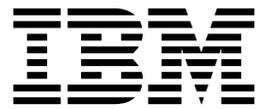


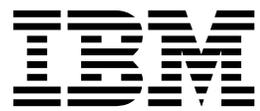
IBM Rational Developer for z Systems
Version 9.5.1

Host Configuration Reference Guide



IBM Rational Developer for z Systems
Version 9.5.1

Host Configuration Reference Guide



Note

Before using this information, be sure to read the general information under “Notices” on page 53.

Tenth edition (September, 2015)

| This edition applies to IBM Rational Developer for z Systems Version 9.5 (program number 5724-T07, or part of
| program number 5697-CDT) 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 2015, 2015.

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

Contents

Figures v

Tables vii

About this document ix

Who should use this document ix

Summary of changes x

Description of document content xii

 Understanding Developer for z Systems. xii

 Security considerations xii

 TCP/IP considerations xii

 WLM considerations xii

 Push-to-client considerations xii

 CICSTS considerations xii

 Setting up AT-TLS xii

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

**Chapter 1. Understanding Developer for
z Systems. 3**

Component overview 3

Task owners 4

Integrated debugger 6

CARMA 7

 CARMA configuration files 7

 CRASTART 8

 Batch submit 8

z/OS UNIX directory structure 8

Chapter 2. Security considerations. 11

Authentication methods 11

 Debug Manager authentication 11

Connection security 12

 Integrated Debugger encrypted communication 12

Debug security 12

CICSTS security 13

SCLM security 13

Security definitions 13

 Requirements and checklist 14

 Activate the security settings and classes 14

 Define an OMVS segment for Developer for z

 Systems users 15

 Define the Developer for z Systems started tasks 15

 Define Debug Manager as a secure z/OS UNIX

 server 16

 Define the MVS program controlled libraries for

 Debug Manager 16

 Define the PassTicket support for RSE 17

 Define z/OS UNIX file access permission for RSE 18

 Define the application protection for RSE 18

 Define the z/OS UNIX program controlled files

 for RSE. 18

 Define the JES command security 19

 Define access to Integrated Debugger. 20

 Define the data set profiles 21

 Verify the security settings 22

Chapter 3. TCP/IP considerations 23

TCP/IP ports 23

 External communication 23

 Internal communication 24

 TCP/IP port reservation 24

CARMA and TCP/IP 25

 CARMA and TCP/IP ports 25

 CARMA and stack affinity 25

 crastart*.conf 26

 CRASUB* 26

Chapter 4. WLM considerations 27

Workload classification 27

 Classification rules 28

Setting goals 29

 Considerations for goal selection 30

 STC 31

 OMVS 31

 JES 32

**Chapter 5. Push-to-client
considerations 33**

Introduction 33

Host-based projects. 34

Chapter 6. CICSTS considerations 35

Bidirectional language support 35

Diagnostic IRZ messages for Enterprise Service

Tools 35

CICS transaction debugging. 35

Chapter 7. Setting up AT-TLS 37

Setting up syslogd 38

AT-TLS configuration in PROFILE.TCPIP 38

Policy Agent started task 38

Policy Agent configuration 39

AT-TLS policy 39

 TLS v1.2 considerations 41

AT-TLS security updates 41

AT-TLS policy activation 44

Bibliography 45

Referenced publications 45

Informational publications 46

Glossary	49	Terms and conditions for product documentation..	55
Notices	53	Copyright license	56
Programming interface information	55	Trademark acknowledgments	56
Trademarks	55	Index	57

Figures

1. Component overview	3	5. z/OS UNIX directory structure	8
2. Task owners.	5	6. AT-TLS policy for Debug Manager.	12
3. Integrated debugger	6	7. TCP/IP ports	23
4. CARMA flow	7	8. WLM classification	27

Tables

1.	SAF information for debug functions	13		8.	WLM workloads - STC.	31
2.	Security setup variables	14		9.	WLM workloads - OMVS.	31
3.	JES2 Job Monitor operator commands.	20		10.	WLM workloads - JES	32
4.	JES3 Job Monitor operator commands.	20		11.	Referenced publications	45
5.	WLM entry-point subsystems	28		12.	Referenced Web sites	46
6.	WLM work qualifiers	28		13.	Informational publications	46
	7.	WLM workloads.	29			

About this document

This document gives background information for various configuration tasks of IBM® Rational® Developer for z Systems™ itself and other z/OS® components and products (such as WLM and TCP/IP).

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

- *IBM Explorer for z/OS* is called *z/OS Explorer*.
- *IBM Rational Developer for z Systems* is called *Developer for z Systems*.
- *IBM Rational Developer for z Systems 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*.
- *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 z Systems host configuration. Each of these documents has a specific target audience. You are not required to read all documents to complete the Developer for z Systems configuration.

- *IBM Rational Developer for z Systems Host Configuration Guide (SC27-8577)* describes in detail all planning tasks, configuration tasks and options (including optional ones) and provides alternative scenarios.
- *IBM Rational Developer for z Systems Host Configuration Reference (SC27-8578)* describes Developer for z Systems design and gives background information for various configuration tasks of Developer for z Systems, z/OS components, and other products (such as WLM and TCP/IP) related to Developer for z Systems.
- *IBM Rational Developer for z Systems Host Configuration Quick Start Guide (GI11-9201)* describes a minimal setup of Developer for z Systems.

The information in this document applies to all IBM Rational Developer for z Systems Version 9.5.1 packages.

For the most up-to-date versions of this document, see the *IBM Rational Developer for z Systems Host Configuration Reference Guide (SC27-8578)* available at <http://www-05.ibm.com/e-business/linkweb/publications/servlet/pbi.wss?CTY=US&FNC=SRX&PBL=SC27-8578>.

For the most up-to-date versions of the complete documentation, including installation instructions, white papers, podcasts, and tutorials, see the library page of the IBM Rational Developer for z Systems website (http://www-01.ibm.com/software/sw-library/en_US/products/Z964267S85716U24/).

Who should use this document

This document is intended for system programmers configuring and tuning IBM Rational Developer for z Systems Version 9.5.1.

While the actual configuration steps are described in another publication, this publication lists in detail various related subjects, such as tuning, security setup, and more. 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 z Systems Version 9.5.1 Host Configuration Reference*, SC27-8578-00 (updated December 2015).

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

New information:

- Use new MVS data set names and z/OS UNIX paths

Removed information:

In version 9.5.1, the RSE and JES Job Monitor related functions moved from IBM Rational Developer for z Systems to another product, IBM Explorer for z/OS. This move includes the related documentation.

- RSE specific data is removed from all chapters
- JES Job Monitor specific data is removed from all chapters
- TSO command service specific data is removed from all chapters
- Push-to-client data for configuration and version management is removed from all chapters
- Documentation on how to set up TCP/IP is removed

This document contains information previously presented in *IBM Rational Developer for z Systems Version 9.5 Host Configuration Reference*, SC14-7290-09..

New information:

- Documented security checks for the new “send message” facility. See Send message security
- Added details on used JES operator commands. See Actions against jobs - execution limitations
- Added information on push-to-client group name limitations. See Group name limitations
- Added information on SYSPLEX limitations. See SYSPLEX
- Added information for managing encryption protocols and ciphers. See Manage encryption protocols and ciphers
- Added instructions for a simple multi-server setup. See Identical software level, different configuration files

Removed information:

- Application Deployment Manager is no longer provided, so all information about it is removed.

This document contains information previously presented in *IBM Rational Developer for System z Version 9.1.1 Host Configuration Reference*, SC14-7290-08.

New information:

- Updated Integrated Debugger security profiles. See “Debug security” on page 12.
- Added information on passphrase support. See “Authentication methods” on page 11.

This document contains information previously presented in *IBM Rational Developer for System z Version 9.1.1 Host Configuration Reference*, SC14-7290-07.

New information:

- Added information on log file security. See Log file security.
- Added information on group support for rejecting push-to-client updates. See Multiple developer groups.
- Updated resource usage information. See Tuning considerations.
- Updated log file and trace information. See Troubleshooting configuration problems.

This document contains information previously presented in *IBM Rational Developer for System z Version 9.0.1 Host Configuration Reference*, SC14-7290-06.

New information:

- Added information on setting up AT-TLS. See Chapter 7, “Setting up AT-TLS,” on page 37.

This document contains information previously presented in *IBM Rational Developer for System z Version 9.0.1 Host Configuration Reference*, SC14-7290-05.

New information:

- Added information on time-stamped log file names. See Log files.
- Added information on new auditable events. See Audit data.

This document contains information previously presented in *IBM Rational Developer for System z Version 9.0 Host Configuration Reference*, SC14-7290-04.

New information:

- Updated TCP/IP port usage. See “TCP/IP ports” on page 23.
- Added sample to automatically synchronize 2 RSE daemons. See Automated synchronizing.
- Added information about new log files. See Log files.

This document contains information previously presented in *IBM Rational Developer for System z Version 8.5.1 Host Configuration Reference*, SC14-7290-03.

New information:

- Added information about SAF profiles to alter client functions. See Altering client functions.
- Updated resource usage numbers. See Tuning considerations
- Updated default value for maximum number of users per threadpool. See Tuning considerations.

This document contains information previously presented in *IBM Rational Developer for System z Version 8.5 Host Configuration Reference*, SC14-7290-02.

New information:

- Updated JES Job Monitor security information. See Chapter 2, “Security considerations,” on page 11.
- Added information about user exits. See User exit considerations.

Description of document content

This section summarizes the information presented in this document.

Understanding Developer for z Systems

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

Security considerations

Developer for z Systems interacts with other host components, which has security implications.

TCP/IP considerations

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

Push-to-client considerations

Developer for z Systems extends the z/OS Explorer push-to-client, or host-based client control, with support for project definitions.

CICSTS considerations

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

Setting up AT-TLS

This section is provided to assist you with some common problems that you may encounter when setting up Application Transparent Transport Layer Security (AT-TLS), or during checking or modifying an existing setup.

IBM Rational Developer for System z Host Configuration Reference Guide

Chapter 1. Understanding Developer for z Systems

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

The following topics are covered in this chapter:

- “Component overview”
- “Task owners” on page 4
- “Integrated debugger” on page 6
- “CARMA” on page 7
- “z/OS UNIX directory structure” on page 8

Developer for z Systems builds on top of IBM Explorer for z/OS . For z/OS Explorer related information, see “Security consideration” in the *IBM Explorer for z/OS Host Configuration Reference (SC27-8438)*.

Component overview

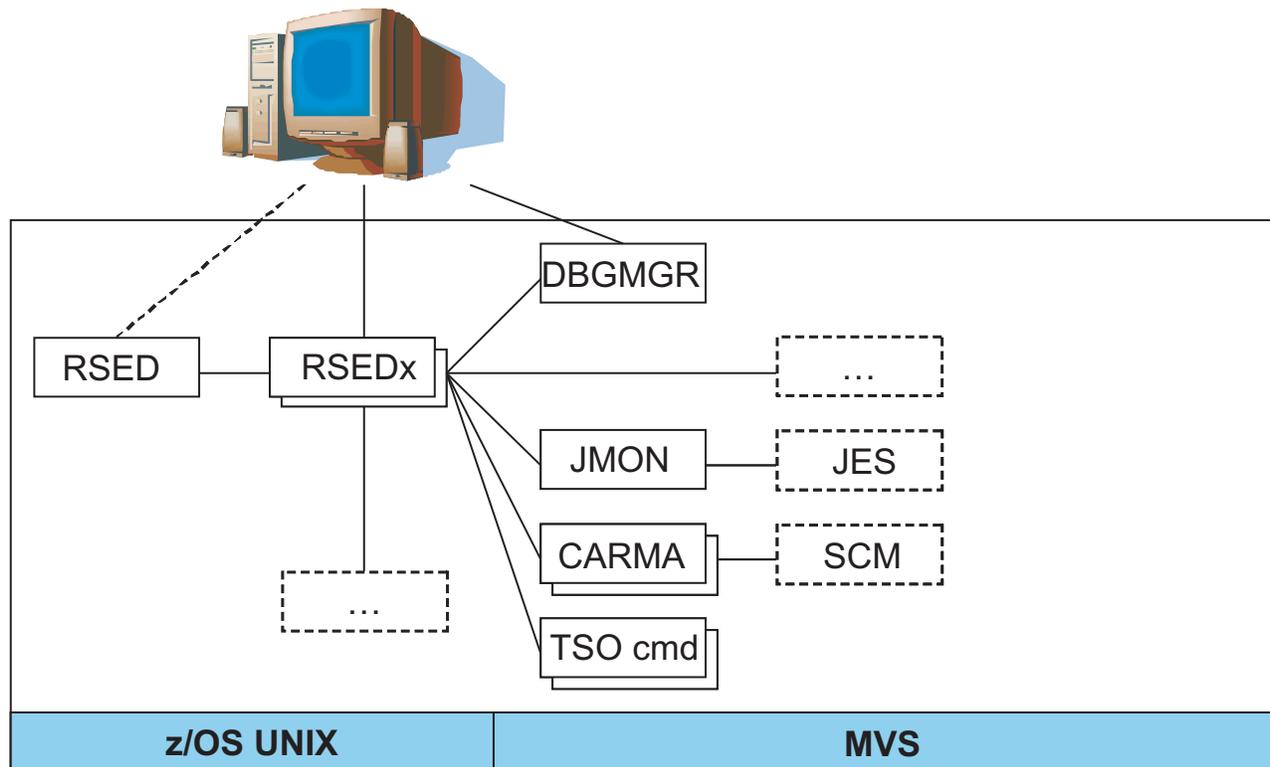


Figure 1. Component overview

Figure 1 shows a generalized overview of the combined z/OS Explorer and Developer for z Systems layout on your host system.

- Remote Systems Explorer (RSE) provides core services, such as connecting the client to the host and starting other servers for specific services. RSE consists of two logical entities:
 - RSE daemon (RSED), which manages connection setup. RSE daemon is also responsible for running in single server mode. To do so, RSE daemon creates one or more child processes known as RSE thread pools (RSEDx).
 - RSE server, which handles individual client request. An RSE server is active as a thread inside a RSE thread pool.
- Debug Manager (DBGMGR) coordinates Integrated Debugger activity.
- (z/OS Explorer) TSO Commands Service (TSO cmd) provides a batch-like interface for TSO and ISPF commands.
- (z/OS Explorer) JES Job Monitor (JMON) provides all JES related services.
- Common Access Repository Manager (CARMA) provides an interface to interact with Software Configuration Managers (SCMs), such as CA Endeavor.
- More services are available, which can be provided by Developer for z Systems itself or corequisite software.

The description in the previous paragraph and list shows the central role assigned to RSE. With few exceptions, all client communication goes through RSE. This allows for easy security related network setup, as only a limited set of ports are used for client-host communication.

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. Based upon the values defined in the `rse.env` configuration file, and the amount of actual client connections, multiple thread pool address spaces can be started by the daemon.

Task owners

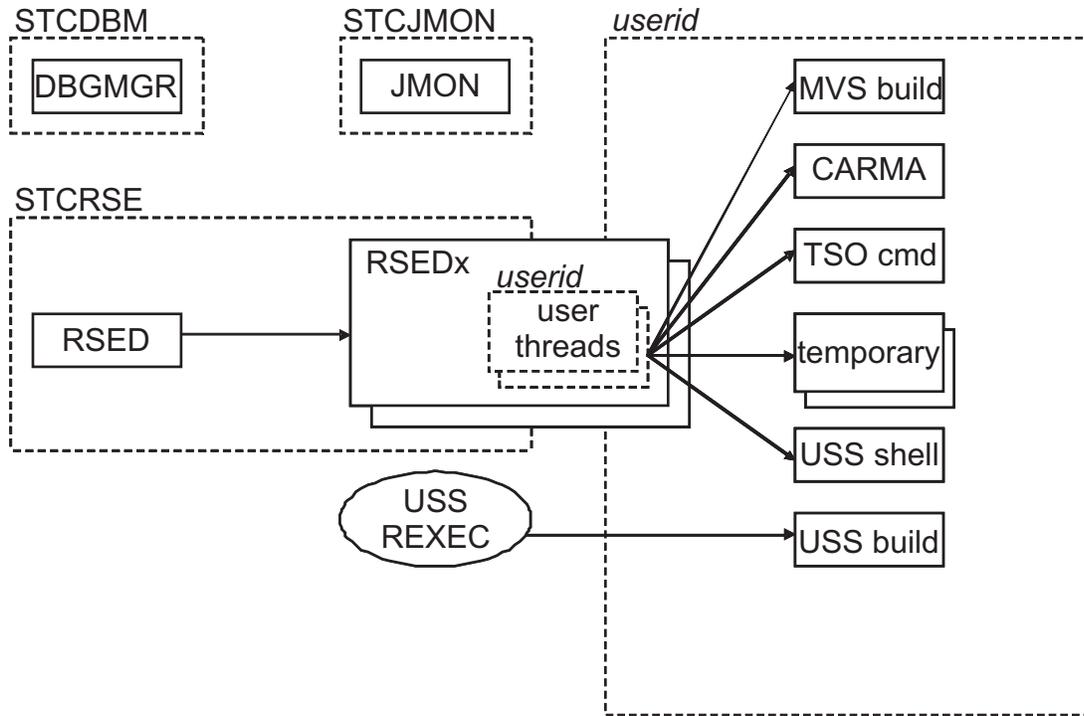


Figure 2. Task owners

Figure 2 shows a basic overview of the owner of the security credentials used for various z/OS Explorer and Developer for z Systems tasks.

The ownership of a task can be divided into two sections. Started tasks are owned by the user ID that is assigned to the started task in your security software. All other tasks, with the RSE thread pools (RSEDx) as exception, are owned by the client user ID.

Figure 2 shows the z/OS Explorer and Developer for z Systems started tasks (DBGMGR, JMON, and RSED), and sample started tasks and system services that Developer for z Systems communicates with. The USS REXEC tag represents the z/OS UNIX REXEC (or SSH) service.

RSE daemon (RSED) creates one or more RSE thread pool address spaces (RSEDx) to process client requests. Each RSE thread pool supports multiple clients and is owned by the same user as the RSE daemon. Each client has his own threads inside a thread pool, and these threads are owned by the client user ID.

Depending on actions done by the client, one or more additional address spaces can be started, all owned by the client user ID, to perform the requested action. These address spaces can be an MVS batch job, an APPC transaction, or a z/OS UNIX child process. Note that a z/OS UNIX child process is active in a z/OS UNIX initiator (BPXAS), and the child process shows up as a started task in JES.

The creation of these address spaces is most often triggered by a user thread in a thread pool, either directly or by using system services like ISPF. But the address space could also be created by a third party. For example, z/OS UNIX REXEC or SSH are involved when starting builds in z/OS UNIX.

The user-specific address spaces end at task completion or when an inactivity timer expires. The started tasks remain active. The address spaces listed in Figure 2 on page 5 remain in the system long enough to be visible. However, you should be aware that due to the way z/OS UNIX is designed, there are also several short-lived temporary address spaces.

Integrated debugger

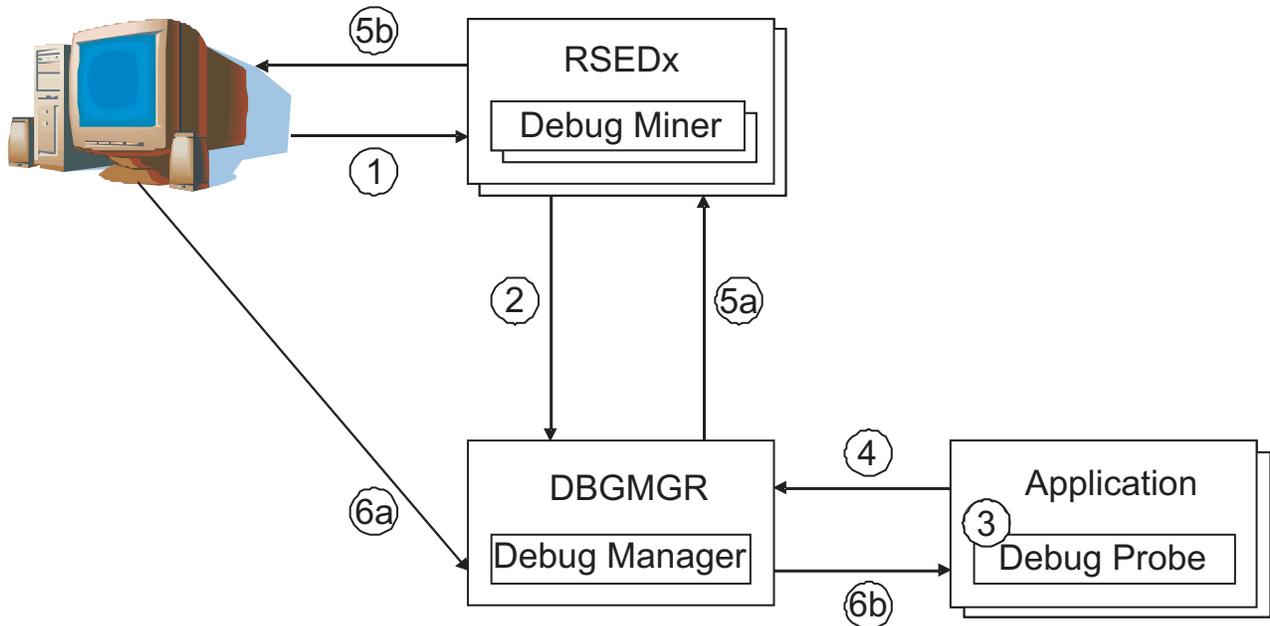


Figure 3. Integrated debugger

The Integrated Debugger is used to debug various applications. Figure 5 shows a schematic overview of how a Developer for z Systems client can debug an application.

1. The client connects to the host, using the normal Developer for z Systems host logon.
2. As part of the logon, a debug miner will register the user with the debug manager, which is active within the DBGMGR started task.
3. When an application is started with an indicator that it should be debugged, Language Environment® (LE) will invoke the debug probe.
4. The debug probe will register with the debug manager.
5. Using the debug miner, the debug manager will notify the Developer for z Systems client of the user who will receive this debug session. If the user is not registered at this moment, the debug session goes dormant, waiting for the user to register with the debug manager.
6. The debug engine within the client contacts the debug manager, which in turn will pass the data back and forth between the debug engine and debug probe.

CARMA

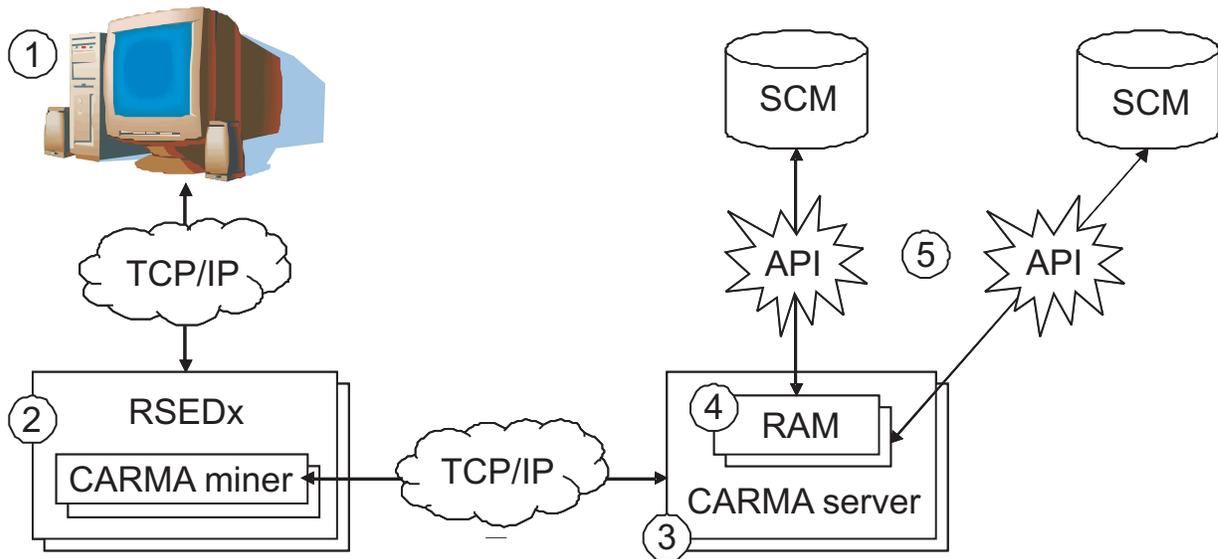


Figure 4. CARMA flow

CARMA (Common Access Repository Manager) is used to access a host based Software Configuration Manager (SCM), for example CA Endeavor® SCM. Figure 4 shows a schematic overview of how a Developer for z Systems client can access any supported host-based Software Configuration Manager (SCM).

1. The client has a Common Access Repository Manager (CARMA) plugin.
2. The CARMA plugin communicates with the CARMA miner, active as a user-specific thread within the RSE thread pool (RSEDx). This communication is done by way of the existing RSE connection.
3. When the client requests access to a SCM, the CARMA miner will bind to a TCP/IP port and will start a user-specific CARMA server, with the port number as startup argument. The CARMA server will then connect to this port and use this path for communication with the client. Note that host based SCMs expect single-user address spaces to access their services, which requires CARMA to start a CARMA server per user. It is not possible to create a single server supporting multiple users.
4. The CARMA server will load the Repository Access Manager (RAM) that supports the requested SCM.
5. The RAM deals with the technical details of interacting with the specific SCM, and presents a common interface to the client.

CARMA configuration files

Developer for z Systems supports multiple methods to start a CARMA server. Each method has benefits and drawbacks. Developer for z Systems also provides multiple Repository Access Managers (RAMs), which can be divided into two groups, production RAMs and sample RAMs. Various combinations of RAMs and server startup methods are available as a preconfigured setup.

All server startup methods share a common configuration file, CRASRV.properties, which (among other things) specifies which startup method will be used.

CRASTART

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

RSE invokes load module CRASTART, which uses the definitions in `crastart*.conf` to create a valid environment to execute batch TSO and ISPF commands. Developer for z Systems uses this environment to run the CARMA server, CRASERV. Developer for z Systems provides multiple `crastart*.conf` files, each preconfigured for a specific RAM.

Batch submit

The "batch submit" method starts the CARMA server by submitting a job. This is the default method 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 per developer starting a CARMA server.

RSE invokes CLIST CRASUB*, which in turn submits an embedded JCL to create a valid environment to execute batch TSO and ISPF commands. Developer for z Systems uses this environment to run the CARMA server, CRASERV. Developer for z Systems provides multiple CRASUB* members, each preconfigured for a specific RAM.

z/OS UNIX directory structure

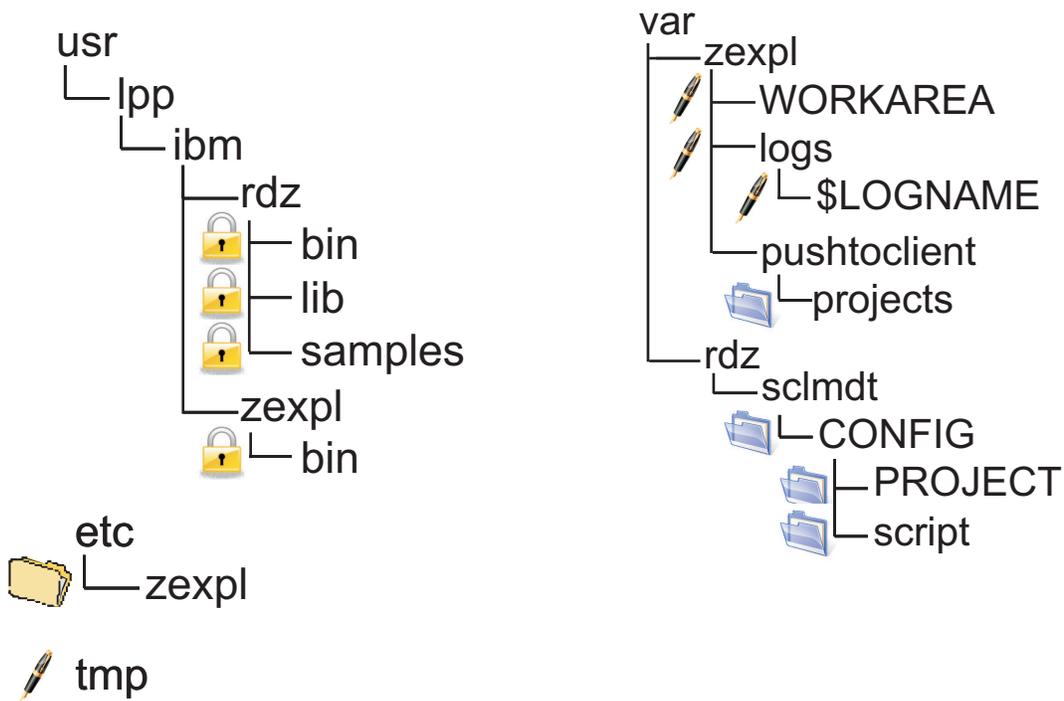


Figure 5. z/OS UNIX directory structure

Figure 5 on page 8 shows an overview of the z/OS UNIX directories used by Developer for z Systems. The following list describes each directory touched by Developer for z Systems, how the location can be changed, and who maintains the data within.

- /usr/lpp/ibm/rdz/ is the root path for the Developer for z Systems product code. The actual location is specified in the rdz.env configuration file (variable RDZ_HOME). The files within are maintained by SMP/E.
- Developer for z Systems places files in /usr/lpp/ibm/zexpl/bin, the binaries directory of z/OS Explorer. The actual location is specified in the z/OS Explorer configuration. The files within are maintained by SMP/E.
- /etc/zexpl/ holds the z/OS Explorer and Developer for z Systems configuration files. The actual location is specified in the RSED started task (variable CNFG). The files within are maintained by the system programmer.
- /tmp/ is used by Legacy ISPF Gateway to store temporary data. Some IVPs store their output here. The files within are maintained by ISPF and the IVPs. The location can be customized with the TMPDIR variable in rse.env. It is also the default location for Java™ dump files, which can be customized with the _CEE_DUMPTARG variable in rse.env.

Note: /tmp/ requires permission bit mask 777 to allow each client to create temporary files.

- /var/zexpl/WORKAREA/ is used by Legacy ISPF Gateway and SCLMDT to transfer data between z/OS UNIX and MVS based address spaces. The actual location is specified in rse.env (variable CGI_ISPWORK). The files within are maintained by ISPF and SCLMDT.

Note: /var/zexpl/WORKAREA/ requires permission bit mask 777 to allow each client to create temporary files.

Developer for z Systems writes log messages in the z/OS Explorer log files located in /var/zexpl/zexpl/logs/\$LOGNAME. The actual location is specified in the z/OS Explorer configuration. The files within are maintained by z/OS Explorer and Developer for z Systems product code.

- /var/rdz/sclmdt/CONFIG/ holds general SCLMDT configuration files. The actual location is specified in rdz.env (variable SCLMDT_CONF_HOME). The files within are maintained by the SCLM administrator.
- /var/rdz/sclmdt/CONFIG/PROJECT/ holds SCLMDT project configuration files. The actual location is specified in rdz.env (variable SCLMDT_CONF_HOME). The files within are maintained by the SCLM administrator.
- /var/rdz/sclmdt/CONFIG/script/ holds SCLMDT-related scripts that can be used by other products. The actual location is not specified anywhere. The files within are maintained by the SCLM administrator.
- /var/rdz/pushtoclient/ holds client configuration files, client product update information, and host-based project information that is pushed to the client upon connection to the host. The actual location is specified in pushtoclient.properties (variable pushtoclient.folder). The files within are maintained by a Developer for z Systems client administrator.
- /var/rdz/pushtoclient/projects/ holds the host-based project definition files. The actual location is specified in /var/rdz/pushtoclient/keymapping.xml, which is created and maintained by a Developer for z Systems client administrator. The files within are maintained by a project manager or lead developer.

Chapter 2. Security considerations

Developer for z Systems extends z/OS Explorer by providing additional functions, some of which interact with other system components and products, such as a Software Configuration Manager (SCM). Developer for z Systems specific security definitions are used to secure the provided functions.

The security mechanisms used by Developer for z Systems servers and services rely on the data sets and file systems it resides in being secure. This implies that only trusted system administrators should be able to update the program libraries and configuration files.

Developer for z Systems builds on top of IBM Explorer for z/OS . For z/OS Explorer related information, see “Security consideration” in the *IBM Explorer for z/OS Host Configuration Reference (SC27-8438)*.

The following topics are covered in this chapter:

- “Authentication methods”
- “Connection security” on page 12
- “Debug security” on page 12
- “CICSTS security” on page 13
- “SCLM security” on page 13
- “Security definitions” on page 13

Authentication methods

CARMA authentication

Client authentication is done by RSE daemon as part of the client's connection request. CARMA is started from a user specific thread, and inherits the user's security environment, bypassing the need for additional authentication.

SCLM Developer Toolkit authentication

Client authentication is done by RSE daemon as part of the client's connection request. SCLMDT is started from a user specific thread, and inherits the user's security environment, bypassing the need for additional authentication.

Debug Manager authentication

Client authentication is done by RSE daemon as part of the client's connection request. After the user is authenticated, self-generated PassTickets are used for all future authentication requests, including the automatic logon to Debug Manager.

In order for Debug Manager to validate the user ID and PassTicket presented by RSE, Debug Manager must be allowed to evaluate the PassTicket. This implies that load module AQEZPCM, by default located in load library FEL.SFEKAUTH, must be APF-authorized.

When a client-based Debug Engine connects to the Debug Manager, it must present a valid security token for authentication.

Connection security

Most communication between Developer for z Systems client and host goes through RSE, thus utilizing the connection security provided by z/OS Explorer.

Some Developer for z Systems services use a separate external (client-host) communication path:

- The Integrated Debugger Engine on the client connects to the Debug Manager on the host. The encryption details are controlled by an Application Transparent Transport Layer Security (AT-TLS) policy.
- Remote (host-based) actions in z/OS UNIX subprojects use an REXEC or SSH server on the host. SSH communication is always encrypted.

Integrated Debugger encrypted communication

External (client-host) communication with the optional Debug Manager can also be encrypted. To enable encryption, create an Application Transparent TLS (AT-TLS) policy for the port used by Debug Manager for external communication, by default 5335. A sample policy is provided in Figure 6. See Chapter 7, “Setting up AT-TLS,” on page 37 for details on setting up AT-TLS.

```
TTLRule                                RDz_Debug_Manager
{
  LocalPortRange                        5335
  Direction                              Inbound
  TTLGroupActionRef                     grp_Production
  TTLEnvironmentActionRef               RDz_Debug_Manager
}
TTLEnvironmentAction                   RDz_Debug_Manager
{
  HandshakeRole Server
  TLSKeyRingParms
  {
    Keyring dbgmgr.racf                 # Keyring must be owned by the Debug Manager
  }
}
TTLGroupAction                          grp_Production
{
  TTLEnabled                            On
  Trace                                  2
}
```

Figure 6. AT-TLS policy for Debug Manager

Note: The communication method used by the Debug Engine on the Developer for z Systems client to talk to the Debug Manager on the host is by default tied to the communication method used by the Developer for z Systems client to talk to the RSE daemon. This implies that if encryption is enabled for RSE, it is assumed it is also enabled for Debug Manager. However, there are alternate scenario’s available for other setups.

Debug security

The optional Integrated Debugger requires that users have sufficient access permits to specified security profiles. If the user does not have the required permit, the debug session will not start.

Developer for z Systems verifies access to the profiles listed in Table 1 on page 13 to determine which debug permits are granted.

Table 1. SAF information for debug functions

FACILITY profile	Required access	Result
AQE.AUTHDEBUG.STDPGM	READ	User is able to debug problem-state applications
AQE.AUTHDEBUG.AUTHPGM	READ	User is able to debug problem-state and authorized applications

Note:

- Developer for z Systems assumes that a user has no access authorization when your security software indicates it cannot determine whether or not a user has access authorization to a profile. An example of this is when the profile is not defined.
- Developer for z Systems versions that pre-date version 9.1.1 checked for UPDATE permission to profile AQE.AUTHDEBUG.WRITEBUFFER to allow debugging of read-only CICS transactions. This profile is no longer used and can be removed if your host system only has Developer for z Systems version 9.1.1 or higher.

The following sample security definitions allow all users in group RDZDEBUG to debug problem-state applications:

```
RDEFINE FACILITY (AQE.AUTHDEBUG.STDPGM) -
  UACC(NONE) DATA('RATIONAL DEVELOPER FOR Z SYSTEMS - DEBUG PROBLEM-STATE')
PERMIT AQE.AUTHDEBUG.STDPGM CLASS(FACILITY) -
  ID(RDZDEBUG) ACCESS(READ)
SETROPTS RACLIST(FACILITY) REFRESH
```

CICSTS security

The optional Integrated Debugger is capable of debugging CICS transactions. See “CICS transaction debugging” on page 35 for more details.

SCLM security

The SCLM Developer Toolkit service offers optional security functionality for the Build, Promote, and Deploy functions.

If security is enabled for a function by the SCLM administrator, SAF calls are made to verify authority to execute the protected function with the caller’s or a surrogate user ID.

Refer to *SCLM Developer Toolkit Administrator’s Guide* (SC23-9801), for more information on the required SCLM security definitions.

Security definitions

| Customize and submit the sample FELRACF job, which has sample RACF®
 | commands to create the basic security definitions for Developer for z Systems.
 | Customize and submit the sample AQERACF job, which has sample RACF
 | commands to create the security definitions for Integrated Debugger.

| FELRACF and AQERACF are is located in FEL.#CUST.JCL, unless you specified a
 | different location when you customized and submitted the

FEL.SFELSAMP(FELSETUP) job. See "Customization setup" in the *Rational Developer for z Systems Host Configuration Guide* for more details.

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

Requirements and checklist

To complete the security setup, the security administrator must know the values that are listed in Table 2. These values were defined during previous steps of the installation and customization of Rational Developer for z Systems.

Table 2. Security setup variables

Description	<ul style="list-style-type: none"> • Default value • Where to find the answer 	Value
Developer for z Systems product high-level qualifier	<ul style="list-style-type: none"> • FEL • SMP/E installation 	
Developer for z Systems customization high-level qualifier	<ul style="list-style-type: none"> • FEL.#CUST • FEL.SFELSAMP(FELSETUP), as described in "Customization setup" in the <i>Rational Developer for z Systems Host Configuration Guide</i>. 	
Integrated Debugger started task name	<ul style="list-style-type: none"> • DBGMGR • FEL.#CUST.PROCLIB(DBGMGR), as described in "PROCLIB changes" in the <i>Rational Developer for z Systems Host Configuration Guide</i> 	

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

- "Activate the security settings and classes"
- "Define the Developer for z Systems started tasks" on page 15
- "Define Debug Manager as a secure z/OS UNIX server" on page 16
- "Define the MVS program controlled libraries for Debug Manager" on page 16
- "Define access to Integrated Debugger" on page 20
- "Define the data set profiles" on page 21
- "Verify the security settings" on page 22

Activate the security settings and classes

Developer for z Systems 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 Integrated Debugger
 - SETROPTS GENERIC(FACILITY)
 - SETROPTS CLASSACT(FACILITY) RACLIST(FACILITY)
- Activate started task definitions for Integrated Debugger
 - SETROPTS GENERIC(STARTED)
 - RDEFINE STARTED ** STDATA(USER(=MEMBER) GROUP(STCGROUP) TRACE(YES))
 - SETROPTS CLASSACT(STARTED) RACLIST(STARTED)
- Activate program control for Integrated Debugger
 - RDEFINE PROGRAM ** ADDMEM('SYS1.COMDLIB'//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.

Define an OMVS segment for Developer for z Systems 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 z Systems. 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 z Systems started tasks

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

- ```
ADDGROUP STCGROUP OMVS(AUTOGID)
DATA('GROUP WITH OMVS SEGMENT FOR STARTED TASKS')
```
- ```
ADDUSER STCDBM DFLTGRP(STCGROUP) NOPASSWORD NAME('DEBUG MANAGER')
OMVS(AUTOUID HOME(/tmp) PROGRAM(/bin/sh) )
DATA('Rational Developer for z Systems')
```
- ```
RDEFINE STARTED DBGMGR.* DATA('DEBUG MANAGER')
STDATA(USER(STCDBM) GROUP(STCGROUP) TRUSTED(NO))
```
- SETROPTS RACLIST(STARTED) REFRESH

**Note:**

- Ensure that the started tasks user IDs are protected by specifying the NOPASSWORD keyword.
- The Debug Manager started task (DBGMGR) is used only by the Integrated Debugger feature.

## Define Debug Manager as a secure z/OS UNIX server

Integrated Debugger requires UPDATE access to the BPX.SERVER profile to create or delete the security environment for the debug thread. Note that using UID(0) to bypass this requirement is not supported. 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(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 Debug Manager

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

Debug Manager uses system libraries, Language Environment's runtime, and the Developer for z Systems' (ISP.SISPLOAD) load library.

- RALTER PROGRAM \*\* UACC(READ) ADDMEM('SYS1.LINKLIB'//NOPADCHK)
- RALTER PROGRAM \*\* UACC(READ) ADDMEM('SYS1.CSSLIB'//NOPADCHK)
- RALTER PROGRAM \*\* UACC(READ) ADDMEM('CEE.SCEERUN'//NOPADCHK)
- RALTER PROGRAM \*\* UACC(READ) ADDMEM('CEE.SCEERUN2'//NOPADCHK)
- RALTER PROGRAM \*\* UACC(READ) ADDMEM('FEL.SFELAUTH'//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 z Systems interacts with, such as IBM Explorer for z/OS.

- Alternate REXX runtime library, for SCLM Developer Toolkit
  - REXX.\*.SEAGALT

**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:

- REXX runtime library, for SCLM Developer Toolkit
  - REXX.\*.SEAGLPA
- Developer for z Systems, for CARMA
  - FEL.SFELLPA

## 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 Z SYSTEMS')
- RDEFINE PTKTDATA IRRPTAUTH.FEKAPPL.\* UACC(NONE)  
DATA('RATIONAL DEVELOPER FOR Z SYSTEMS')
- PERMIT IRRPTAUTH.FEKAPPL.\* CLASS(PTKTDATA) ACCESS(UPDATE) ID(STCRSE)
- SETROPTS RACLIST(PTKTDATA) REFRESH

RSE supports the using of an application ID other than FEKAPPL. Uncomment and customize the "APPLID=FEKAPPL" option in `rdz.env` to activate this, as documented in "Defining extra Java startup parameters with `_RSE_JAVAOPTS`" in the *IBM Rational Developer for z Systems Host Configuration Guide*. 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 z/OS UNIX file access permission for RSE

The **MODIFY LOGS** operator command uses the RSED started task user ID to collect host logs and setup information. And by default, user log files are created with secure file access permissions (only owner has access). To be able to collect secure user log files, the RSED started task user ID must be permitted to read them.

The **OWNER** argument of the **MODIFY LOGS** operator command results in the specified user ID becoming the owner of the collected data. In order to change ownership, the RSED started task user ID must be permitted to use the **CHOWN** z/OS UNIX service.

- `RDEFINE UNIXPRIV SUPERUSER.FILESYS UACC(NONE) DATA('OVERRIDE UNIX FILE ACCESS RESTRICTIONS')`
- `RDEFINE UNIXPRIV SUPERUSER.FILESYS.CHOWN UACC(NONE) DATA('OVERRIDE UNIX CHANGE OWNER RESTRICTIONS')`
- `PERMIT SUPERUSER.FILESYS CLASS(UNIXPRIV) ACCESS(READ) ID(STCRSE)`
- `PERMIT SUPERUSER.FILESYS.CHOWN CLASS(UNIXPRIV) ACCESS(READ) ID(STCRSE)`
- `SETROPTS RACLIST(UNIXPRIV) REFRESH`

Note that when the `SUPERUSER.FILESYS.ACLOVERRIDE` profile is defined, access permissions defined in **ACL** (access Control List) take precedence over the permissions granted through `SUPERUSER.FILESYS`. The RSED started task user ID will need **READ** access permit to the `SUPERUSER.FILESYS.ACLOVERRIDE` profile to bypass **ACL** definitions.

## 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 Z SYSTEMS')`
- `SETROPTS RACLIST(APPL) REFRESH`

### Note:

- As described in more detail in “Define the PassTicket support for RSE” on page 17, 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 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 z Systems 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
```

## 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 "FEJCNFG, JES Job Monitor configuration file" in the *Rational Developer for z Systems Host Configuration Guide*.

The following sample RACF commands give Developer for z Systems 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 Z SYSTEMS')`
- `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 3 on page 20 and Table 4 on page 20 show the operator commands issued for JES2 and JES3, and the discrete security profiles that can be used to protect them.

Table 3. JES2 Job Monitor operator commands

| Action  | Command                               | OPERCMD5 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 4. 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 access to Integrated Debugger

Users require READ access to one of the listed AQE.AUTHDEBUG.\* profiles to be able to use the Integrated Debugger for debugging problem-state programs. Users permitted to the AQE.AUTHDEBUG.AUTHPGM profile are also allowed to debug APF authorized programs. Replace the #apf placeholder with valid user IDs or RACF group names for those users that are allowed to debug authorized programs.

- RDEFINE FACILITY AQE.AUTHDEBUG.STDPGM UACC(NONE)
- PERMIT AQE.AUTHDEBUG.STDPGM CLASS(FACILITY) ACCESS(READ) ID(\*)
- RDEFINE FACILITY AQE.AUTHDEBUG.AUTHPGM UACC(NONE)
- PERMIT AQE.AUTHDEBUG.AUTHPGM CLASS(FACILITY) ACCESS(READ) ID(#apf)
- SETROPTS RACLIST(FACILITY) REFRESH

**Note:** IBM Rational Developer for System z<sup>®</sup> versions that pre-date version 9.1.1 used another FACILITY class profile, AQE.AUTHDEBUG.WRITEBUFFER, which is no longer in use. It can be removed if your host system only has IBM Rational Developer for System z version 9.1.1 or higher.

## Define the data set profiles

READ access for users and ALTER for system programmers is sufficient for most Developer for z Systems 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 FEL.#CUST is the default high-level qualifier for data sets created during the customization process.

- ```

|          ADDGROUP (FEL) OWNER(IBMUSER) SUPGROUP(SYS1)
|          DATA('IBM Rational Developer for z Systems - HLQ STUB')
|
|          •
|          ADDSD 'FEL.*.**' UACC(READ)
|          DATA('IBM Rational Developer for z Systems')
|
|          •
|          PERMIT 'FEL.*.**' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
|
|          •
|          SETROPTS GENERIC(DATASET) REFRESH
      
```

Note:

- Protect FEL.SFELAUTH against updates because this data set is APF-authorized.
- The sample commands in this publication and in the FELRACF 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 Developer for z Systems components require additional security data set profiles. Replace the #sysprog and #ram-developer 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, FEL.#CUST.LSTRANS.FILE.

```

|          -
|          ADDSD 'FEL.#CUST.LSTRANS.*.**' UACC(UPDATE)
|          DATA('IBM Rational Developer for z Systems - SCLMDT')
|          -
|          PERMIT 'FEL.#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, FEL.#CUST.CRA*.

```

|          -
      
```

```

|          ADDSD 'FEL.#CUST.CRA*.*' UACC(READ)
|          DATA('IBM Rational Developer for z Systems - CARMA')
|
|          -
|          PERMIT 'FEL.#CUST.CRA*.*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
|
|          -
|          PERMIT 'FEL.#CUST.CRA*.*' CLASS(DATASET) ACCESS(UPDATE) ID(#ram-developer)
|
|          -
|          SETROPTS GENERIC(DATASET) REFRESH

```

Verify the security settings

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

- Security settings and classes
 - SETROPTS LIST
- Started tasks
 - LISTGRP STCGROUP OMVS
 - LISTUSER STCDBM OMVS
 - RLIST STARTED DBGMR.* ALL STDATA
- Debug Manager as a secure z/OS UNIX server
 - RLIST FACILITY BPX.SERVER ALL
- MVS program controlled libraries for Debug Manager
 - RLIST PROGRAM ** ALL
- Integrated Debugger access
 - RLIST FACILITY AQE.** ALL
- Data set profiles
 - LISTGRP FEL
 - LISTDSD PREFIX(FEL) ALL

Chapter 3. TCP/IP considerations

Developer for z Systems 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.

The following topics are covered in this chapter:

- “TCP/IP ports”
- “CARMA and TCP/IP ports” on page 25

Developer for z Systems builds on top of IBM Explorer for z/OS . For z/OS Explorer related information, see “TCP/IP considerations” in the *IBM Explorer for z/OS Host Configuration Reference (SC27-8438)*.

TCP/IP ports

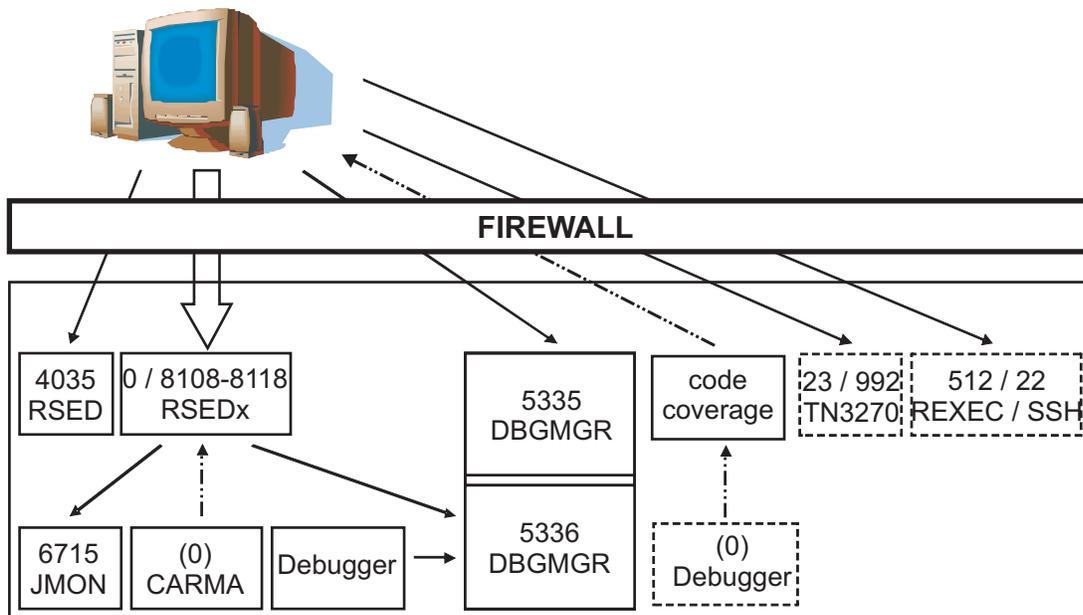


Figure 7. TCP/IP ports

Figure 7 shows the TCP/IP ports that can be used by z/OS Explorer and Developer for z Systems. The arrows show which party does the bind (arrowhead side) and which one connects.

External communication

Define the following ports to your firewall protecting the z/OS host, as they are used for client-host communication (using the tcp protocol):

- (z/OS Explorer) RSE daemon for client-host communication setup, default port 4035. The port can be set in the `rse.env` configuration file. Communication on this port can be encrypted.
- (z/OS Explorer) RSE server for client-host communication. By default, any available port is used, but this can be limited to a specified range with the

| `_RSE_PORTRANGE` definition in `rse.env`. The default port range for `_RSE_PORTRANGE`
| is 8108-8118 (11 ports). Communication on this port can be encrypted.

- | • Debug manager for Integrated Debugger services, default port 5335. The port
| can be set in the `DBGMGR` started task `JCL`. Communication on this port can be
| encrypted.
- | • Either `INETD` service for remote (host-based) actions in z/OS UNIX subprojects:
| – `REXEC` (z/OS UNIX version), default port 512.
| – `SSH` (z/OS UNIX version), default port 22. Communication on this port is
| encrypted.
- | • (z/OS Explorer) `TN3270` Telnet service for the Host Connect Emulator, default
| port 23. Communication can be encrypted (default port 992). The default port
| assigned to the `TN3270` Telnet service depends on whether or not the user
| chooses to use encryption.
- | • Host-based code coverage can be instructed to connect to the Integrated
| Debugger Engine of a Developer for z Systems client. Communication on this
| port can be encrypted. Note that in this scenario, the z/OS-based code coverage
| collector is a client for TCP/IP and the Integrated Debugger Engine on the
| user's personal computer is a server for TCP/IP. The default is to work with
| IBM Debug Tool locally on the same host.

Note: Normally the client specifies which TCP/IP address is used to connect to
the host. However, to ensure that debug sessions communicate with the correct
host, the Debug Manager dictates to the client which TCP/IP address must be
used.

Internal communication

Several Developer for z Systems host services run in separate threads or address
spaces and are using TCP/IP sockets as communication mechanism, using your
system's loopback address. All these services use RSE for communicating with the
client, making their data stream confined to the host only. For some services any
available port will be used, for others the system programmer can choose the port
or port range that will be used:

- | • JES Job Monitor for JES-related services, default port 6715. The port can be set in
| the `FEJJCNFG` configuration member and is repeated in the `rse.env` configuration
| file.
- | • (optional) `CARMA` communication uses by default an ephemeral port, but a port
| range can be set in the `CRASRV.properties` configuration file.
- | • (optional) Debug Manager for debug related services, default port 5336. The port
| can be set in the `DBGMGR` started task `JCL`.
- | • Host-based code coverage, which is a batch job, allocates an ephemeral port to
| allow the IBM Debug Tool for z/OS to communicate with it and deliver data
| needed for the code coverage report.

TCP/IP port reservation

| If you use the `PORT` or `PORTRANGE` statement in `PROFILE.TCPIP` to reserve the ports
| used by z/OS Explorer and Developer for z Systems, note that many binds are
| done by threads active in an RSE thread pool. The job name of the RSE thread pool
| is `RSEdx`, where `RSED` is the name of the RSE started task, and `x` is a random single
| digit number, so wildcards are required in the definition.

```
| PORT      4035      TCP RSED ; z/OS Explorer - RSE daemon  
| PORT      6715      TCP JMON  ; z/OS Explorer - JES job monitor  
| PORT      5335      TCP DBGMGR ; Developer for z Systems - Integrated debugger
```

```

|          PORT      5336      TCP DBGMGR ; Developer for x Systems - Integrated debugger
|          PORTRange 8108 11  TCP RSED*  ; z/OS Explorer - RSE_PORTRANGE
|          ;PORTRange 5227 100 TCP RSED*  ; Developer for z Systems - CARMA

```

CARMA and TCP/IP

CARMA and TCP/IP ports

CARMA (Common Access Repository Manager) is used to access a host-based Software Configuration Manager (SCM), for example CA Endeavor® SCM. In most cases, like for RSE daemon, a server binds to a port and listens for connection requests. CARMA however uses a different approach, as the CARMA server is not active yet when the client initiates the connection request.

When the client sends a connection request, the CARMA miner, which is active as a user thread in an RSE thread pool, will request an ephemeral port or find a free port in the range specified in the CRASRV.properties configuration file and binds to it. The miner then starts the CARMA server and passes the port number, so that the server knows to which port to connect. When the server is connected, the client can send requests to the server and receive the results.

From a TCP/IP perspective, RSE (by way of the CARMA miner) is the server that binds to the port, and the CARMA server is the client connecting to it.

If you use the PORT or PORTRANGE statement in PROFILE.TCPIP to reserve the port range used by CARMA, note that the CARMA miner is active in an RSE thread pool. The jobname of the RSE thread pool is RSEDx, where RSED is the name of the RSE started task and x is a random single digit number, so wildcards are required in the definition.

```

PORTRange 5227 100 RSED*          ; DEVELOPER FOR Z SYSTEMS - CARMA

```

Note: The CARMA IVP, fekfivpc, will fail if you reserve the CARMA ports for usage by the RSE address spaces. This is to be expected because the IVP runs in the address space of the person executing the IVP, not in RSE's address space, and TCP/IP will fail the bind request.

CARMA and stack affinity

CARMA (Common Access Repository Manager) is used to access a host-based Software Configuration Manager (SCM), for example CA Endeavor® SCM. To do so, CARMA starts a user-specific server, which needs additional configuration to enforce stack affinity.

```

| Similar to the z/OS Explorer and Developer for z Systems started tasks, stack
| affinity for a CARMA server is set with the _BPXK_SETIBMOPT_TRANSPORT variable,
| which must be passed on to LE (Language Environment). This can be done by
| adjusting the startup command in the active crastart*.conf or CRASUB*
| configuration file.

```

Note:

- The exact name of the configuration file that holds the startup command depends on various choices made by the systems programmer who configured CARMA. Refer to "Chapter 3. (Optional) Common Access Repository Manager (CARMA)" in the *Host Configuration Guide* (SC27-8577) for more information about this.

- `_BPXK_SETIBMOPT_TRANSPORT` specifies the name of the TCP/IP stack to be used, as defined in the `TCPIPJOBNAME` statement in the related `TCPIP.DATA`.
- Coding a `SYSTCPD DD` statement does not set the requested stack affinity.
- By default, CARMA does not use the normal TCP/IP stacks. CARMA uses the loopback address for the communication between CARMA miner and CARMA server. This improves security (only local processes have access to the loopback address) and is likely to prevent the need to add stack affinity to CARMA communication.

crastart*.conf

Replace the following part:

```
... PARM(&CRAPRM1. &CRAPRM2.)
```

with this (where `TCPIP` represents the desired TCP/IP stack):

```
... PARM(ENVAR("_BPXK_SETIBMOPT_TRANSPORT=TCPIP") / &CRAPRM1. &CRAPRM2.)
```

Note: `CRASTART` does not support line continuations, but there is no limit on the accepted line length.

CRASUB*

Replace the following part:

```
... PARM(&PORT &TIMEOUT)
```

with this (where `TCPIP` represents the desired TCP/IP stack):

```
... PARM(ENVAR("_BPXK_SETIBMOPT_TRANSPORT=TCPIP") / &PORT &TIMEOUT)
```

Note: Job submission limits line length to 80 characters. You can break a longer line at a blank () and use a plus (+) sign at the end of the first line to concatenate 2 lines.

Chapter 4. WLM considerations

Unlike traditional z/OS applications, Rational Developer for z Systems is not a monolithic application that can be identified easily to Workload Manager (WLM). Developer for z Systems consists of several components that interact to give the client access to the host services and data. As described in Chapter 1, “Understanding Developer for z Systems,” on page 3, some of these services are active in different address spaces, resulting in different WLM classifications.

The following topics are covered in this chapter:

- “Workload classification”
- “Setting goals” on page 29

Developer for z Systems builds on top of IBM Explorer for z/OS . For z/OS Explorer related information, see “WLM considerations” in the *IBM Explorer for z/OS Host Configuration Reference (SC27-8438)*.

Workload classification

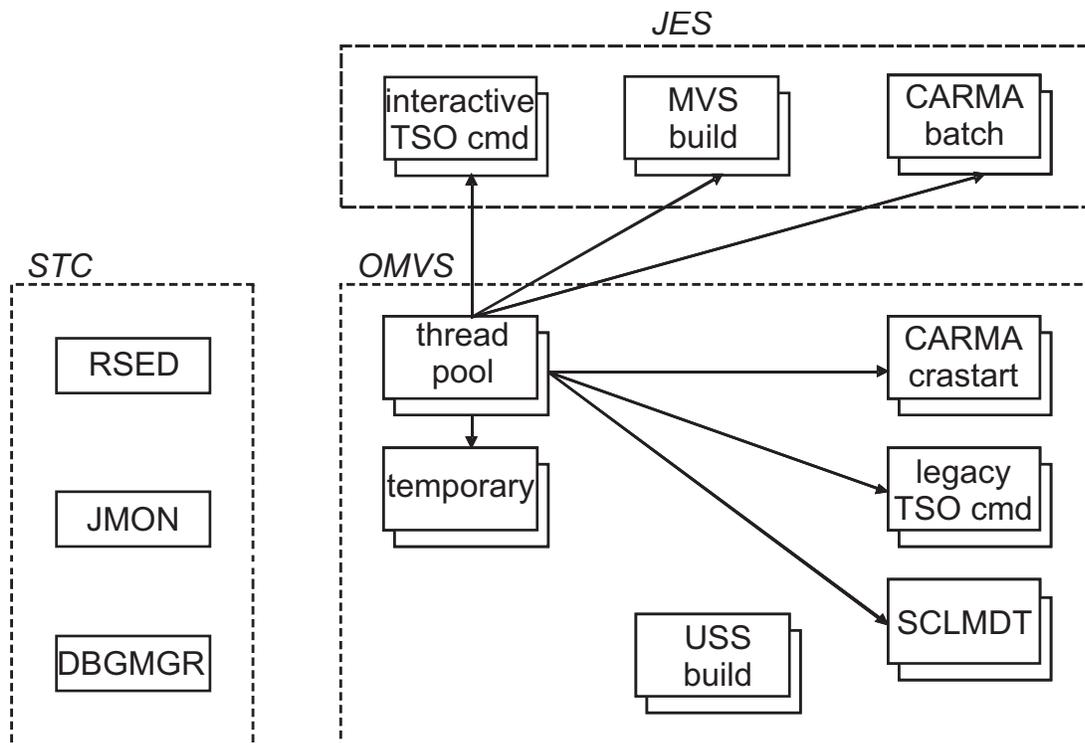


Figure 8. WLM classification

Figure 8 shows a basic overview of the subsystems through which z/OS Explorer and Developer for z Systems workloads are presented to WLM.

RSE daemon (RSED), Debug Manager (DBGMGR) and JES Job Monitor (JMON) are z/OS Explorer and Developer for z Systems started tasks (or long-running batch jobs), each with their individual address space.

RSE daemon spawns a child process for each RSE thread pool server (which supports a variable number of clients). Each thread pool is active in a separate address space (using a z/OS UNIX initiator, BPXAS). Because these are spawned processes, they are classified using the WLM OMVS classification rules, not the started task classification rules.

The clients that are active in a thread pool can create a multitude of other address spaces, depending on the actions done by the users. Depending on the configuration of Developer for z Systems, some workloads, such as the TSO Commands service (TSO cmd) or CARMA, can run in different subsystems.

The address spaces listed in Figure 8 on page 27 remain in the system long enough to be visible, but you should be aware that due to the way z/OS UNIX is designed, there are also several short-lived temporary address spaces. These temporary address spaces are active in the OMVS subsystem.

Note that while the RSE thread pools use the same user ID and a similar job name as the RSE daemon, all address spaces started by a thread pool are owned by the user ID of the client requesting the action. The client user ID is also used as (part of) the job name for all OMVS-based address spaces started by the thread pool.

More address spaces are created by other services that Developer for z Systems uses, such as z/OS UNIX REXEC (USS build).

Classification rules

WLM uses classification rules to map work coming into the system to a service class. This classification is based upon work qualifiers. The first (mandatory) qualifier is the subsystem type that receives the work request. Table 5 lists the subsystem types that can receive Developer for z Systems workloads.

Table 5. WLM entry-point subsystems

Subsystem type	Work description
ASCH	The work requests include all APPC transaction programs scheduled by the IBM-supplied APPC/MVS transaction scheduler, ASCH.
JES	The work requests include all jobs that JES2 or JES3 initiates.
OMVS	The work requests include work processed in z/OS UNIX System Services forked children address spaces.
STC	The work requests include all work initiated by the START and MOUNT commands. STC also includes system component address spaces.

Table 6 lists additional qualifiers that can be used to assign a workload to a specific service class. Refer to MVS Planning: Workload Management (SA22-7602) for more details on the listed work qualifiers.

Table 6. WLM work qualifiers

		ASCH	JES	OMVS	STC
AI	Accounting Information	x	x	x	x
LU	LU Name (*)				
PF	Perform (*)		x		x
PRI	Priority		x		
SE	Scheduling Environment Name		x		

Table 6. WLM work qualifiers (continued)

		ASCH	JES	OMVS	STC
SSC	Subsystem Collection Name		x		
SI	Subsystem Instance (*)		x		
SPM	Subsystem Parameter				x
PX	Sysplex Name	x	x	x	x
SY	System Name (*)	x		x	x
TC	Transaction/Job Class (*)	x	x		
TN	Transaction/Job Name (*)	x	x	x	x
UI	User ID (*)	x	x	x	x

Note: For the qualifiers marked with (*), you can specify classification groups by adding a G to the type abbreviation. For example, a transaction name group would be TNG.

Setting goals

As documented in “Workload classification” on page 27, Developer for z Systems creates different types of workloads on your system. These different tasks communicate with each other, which implies that the actual elapse time becomes important to avoid time-out issues for the connections between the tasks. As a result, Developer for z Systems tasks should be placed in high-performance service classes, or in moderate-performance service classes with a high priority.

A revision, and possibly an update, of your current WLM goals is therefore advised. This is especially true for traditional MVS shops new to time-critical OMVS workloads.

Note:

- The goal information in this section is deliberately kept at a descriptive level, because actual performance goals are very site-specific.
- To help understand the impact of a specific task on your system, terms like minimal, moderate and substantial resource usage are used. These are all relative to the total resource usage of Developer for z Systems itself, not the whole system.

Table 7 lists the address spaces that are used by z/OS Explorer and Developer for z Systems. z/OS UNIX will substitute "x" in the "Task Name" column by a random 1-digit number.

Table 7. WLM workloads

Description	Task name	Workload
Debug Manager	DBGMGR	STC
(z/OS Explorer) JES Job Monitor	JMON	STC
(z/OS Explorer) RSE daemon	RSED	STC
(z/OS Explorer) RSE thread pool	RSEDx	OMVS
(ISPF) Interactive ISPF Gateway (TSO Commands serverice)	<userid>	JES

Table 7. WLM workloads (continued)

Description	Task name	Workload
(ISPF) Legacy ISPF Gateway (TSO Commands service and SCLMDT)	<userid>x	OMVS
(z/OS Explorer) TSO Commands service (APPC)	FEKFRSRV	ASCH
CARMA (batch)	CRA<port>	JES
CARMA (crastart)	<userid>x	OMVS
CARMA (ISPF Client Gateway)	<userid> and <userid>x	OMVS
MVS build (batch job)	*	JES
z/OS UNIX build (shell commands)	<userid>x	OMVS
z/OS UNIX shell	<userid>	OMVS

Considerations for goal selection

The following general WLM considerations can help you to properly define the correct goal definitions for Developer for z Systems:

- You should base goals on what can actually be achieved, not what you want to happen. If you set goals higher than necessary, WLM moves resources from lower importance work to higher importance work which might not actually need the resources.
- Limit the amount of work assigned to the SYSTEM and SYSSTC service classes, because these classes have a higher dispatching priority than any WLM managed class. Use these classes for work that is of high importance but uses little CPU.
- Work that falls through the classification rules ends up in the SYSOTHER class, which has a discretionary goal. A discretionary goal tells WLM to just do the best it can when the system has spare resources.

When using response time goals:

- There must be a steady arrival rate of tasks (at least 10 tasks in 20 minutes) for WLM to properly manage a response time goal.
- Use average response time goals only for well controlled workloads, because a single long transaction has a big impact on the average response time and can make WLM overreact.

When using velocity goals:

- You usually cannot achieve a velocity goal greater than 90% for various reasons. For example, all the SYSTEM and SYSSTC address spaces have a higher dispatching priority than any velocity-type goal.
- WLM uses a minimum number of (using and delay) samples on which to base its velocity goal decisions. So the less work running in a service class, the longer it will take to collect the required number of samples and adjust the dispatching policy.
- Reevaluate velocity goals when you change your hardware. In particular, moving to fewer, faster processors requires changes to velocity goals.

STC

All Developer for z Systems started tasks are servicing real-time client requests.

Table 8. WLM workloads - STC

Description	Task name	Workload
Debug Manager	DBGMGR	STC

- Debug Manager

Debug Manager provides services to connect programs being debugged to clients debugging them. You should specify a high-performance, one-period velocity goal, because the task does not report individual transactions to WLM. Resource usage depends heavily on user actions, and will therefore fluctuate, but is expected to be minimal.

OMVS

All workloads use the client user ID as base for the address space name. (z/OS UNIX will substitute "x" in the "Task Name" column by a random 1-digit number.)

The workloads will all end up in the same service class due to a common address space naming convention. You should specify a multi-period goal for this service class. The first periods should be high-performance, percentile response time goals, while the last period should have a moderate-performance velocity goal. Some workloads, such as the ISPF Client Gateway, will report individual transactions to WLM, while others do not.

Table 9. WLM workloads - OMVS

Description	Task name	Workload
Legacy ISPF Gateway (TSO Commands service and SCLMDT)	<userid>x	OMVS
CARMA (crastart)	<userid>x	OMVS
CARMA (ISPF Client Gateway)	<userid> and <userid>x	OMVS
z/OS UNIX build (shell commands)	<userid>x	OMVS
z/OS UNIX shell	<userid>	OMVS

- Legacy ISPF Gateway

The Legacy ISPF Gateway is an ISPF service invoked by Developer for z Systems to execute non-interactive TSO and ISPF commands. This includes explicit commands issued by the client as well as implicit commands issued by the SCLMDT component of Developer for z Systems. Resource usage depends heavily on user actions, and will therefore fluctuate, but is expected to be minimal.

- CARMA

CARMA is an optional Developer for z Systems server that is used to interact with host based Software Configuration Managers (SCMs), such as CA Endeavor[®] SCM. Developer for z Systems allows for different startup methods for a CARMA server, some of which become an OMVS workload. Resource usage depends heavily on user actions, and will therefore fluctuate, but is expected to be minimal.

- z/OS UNIX build

When a client initiates a build for a z/OS UNIX project, z/OS UNIX REXEC (or SSH) will start a task that executes a number of z/OS UNIX shell commands to perform the build. Resource usage depends heavily on user actions, and will therefore fluctuate, but is expected to be moderate to substantial, depending on the size of the project.

- z/OS UNIX shell

This workload processes z/OS UNIX shell commands that are issued by the client. Resource usage depends heavily on user actions, and will therefore fluctuate, but is expected to be minimal.

JES

JES-managed batch processes are used in various manners by Developer for z Systems. The most common usage is for MVS builds, where a job is submitted and monitored to determine when it ends. But Developer for z Systems could also start a CARMA server in batch, and communicate with it using TCP/IP.

Table 10. WLM workloads - JES

Description	Task name	Workload
CARMA (batch)	CRA<port>	JES
MVS build (batch job)	*	JES

- CARMA

CARMA is a Developer for z Systems server that is used to interact with host based Software Configuration Managers (SCMs), such as CA Endeavor[®] SCM. Developer for z Systems allows for different startup methods for a CARMA server, some of which become a JES workload. You should specify a high-performance, one-period velocity goal, because the task does not report individual transactions to WLM. Resource usage depends heavily on user actions, and will therefore fluctuate, but is expected to be minimal.

- MVS build

When a client initiates a build for an MVS project, Developer for z Systems will start a batch job to perform the build. Resource usage depends heavily on user actions, and will therefore fluctuate, but is expected to be moderate to substantial, depending on the size of the project. Different moderate-performance goal strategies can be advisable, depending on your local circumstances.

- You could specify a multi-period goal with a percentile response time period and a trailing velocity period. In this case, your developers should be using mostly the same build procedure and similar sized input files to create jobs with uniform response times. There must also be a steady arrival rate of jobs (at least 10 jobs in 20 minutes) for WLM to properly manage a response time goal.
- A velocity goal is best suited for most batch-jobs, because this goal can handle highly variable execution times and arrival rates.

Chapter 5. Push-to-client considerations

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

- Client configuration files
- Client product version
- Project definitions

The following topics are covered in this chapter:

- “Introduction”
- “Host-based projects” on page 34

Developer for z Systems builds on top of IBM Explorer for z/OS . For z/OS Explorer related information, see “Push-to-client considerations” in the *IBM Explorer for z/OS Host Configuration Reference (SC27-8438)*.

Introduction

Developer for z Systems clients can pull client configuration files and product update information from the host when they connect, ensuring that all clients have common settings and that they are up-to-date.

The client administrator can create multiple client configuration sets and multiple client update scenarios to fit the needs of different developer groups. This allows users to receive a customized setup, based on criteria like membership of an LDAP group or permit to a security profile.

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

A development project manager defines a project and assigns individual developers to it.

See the Developer for z Systems IBM Knowledge Center (http://www-01.ibm.com/support/knowledgecenter/SSQ2R2/rdz_welcome.html) for details about how the development project manager can perform the tasks assigned to them.

When enabling configuration or version control support for multiple developer groups, one additional team will be involved in managing push-to-client. Which team this is depends on the option chosen to identify the groups a developer belongs to:

- An LDAP administrator maintains group definitions that place each developer in none, one, or more FEL.PTC.* LDAP groups.
- A security administrator maintains access lists to FEL.PTC.* security profiles. A developer can be authorized to none, one, or more profiles.

Host-based projects

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

The base directory for host-based projects is defined (by the client administrator) in `/var/rdz/pushtoclient/keymapping.xml`, and is `/var/rdz/pushtoclient/projects` by default.

To configure host-based projects, the project manager or lead developer needs to define the following types of configuration files. All files are UTF-8 encoded XML files.

- Project instance files are specific to a single user ID and point to reusable project definition files. Each user who works with host-based projects needs a subdirectory, `/var/zexpl/pushtoclient/projects/<userid>/`, containing one project instance file (`*.hbpin`) for each project to be downloaded.
- Project definition files define the structure and contents of the project and can be reused by multiple users. Project definition files (`*.hbppd`) list the subprojects contained by the project and are located in the root project definition directory or one of its subdirectories.
- Subproject definition files define the structure and contents of the subproject and can be reused by multiple users. Subproject definition files (`*.hbpsd`) define the set of resources required to build a single load module and are located in the root project definition directory or one of its subdirectories.
- Subproject properties files are properties files with variable substitution support and can be reused by multiple subprojects. Subproject property files (`*.hbppr`) support variable substitution to allow sharing of property files among multiple users and are located in the root project definition directory or one of its subdirectories.

| Host-based projects are also eligible to participate in the multiple group setup .
| This eligibility means that host-based projects can be defined also in
| `/var/rdz/pushtoclient/grouping/<devgroup>/projects/`.

When a workspace is bound to a specific group, and there are project definitions for a user in this group and in the default group, the user receives the project definitions from both the default and the specific group.

Chapter 6. CICSTS considerations

| This chapter groups references to Developer for z Systems components that can
| work inside CICSTS regions.

Bidirectional language support

| For more information on Bidirectional language support, see the section "CICS
| bidirectional language support" in chapter "Other customization tasks" of the
| *Rational Developer for z Systems Host Configuration Guide (SC27-8577)*.

Diagnostic IRZ messages for Enterprise Service Tools

| For more information on diagnostic IRZ messages for Enterprise Service Tools, see
| the section "Diagnostic IRZ messages for Enterprise Service Tools" in chapter
| "Other customization tasks" of the *Rational Developer for z Systems Host Configuration*
| *Guide (SC27-8577)*.

CICS transaction debugging

| For more information on CICS transaction debugging, see the section "Integrated
| Debugger CICS updates" in chapter "(Optional) Integrated Debugger" of the *IBM*
| *Rational Developer for z Systems Host Configuration Guide (SC23-7658)*.

Chapter 7. Setting up AT-TLS

This section is provided to assist you with some common problems that you may encounter when setting up Application Transparent Transport Layer Security (AT-TLS), or during checking or modifying an existing setup.

The Transport Layer Security (TLS) protocol defined in RFC 2246 provides communications privacy over the Internet. Similar to its predecessor Secure Socket Layer (SSL), the protocol enables client and server applications to communicate in a way that is designed to prevent eavesdropping, tampering, and message forgery. Application Transparent Transport Layer Security (AT-TLS) consolidates TLS implementation for z/OS-based applications in one location, allowing all applications to support TLS-based encryption without knowledge of the TLS protocol. See *Communications Server IP Configuration Guide* (SC31-8775) for more information on AT-TLS.

The Integrated Debugger in Developer for z Systems relies on AT-TLS for encrypted communication with the client, because the data for the debug session does not flow through the same pipe as other Developer for z Systems client-host communication.

The actions needed to set up AT-TLS will vary from site to site, depending on the exact needs, and depending on what is already available at the site.

The information in this section shows how to set up the TCP/IP Policy Agent that manages AT-TLS and define a policy for usage by Developer for z Systems Integrated Debugger on a z/OS 1.13 system, with support for TLS v1.2.

1. "Setting up syslogd" on page 38
2. "AT-TLS configuration in PROFILE.TCPIP" on page 38
3. "Policy Agent started task" on page 38
4. "Policy Agent configuration" on page 39
5. "AT-TLS policy" on page 39
6. "AT-TLS security updates" on page 41
7. "AT-TLS policy activation" on page 44

Throughout this section, a uniform naming convention is used:

- Debug Manager port for external communication: 5335
- Debug Manager user ID: stcdbm
- Policy agent user ID: pagent
- Certificate: dbgmgr
- Key and certificate storage: dbgmgr.racf

Some tasks described in the following sections expect you to be active in z/OS UNIX. This can be done by issuing the TSO command **OMVS**. Use the **oedit** command to edit files in z/OS UNIX. Use the **exit** command to return to TSO.

Setting up syslogd

The TCP/IP documentation recommends writing Policy Agent messages to the z/OS UNIX syslog instead of using the default log file. AT-TLS will always write messages to the z/OS UNIX syslog.

In order to do so, the z/OS UNIX syslog daemon, `syslogd`, must be configured and active. You will also need a mechanism to control the size of the log files created by `syslogd`.

The following sample configuration file updates can be used to configure and start `syslogd`, with a simple log file management mechanism (erase existing logs when z/OS UNIX starts and create new ones upon `syslogd` startup).

- `/etc/services`

```
syslog          514/udp
```
- `/etc/syslog.conf`

```
# /etc/syslog.conf - control output of syslogd
# 1. all files with will be printed to /tmp/syslog.auth.log
auth.*          /tmp/syslog.auth.log
# 2. all error messages printed to /tmp/syslog.error.log
*.err           /tmp/syslog.error.log
# 3. all debug and above messages printed to /tmp/syslog.debug.log
*.debug         /tmp/syslog.debug.log
# The files named must exist before the syslog daemon is started,
# unless -c startup option is used
```
- `/etc/rc`

```
# Start the SYSLOGD daemon for logging
# (clean up old logs)
sed -n '/^#!/s/.* \(.*)/\1/p' /etc/syslog.conf | xargs -i rm {}
# (create new logs and add userid of message sender)
_BPX_JOBNAME='SYSLOGD' /usr/sbin/syslogd -cuf /etc/syslog.conf &
sleep 5
```

AT-TLS configuration in PROFILE.TCPIP

AT-TLS support is activated by the TTLS parameter on the TCPCONFIG statement in the PROFILE.TCPIP data set. AT-TLS is managed by the Policy Agent, which must be active to be able to enforce the AT-TLS policy. Since the Policy Agent must wait for TCP/IP to be active, the AUTOSTART statement in PROFILE.TCPIP is a good place to trigger startup of this server.

These requirements result in following changes to PROFILE.TCPIP, often named TCP/IP.TCPPARMS(TCPPROF).

```
TCPCONFIG TTLS          ; Required for AT-TLS
AUTOLOG
  PAGENT                 ; POLICY AGENT, required for AT-TLS
ENDAUTOLOG
```

Policy Agent started task

As mentioned before, AT-TLS is managed by the Policy Agent, which can be started as a started task. Use the following JCL to create `SYS1.PROCLIB(PAGENT)`, using the default configuration file and the recommended log location (`SYSLOGD`). The necessary definitions in your security software are covered later.

```

//PAGENT PROC PRM='-L SYSLOGD' * '' or '-L SYSLOGD'
/**
/** TCP/IP POLICY AGENT
/** (PARM) (envar)
/** default cfg file: /etc/pagent.conf (-C) (PAGENT_CONFIG_FILE)
/** default log file: /tmp/pagent.log (-L) (PAGENT_LOG_FILE)
/** default log size: 300,3 (3x 300KB files) (PAGENT_LOG_FILE_CONTROL)
/**
//PAGENT EXEC PGM=PAGENT,REGION=0M,TIME=NOLIMIT,
// PARM='ENVAR("TZ=EST5DST")/&PRM'
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
/**

```

Policy Agent configuration

The Policy Agent enforces TCP/IP related policies created by the TCP/IP administrator. It manages policies for AT-TLS, called TTLS, but also for other services such as IPSec. The Policy Agent uses a configuration file to know which policies must be enforced, and where they can be found. The default configuration file is `/etc/pagent.conf`, but a different location can be specified in the Policy Agent started task JCL.

```

#
# TCP/IP Policy Agent configuration information.
#
TTLSConfig /etc/pagent.ttls.conf
# Specifies the path of a TTLS policy file holding stack specific
# statements.
#
#TcpImage TCP/IP /etc/pagent.conf
# If no TcpImage statement is specified, all policies will be installed
# to the default TCP/IP stack.
#
#LogLevel 31
# The sum of the following values that represent log levels:
# LOGL_SYSERR 1
# LOGL_OBJERR 2
# LOGL_PROTERR 4
# LOGL_WARNING 8
# LOGL_EVENT 16
# LOGL_ACTION 32
# LOGL_INFO 64
# LOGL_ACNTING 128
# LOGL_TRACE 256
# Log Level 31 is the default log loglevel.
#
#Codepage IBM-1047
# Specify the EBCDIC code page to be used for reading all configuration
# files and policy definition files. IBM-1047 is the default code page.

```

This sample configuration file specifies where the Policy Agent can find the TTLS policy. It uses Policy Agent default values for other statements.

AT-TLS policy

A TTLS policy describes the desired AT-TLS rules. As defined in the Policy Agent configuration file, the TTLS policy is located in `/etc/pagent.ttls.conf`. The necessary definitions in your security software are covered later.

This example shows a fairly simple, two-rule policy that disables SSL v3 and enables TLS v1, TLS v1.1 and TLS v1.2 support for both communication paths supported by Developer for z Systems Integrated Debugger, Debug Manager and Probe-Client. As defined in the Policy Agent configuration file, the TTLS policy is located in /etc/pagent.ttls.conf.

```
##
## TCP/IP Policy Agent AT-TLS configuration information.
##
##-----
TTLRule                RDz_Debug_Manager
{
  LocalPortRange       5335
  Direction             Inbound
  TTLSGroupActionRef   grp_Production
  TTLSEnvironmentActionRef act_RDz_Debug_Manager
}
##-----
TTLSEnvironmentAction  act_RDz_Debug_Manager
{
  HandshakeRole Server
  TTLSKeyRingParms
  {
    Keyring dbgmgr.racf # Keyring must be owned by the Debug Manager
  }
  TTLSEnvironmentAdvancedParms
  {
    ## TLSV1.2 only for z/OS 2.1 and higher
    # TLSV1.2 On          # TLSv1 & TLSv1.1 are on by default
    # SSLV3 Off          # disable SSLv3 }
  }
}
##-----
TTLRule                RDz_Debug_Probe-Client
{
  RemotePortRange      8001
  Direction             Outbound
  TTLSGroupActionRef   grp_Production
  TTLSEnvironmentActionRef act_RDz_Debug_Probe-Client
}
##-----
TTLSEnvironmentAction  act_RDz_Debug_Probe-Client
{
  HandshakeRole        Client
  TTLSKeyRingParms
  {
    Keyring *AUTH/*    # virtual key ring holding CA certificates
  }
  TTLSEnvironmentAdvancedParms
  {
    ## TLSV1.2 only for z/OS 2.1 and higher
    # TLSV1.2 On          # TLSv1 & TLSv1.1 are on by default
  }
}
##-----
TTLSGroupAction        grp_Production
{
  TTLSEnabled           On
  ## TLSv1.2zOS1.13 only for z/OS 1.13
  TTLSGroupAdvancedParmsRef TLSv1.2zOS1.13
  Trace                 3 # Log Errors to syslogd & IP joblog
  #Trace                254 # Log everything to syslogd
}
##-----
TTLSGroupAdvancedParms TLSv1.2zOS1.13
{
  Envfile /etc/pagent.ttls.TLS1.2zOS1.13.env
}
```

A TTLS policy allows for a wide range of filters to specify when a rule applies.

The Debug Manager is a server that listens on port 5335 for incoming connections from the Debug Engine. This information is captured in the RDz_Debug_Manager rule.

Since encrypted communication requires the usage of a server certificate, you specify that the Policy Manager must use the certificates on the dbgmgr.racf key ring, which is owned by the Debug Manager started task user ID. By default, TLS v1.2 support is disabled, so this policy explicitly enables it. SSLv3.0 is explicitly disabled due to known vulnerabilities in this protocol.

When the Debug Probe is started with Language Environment (LE) option TEST(,, ,TCP/IP&&ipaddress%8001:*), it is instructed to not use the Debug Manager but contact the Developer for z Systems client directly at port 8001. This implies, from a TCP/IP perspective, that the host-based Debug Probe is a client contacting a server (the Debug UI) in the Developer for z Systems client. This information is captured in the RDz_Debug_Probe-Client rule.

With the host being a TCP/IP client, the Policy Manager will need a way to validate the server certificate presented by the Debug UI. Instead of using a uniformly named key ring for all users that might require an encrypted debug session, we are using RACF's CERTAUTH virtual key ring (*AUTH*/*). This virtual key ring holds the public certificates of Certificate Authorities (CAs), and can be used if the Debug UI presents a server certificate that is signed by one of the trusted CAs.

Note that for more complex policies, you should use the IBM Configuration Assistant for z/OS Communications Server. This is a GUI-based tool that provides a guided interface for configuring TCP/IP policy-based networking functions and is available as a task in IBM z/OS Management Facility (z/OSMF), and as a stand-alone workstation application.

TLS v1.2 considerations

TLS v1.2 support became available in z/OS 2.1, and is disabled by default. This policy shows the command (TLsv1.2 0n) to explicitly enable it, but has it commented out as the target system is using z/OS 1.13.

By applying the following two APARs, TLS v1.2 support is added to z/OS 1.13:

- System SSL APAR OA39422
- Communications Server (AT-TLS) APAR PM62905

z/OS 1.13 System SSL, which is used by AT-TLS to implement TLS encrypted communication, requires some additional parameters for TLS v1.2 support. These are supplied through the AT-TLS policy using a file with System SSL environment variables, /etc/pagent.ttls.TLS1.2zOS1.13.env.

```
#
# Add TLsv1.2 support to AT-TLS
# requires z/OS 1.13 with OA39422 and PM62905
#
GSK_RENEGOTIATION=ALL
GSK_PROTOCOL_TLsv1_2=0N
```

AT-TLS security updates

There are several updates required to your security setup for AT-TLS to work properly. This section has sample RACF commands to do the required setup.

As mentioned in “Policy Agent started task” on page 38, you use a started task to run the Policy Agent. This requires the definition of a started task user ID and a profile in the STARTED class.

```
# define started task user ID
# BPX.DAEMON permit is required for non-zero UID
ADDUSER PAGENT DFLTGRP(SYS1) OMVS(UID(0) SHARED HOME('/')) +
  NAME('TCP/IP POLICY AGENT') NOPASSWORD

# define started task
RDEFINE STARTED PAGENT.* STDATA(USER(PAGENT) GROUP(SYS1)) +
  DATA('TCP/IP POLICY AGENT')

# refresh to make the changes visible
SETROPTS RACLIST(STARTED) REFRESH
```

Define a profile named MVS.SERVGR.PAGENT in the OPERCMDS class and give user ID PAGENT CONTROL access to it. The profile restricts who can start the Policy Agent. If the profile is not defined, and access to it is prevented through a generic profile, PAGENT will not be able to start the Policy Agent, which will prevent TCP/IP stack initialization.

```
# restrict startup of policy agent
RDEFINE OPERCMDS MVS.SERVGR.PAGENT UACC(NONE) +
  DATA('restrict startup of policy agent')
PERMIT MVS.SERVGR.PAGENT CLASS(OPERCMDS) ACCESS(CONTROL) ID(PAGENT)

# refresh to make the changes visible
SETROPTS RACLIST(OPERCMDS) REFRESH
```

As mentioned in “AT-TLS configuration in PROFILE.TCPIP” on page 38, the Policy Agent is started after TCP/IP is initialized. This means there is a (small) window where applications can use the TCP/IP stack without the TTLS policy being enforced. Define the EZB.INITSTACK.** profile in the SERVAUTH class to prevent access to the stack during this time window, except for applications with READ access to the profile. You must permit a limited set of administrative applications to the profile to ensure full initialization of the stack, as documented in “TCP/IP stack initialization access control” in *Communications Server IP Configuration Guide (SC31-8775)*.

I **Note:** The Policy Agent issues message EZD1586I when all policies are active.

```
# block stack access between stack and AT-TLS availability
# SETROPTS GENERIC(SERVAUTH)
# SETROPTS CLASSACT(SERVAUTH) RACLIST(FACILITY)
RDEFINE SERVAUTH EZB.INITSTACK.** UACC(NONE)
# Policy Agent
PERMIT EZB.INITSTACK.** CLASS(SERVAUTH) ACCESS(READ) ID(PAGENT)
# OMPROUTE daemon
PERMIT EZB.INITSTACK.** CLASS(SERVAUTH) ACCESS(READ) ID(OMPROUTE)
# SNMP agent and subagents
PERMIT EZB.INITSTACK.** CLASS(SERVAUTH) ACCESS(READ) ID(OSNMPD)
PERMIT EZB.INITSTACK.** CLASS(SERVAUTH) ACCESS(READ) ID(IOBSNMP)
# NAME daemon
PERMIT EZB.INITSTACK.** CLASS(SERVAUTH) ACCESS(READ) ID(NAMED)

# refresh to make the changes visible
SETROPTS RACLIST(SERVAUTH) REFRESH
```

(Optional) The z/OS UNIX **pasearch** command displays active policy definitions. Define profile EZB.PAGENT.** in the SERVAUTH class to restrict access to the **pasearch** command.

```

# restrict access to pasearch command
# RDEFINE SERVAUTH EZB.PAGENT.** UACC(NONE) +
# DATA('restrict access to pasearch command')
# PERMIT EZB.PAGENT.** CLASS(SERVAUTH) ACCESS(READ) ID(tcadmin)

# refresh to make the changes visible
# SETROPTS RACLIST(SERVAUTH) REFRESH

```

As mentioned in “AT-TLS policy” on page 39, Debug Manger needs a certificate so that AT-TLS can set up encrypted communication on Debug Manager’s behalf. These sample commands create a new certificate labeled dbgmgr, which is stored in a RACF key ring named dbgmgr.racf. Both the certificate and the key ring are owned by STCDBM, the Debug Manager started task user ID.

```

# permit Debug Manager to access certificates
#RDEFINE FACILITY IRR.DIGTCERT.LIST UACC(NONE)
#RDEFINE FACILITY IRR.DIGTCERT.LISTRING UACC(NONE)
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ACCESS(READ) ID(stcdbm)
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ACCESS(READ) ID(stcdbm)

# refresh to make the changes visible
SETROPTS RACLIST(FACILITY) REFRESH

# create self-signed certificate
RACDCERT ID(stcdbm) GENCERT SUBJECTSDN(CN('RDz Debug Manager') +
OU('RTP labs') O('IBM') L('Raleigh') SP('NC') C('US')) +
NOTAFTER(DATE(2015-12-31)) KEYUSAGE(HANDSHAKE) WITHLABEL('dbgmgr')

# (optional) additional steps required to use a signed certificate
# 1. create a signing request for the self-signed certificate
RACDCERT ID(stcdbm) GENREQ (LABEL('dbgmgr')) DSN(dsn)
# 2. send the signing request to your CA of choice
# 3. check if the CA credentials (also a certificate) are already known
RACDCERT CERTAUTH LIST
# 4. mark the CA certificate as trusted
RACDCERT CERTAUTH ALTER(LABEL('CA cert')) TRUST
# or add the CA certificate to the database
RACDCERT CERTAUTH ADD(dsn) WITHLABEL('CA cert') TRUST
# 5. add the signed certificate to the database;
# this will replace the self-signed one
RACDCERT ID(stcdbm) ADD(dsn) WITHLABEL('dbgmgr') TRUST
# Do NOT delete the self-signed certificate before replacing it.
# If you do, you lose the private key that goes with the certificate,
# which makes the certificate useless.

# create key ring
RACDCERT ID(stcdbm) ADDRING(dbgmgr.racf)

# add certificate to key ring
RACDCERT ID(stcbm) CONNECT(LABEL('dbgmgr') +
RING(dbgmgr.racf) USAGE(PERSONAL) DEFAULT)

# additional step required to use a signed certificate
# 6. add CA certificate to key ring
RACDCERT ID(stcdbm) CONNECT(CERTAUTH LABEL('CA cert') +
RING(dbgmgr.racf))

# refresh to make the changes visible
SETROPTS RACLIST(DIGTCERT) REFRESH

```

AT-TLS policy also documents the use of the CERTAUTH virtual key ring for validation of the server certificate presented by the Debug UI in the Probe-Client scenario. This implies that the CA certificate used by the Debug UI is trusted by your z/OS host.

```

# check if the CA credentials (also a certificate) are already known
RACDCERT CERTAUTH LIST
# mark the CA certificate as trusted
RACDCERT CERTAUTH ALTER(LABEL('CA cert')) TRUST
# or add the CA certificate to the database
RACDCERT CERTAUTH ADD(dsn) WITHLABEL('CA cert') TRUST

# refresh to make the changes visible
SETROPTS RACLIST(DIGTCERT) REFRESH

```

Use the following commands to verify your setup:

```

# verify started task setup
LISTGRP SYS1 OMVS
LISTUSER PAGENT OMVS
RLIST STARTED PAGENT.* ALL STDATA

# verify Policy Agent startup permission
RLIST OPERCMDS MVS.SERVMGR.PAGENT ALL

# verify initstack protection
RLIST SERVAUTH EZB.INITSTACK.** ALL

# verify pasearch protection
RLIST SERVAUTH EZB.PAGENT.** ALL

# verify certificate setup
RACDCERT CERTAUTH LIST(LABEL('CA cert'))
RACDCERT ID(stcdbm) LIST(LABEL('dbgmgr'))
RACDCERT ID(stcdbm) LISTRING(dbgmgr.racf)

```

AT-TLS policy activation

AT-TLS setup is now complete, and the policy will be activated at next IPL of the system. Follow these steps to start using the policy without an IPL:

1. Activate AT-TLS support in the TCP/IP stack.
 Create a TCP/IP obey file, for example, TCPIP.TCPPARMS(OBEY), with the following content:

```
TCPCONFIG TTLS
```

 Activate it with this operator command:

```
V TCPIP,,OBEY,TCPIP.TCPPARMS(OBEY)
```

 Verify the result by checking for this console message:

```
EZZ4249I stackname INSTALLED TTLS POLICY HAS NO RULES
```
2. Start the Policy Agent.
 Issue operator command:

```
S PAGENT
```

 Verify the result by checking for console message:

```
EZD1586I PAGENT HAS INSTALLED ALL LOCAL POLICIES FOR stackname
```
3. Restart Debug Manager to interrupt all active, non-encrypted, sessions.
 Issue operator commands:

```
P DBGMGR
S DBBMGR
```

Bibliography

Referenced publications

The following publications are referenced in this document:

Table 11. Referenced publications

Publication title	Order number	Reference	Reference Web site
Program Directory for IBM Rational Developer for z Systems	GI11-8298	Developer for z Systems	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Program Directory for IBM Rational Developer for z Systems Host Utilities	GI13-2864	Developer for z Systems	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
IBM Rational Developer for z Systems Host Configuration Guide	SC27-8577	Developer for z Systems	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
IBM Rational Developer for z Systems Host Configuration Reference	SC27-8578	Developer for z Systems	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
IBM Rational Developer for z Systems Common Access Repository Manager Developer's Guide	SC23-7660	Developer for z Systems	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
SCLM Developer Toolkit Administrator's Guide	SC23-9801	Developer for z Systems	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
IBM Explorer for z/OS Host Configuration Guide	SC27-8437	z/OS Explorer	
IBM Explorer for z/OS Host Configuration Reference	SC27-8438	z/OS Explorer	
Communications Server IP CICS Sockets Guide	SC31-8807	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Communications Server IP Configuration Guide	SC31-8775	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Communications Server IP Configuration Reference	SC31-8776	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS Initialization and Tuning Guide	SA22-7591	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS Initialization and Tuning Reference	SA22-7592	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS JCL Reference	SA22-7597	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS Planning Workload Management	SA22-7602	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS System Commands	SA22-7627	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/

Table 11. Referenced publications (continued)

Publication title	Order number	Reference	Reference Web site
Security Server RACF Command Language Reference	SA22-7687	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Security Server RACF Security Administrator's Guide	SA22-7683	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
UNIX System Services Command Reference	SA22-7802	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
UNIX System Services Planning	GA22-7800	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
UNIX System Services User's Guide	SA22-7801	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Using REXX and z/OS UNIX System Services	SA22-7806	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/

The following Web sites are referenced in this document:

Table 12. Referenced Web sites

Description	Reference Web site
Developer for z Systems IBM Knowledge Center	http://www-01.ibm.com/support/knowledgecenter/SSQ2R2/rdz_welcome.html
Developer for z Systems Library	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Developer for z Systems home page	http://www-03.ibm.com/software/products/en/developperforsystemz/
Developer for z Systems Recommended service	http://www-01.ibm.com/support/docview.wss?rs=2294&context=SS2QJ2&uid=swg27006335
Developer for z Systems enhancement request	https://www.ibm.com/developerworks/support/rational/rfe/
Download Apache Ant	http://ant.apache.org/

Informational publications

The following publications can be helpful in understanding setup issues for the requisite host system components:

Table 13. Informational publications

Publication title	Order number	Reference	Reference website
ABCs of z/OS System Programming Volume 9 (z/OS UNIX)	SG24-6989	Redbook	http://www.redbooks.ibm.com/
System Programmer's Guide to: Workload Manager	SG24-6472	Redbook	http://www.redbooks.ibm.com/
TCPIP Implementation Volume 1: Base Functions, Connectivity, and Routing	SG24-7532	Redbook	http://www.redbooks.ibm.com/

Table 13. Informational publications (continued)

Publication title	Order number	Reference	Reference website
TCPIP Implementation Volume 3: High Availability, Scalability, and Performance	SG24-7534	Redbook	http://www.redbooks.ibm.com/
TCP/IP Implementation Volume 4: Security and Policy-Based Networking	SG24-7535	Redbook	http://www.redbooks.ibm.com/
Tivoli® Directory Server for z/OS	SG24-7849	Redbook	http://www.redbooks.ibm.com/

Glossary

Action ID

A numeric identifier for an action between 0 and 999

Application Server

1. A program that handles all application operations between browser-based computers and an organization's back-end business applications or databases. There is a special class of Java-based appservers that conform to the Java EE standard. Java EE code can be easily ported between these appservers. They can support JSPs and servlets for dynamic Web content and EJBs for transactions and database access.
2. The target of a request from a remote application. In the DB2[®] environment, the application server function is provided by the distributed data facility and is used to access DB2 data from remote applications.
3. A server program in a distributed network that provides the execution environment for an application program.
4. The target of a request from an application requester. The database management system (DBMS) at the application server site provides the requested data.
5. Software that handles communication with the client requesting an asset and queries of the Content Manager.

Bidirectional (bi-di)

Pertaining to scripts such as Arabic and Hebrew that generally run from right to left, except for numbers, which run from left to right. This definition is from the Localization Industry Standards Association (LISA) Glossary.

Bidirectional Attribute

Text type, text orientation, numeric swapping, and symmetric swapping.

Build Request

A request from the client to perform a build transaction.

Build Transaction

A job started on MVS to perform builds after a build request has been received from the client.

Compile

1. In Integrated Language Environment (ILE) languages, to translate source statements into modules that then can be bound into programs or service programs.
2. To translate all or part of a program expressed in a high-level language into a computer program expressed in an intermediate language, an assembly language, or a machine language.

Container

1. In CoOperative Development Environment/400, a system object that contains and organizes source files. An i5/OS[™] library or an MVS-partitioned data set are examples of a container.
2. In Java EE, an entity that provides life-cycle management, security, deployment, and runtime services to components. (Sun) Each type of container (EJB, Web, JSP, servlet, applet, and application client) also provides component-specific services
3. In Backup Recovery and Media Services, the physical object used to store and move media such as a box, a case, or a rack.
4. In a virtual tape server (VTS), a receptacle in which one or more exported logical volumes (LVOLs) can be stored. A stacked volume containing one or more LVOLs and residing outside a VTS library is considered to be the container for those volumes.
5. A physical storage location of the data. For example, a file, directory, or device.

6. A column or row that is used to arrange the layout of a portlet or other container on a page.
7. An element of the user interface that holds objects. In the folder manager, an object that can contain other folders or documents.

Database

A collection of interrelated or independent data items that are stored together to serve one or more applications.

Data Definition View

Contains a local representation of databases and their objects and provides features to manipulate these objects and export them to a remote database

Data Set

The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

Debug

To detect, diagnose, and eliminate errors in programs.

Debugging Session

The debugging activities that occur between the time that a developer starts a debugger and the time that the developer exits from it.

Error Buffer

A portion of storage used to hold error output information temporarily.

Gateway

1. A middleware component that bridges Internet and intranet environments during Web service invocations.
2. Software that provides services between the endpoints and the rest of the Tivoli environment.

3. A component of a Voice over Internet Protocol that provides a bridge between VoIP and circuit-switched environments.
4. A device or program used to connect networks or systems with different network architectures. The systems may have different characteristics, such as different communication protocols, different network architecture, or different security policies, in which case the gateway performs a translation role as well as a connection role.

Interactive System Productivity Facility (ISPF)

An IBM licensed program that serves as a full-screen editor and dialog manager. Used for writing application programs, it provides a means of generating standard screen panels and interactive dialogs between the application programmer and terminal user. ISPF consists of four major components: DM, PDF, SCLM, and C/S. The DM component is the Dialog Manager, which provides services to dialogs and end-users. The PDF component is the Program Development Facility, which provides services to assist the dialog or application developer. The SCLM component is the Software Configuration Library Manager, which provides services to application developers to manage their application development libraries. The C/S component is the Client/Server, which allows you to run ISPF on programmable workstation, to display the panels using the display function of your workstation operating system, and to integrate workstation tools and data with host tools and data.

Interpreter

A program that translates and runs each instruction of a high-level programming language before it translates and runs the next instruction.

Isomorphic

Each composed element (in other words, an element containing other elements) of the XML instance document starting from

the root has one and only one corresponding COBOL group item whose nesting depth is identical to the nesting depth of its XML equivalent. Each non-composed element (in other words, an element that does not contain other elements) in the XML instance document starting from the top has one and only one corresponding COBOL elementary item whose nesting depth is identical to the nesting level of its XML equivalent and whose memory address at runtime can be uniquely identified.

Linkage Section

The section in the data division of an activated unit (a called program or an invoked method) that describes data items available from the activating unit (a program or a method). These data items can be referred to by both the activated unit and the activating unit.

Load Library

A library containing load modules.

Lock Action

Locks a member.

Navigator View

Provides a hierarchical view of the resources in the Workbench.

Non-Isomorphic

A simple mapping of COBOL items and XML elements belonging to XML documents and COBOL groups that are not identical in shape (non-isomorphic). Non-isomorphic mapping can also be created between non-isomorphic elements of isomorphic structures.

Output Console View

Displays the output of a process and allows you to provide keyboard input to a process.

Output View

Displays messages, parameters, and results that are related to the objects that you work with

Perspective

A group of views that show various aspects of the resources in the workbench. The workbench user can switch perspectives, depending on the task at hand, and customize the layout of views and editors within the perspective.

RAM Repository Access Manager

Remote File System

A file system residing on a separate server or operating system.

Remote System

Any other system in the network with which your system can communicate.

Remote Systems Perspective

Provides an interface for managing remote systems using conventions that are similar to ISPF.

Repository

1. A storage area for data. Every repository has a name and an associated business item type. By default, the name will be the same as the name of the business item. For example, a repository for invoices will be called Invoices. There are two types of information repositories: local (specific to the process) and global (reusable).
2. A VSAM data set on which the states of BTS processes are stored. When a process is not executing under the control of BTS, its state (and the states of its constituent activities) are preserved by being written to a repository data set. The states of all processes of a particular process-type (and of their activity instances) are

stored on the same repository data set. Records for multiple process-types can be written to the same repository.

3. A persistent storage area for source code and other application resources. In a team programming environment, a shared repository enables multiuser access to application resources.
4. A collection of information about the queue managers that are members of a cluster. This information includes queue manager names, their locations, their channels, what queues they host, and so on.

Repository Instance

A project or component that exists in an SCM.

Repositories View

Displays the CVS repository locations that have been added to your Workbench.

Response File

1. A file that contains a set of predefined answers to questions asked by a program and that is used instead of entering those values one at a time.
2. An ASCII file that can be customized with the setup and configuration data that automates an installation. The setup and configuration data would have to be entered during an interactive install, but with a response file, the installation can proceed without any intervention.

Servers View

Displays a list of all your servers and the configurations that are associated with them.

Shell A software interface between users and the operating system that interprets commands and user interactions and communicates them to the operating system. A computer may have several layers of shells for various levels of user interaction.

Shell Name

The name of the shell interface.

Shell Script

A file containing commands that can be interpreted by the shell. The user types the name of the script file at the shell

command prompt to make the shell execute the script commands.

Siddeck

A library that publishes the functions of a DLL program. The entry names and module names are stored in the library after the source code is compiled.

Silent Installation

An installation that does not send messages to the console but instead stores messages and errors in log files. Also, a silent installation can use response files for data input.

Silent Uninstallation

An uninstallation process that does not send messages to the console but instead stores messages and errors in log files after the uninstall command has been invoked.

Task List

A list of procedures that can be executed by a single flow of control.

URL Uniform Resource Locator

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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

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

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

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

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

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those

websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

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

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

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

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

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

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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

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

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

COPYRIGHT LICENSE:

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

programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Programming interface information

Trademarks

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

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

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

Personal use

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

Commercial use

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

Rights

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

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

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

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING

BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Copyright license

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

Trademark acknowledgments

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

Adobe and PostScript are trademarks of Adobe Systems Incorporated.

Cell Broadband Engine - Sony Computer Entertainment Inc.

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

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

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

ITIL is a trademark of The Minister for the Cabinet Office

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

Linux is a trademark of Linus Torvalds

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

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

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

Index

A

- access to integrated debugger, Define 20
- APF authorization
 - FEL.SFELAUTH 21
- Application Deployment Manager (ADM) 4
- application protection for RSE, Define 18
- AQEZPCM 11
- AT-TLS configuration, PROFILE.TCPIP 38
- AT-TLS policy 39
- AT-TLS policy activation 44
- AT-TLS security updates 41
- AT-TLS setup 37
- Authentication methods 11
- authentication, Debug Manager 11

B

- bidirectional language support 35

C

- CARMA and TCP/IP ports 25
- CICS transaction debug 35
- CICSTS considerations 35
- CICSTS security 13
- classification rules, WLM 28
- command security, Define JES 19
- communication, External 23
- communication, Internal 24
- component overview, Developer for z Systems
 - graphical representation 3
- Connection security 12
- considerations, Security 11
- controlled libraries for RSE, Define MVS 16

D

- data set profiles, Define 21
- Debug Manager authentication 11
- debug security 12
- debug, CICS transaction 35
- debugger, integrated 6
- Define access to integrated debugger 20
- Define MVS program controlled libraries for RSE 16
- Define PassTicket support for RSE 17
- Define RSE server as a secure z/OS UNIX 16
- Define z/OS UNIX file access permission for RSE 18
- Define z/OS UNIX program controlled files for RSE 18
- definitions, Security 13

- Developer for z Systems started tasks, Define 15
- Developer for z Systems, component overview
 - graphical representation 3
- Developer for z Systems, understanding 3
- diagnostic IRZ messages 35
- directory structure, z/OS UNIX
 - graphical representation 8

E

- encrypted communication
 - Integrated Debugger 12
- Enterprise Service Tools 35
- External communication 23

F

- FEJJCNFG 24
- FELRACF, security definitions 13

G

- goals, setting in WLM 29

H

- host-based projects 34

I

- integrated debugger 6
- Integrated Debugger
 - encrypted communication 12
- Internal communication 24
- introduction, push-to-client considerations 33
- IRZ messages 35

J

- JES command security, Define 19
- JES Job Monitor (JMON) 4
- JMON 19

L

- language support, Bidirectional 35
- libraries for RSE, Define MVS 16
- Lock Daemon (LOCKD) 4

M

- methods, Authentication 11
- MVS program controlled libraries for RSE, Define 16

O

- OMVS segment, Define 15

P

- PassTicket support for RSE, Define 17
- Policy Agent configuration 39
- Policy Agent started task 38
- port reservation, TCP/IP 24
- ports, CARMA and TCP/IP 25
- ports, TCP/IP 23
- PROFILE.TCPIP, AT-TLS
 - configuration 38
- profiles, Define data set 21
- projects, host-based 34
- publications, Referenced 45
- push-to-client considerations 33

R

- Referenced publications 45
- reservation, TCPIP port 24
- RSE, Define MVS program controlled libraries for 16
- RSE, Define PassTicket support for 17
- RSE daemon 23
- RSE daemon (RSED) 4
- RSE server 23
- RSE, define application protection for 18
- RSE, Define as a secure z/OS UNIX server 16
- RSE, Define z/OS UNIX file access permission 18
- RSE, Define z/OS UNIX program controlled files for 18

S

- SCLM Developer Toolkit 16
- SCLM Developer Toolkit (SCLMDT) 4
- SCLM security 13
- secure z/OS UNIX server, Define RSE as a 16
- Security considerations 11
- Security definitions 13
- security definitions, Checklist 14
- security settings and classes, Activate 14
- security settings, verify 22
- security, CICSTS 13
- security, Connection 12
- security, debug 12
- security, Define JES command 19
- security, SCLM 13

- segment, Define OMVS 15
- setting goals, WLM 29
- settings and classes, Activate security 14
- started task, Policy Agent 38
- started tasks, Define for Developer for z Systems
 - JMON started tasks 15
 - RSED started tasks 15
- subsystem types
 - ASCH 28
 - CICS 28
 - JES 28
 - OMVS 28
 - STC 28
- support for RSE, Define PassTicket 17
- syslogd setup 38

T

- task owners 4
- TCP/IP port reservation 24
- TCP/IP ports 23
- TCP/IP ports, graphical representation 23
- TLS v1.2 considerations 41
- TSO Command Service 4

U

- understanding Developer for z Systems 3
- UNIX program controlled files for RSE, Define 18
- UNIX server, Define RSE as 16

V

- Verify security settings 22

W

- WLM classification rules 28
- WLM considerations xii, 27
- workload classification, WLM 27
- workload manager 27

Z

- z/OS UNIX directory structure
 - graphical representation 8
- z/OS UNIX file access permission, Define for RSE 18
- z/OS UNIX program controlled files for RSE, Define 18
- z/OS UNIX server, Define RSE as 16

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

IBM Rational Developer for z Systems
Version 9.5.1
Host Configuration Reference Guide

Publication No. SC27-8578-00

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.

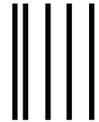
Email address



Fold and Tape

Please do not staple

Fold and Tape



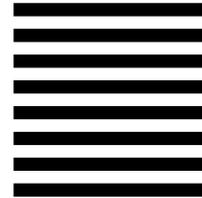
NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

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

SC27-8578-00

