





Printed in USA

SC23-7658-10

