Rational® Developer for System z
Version 8.0.1

IBM Rational Developer for System z Unit Test: Configuration Guide

SC14-7281-01

Rational® Developer for System z
Version 8.0.1

IBM Rational Developer for System z Unit Test: Configuration Guide

SC14-7281-01

Note

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

Note: This Program is licensed only for development and test of applications that run on IBM z/OS. The Program may not be used to run production workloads of any kind, nor more robust development workloads including without limitation production module builds, pre-production testing, stress testing, or performance testing.

Second edition (December 2010)

This edition applies to Rational Developer for System z version 8.0.1 (program number 5724-T07) and to all subsequent releases and modifications until otherwise indicated in new editions.

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

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

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

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

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.

Contents

Tables

About this book

This guide is intended to help you to set up IBM® Rational® Developer for System z® Unit Test, and customize the supplied software distribution for development and test purposes. The guide is intended to allow a person with very little z/OS® systems programming experience to configure the Unit Test feature.

Note: This Program is licensed only for development and test of applications that run on IBM z/OS. The Program may not be used to run production workloads of any kind, nor more robust development workloads including without limitation production module builds, pre-production testing, stress testing, or performance testing.

The information in this document applies to all Rational Developer for System z v8.0.1 packages including IBM Rational Developer for zEnterprise.

Who should read this book

This book is intended for programmers installing and configuring IBM Rational Developer for System z Unit Test.

Rational Developer for System z Unit Test: Configuration Guide

Introduction

This guide is intended to help you to set up IBM Rational Developer for System z Unit Test and customize the supplied software distribution for development and test purposes. The guide is intended to allow a person with very little z/OS systems programming experience to configure the Unit Test feature.

Note: This Program is licensed only for development and test of applications that run on IBM z/OS. The Program may not be used to run production workloads of any kind, nor more robust development workloads including without limitation production module builds, pre-production testing, stress testing, or performance testing.

The customized system will allow you to connect to z/OS using a Developer for System z workstation client, and will locate most of your changes in data sets and volumes that are separate from those provided by the supplied z/OS distribution. All z/OS installations use different conventions, security models, subsystems and tools, so the customized system will not be suitable for immediate deployment. By performing the tasks in this guide, you should gain familiarity with basic concepts of z/OS and the configuration of the supplied z/OS distribution in particular.

Because z/OS is a complex system with many configurable components, you will need some basic z/OS skills, and you will need to do additional customization of z/OS to meet your individual needs. This guide may be viewed as an addendum to the series of Redbooks® titled *System z Personal Development Tool*, and the Developer for System z installation and host configuration guides.

This guide will point out some of the design choices and methods that are helpful in setting up z/OS on the Unit Test feature system. This guide is not intended to replace an experienced system programmer, and some shortcuts in implementation are taken to make the process easier to perform. Any shortcuts used in this document are clearly identified as shortcuts.

The Unit Test feature is based on the System z Personal Development Tool (zPDT). Most details on the installation and customization of the Unit Test feature and the supplied z/OS distribution, which is based on the Application Developer Controlled Distribution (ADCD), are found in the Redbook series titled *System z Personal Development Tool*. Installation and customization information is found in the 2nd volume, *System z Personal Development Tool: Installation and Basic Use* (SG24-7722). This guide is based on the ADCD Summer 2010 distribution.

This guide does not replace the zPDT Redbooks and will describe a system that has been customized slightly differently than the systems described in the zPDT Redbooks. You are *strongly* encouraged to read the installation topics in the zPDT Redbooks before following this guide.

IBM Rational Developer for System z Unit Test utilizes zPDT technology. References in the zPDT Redbooks to the configuration of zPDT and 1090 systems are also applicable to Developer for System z Unit Test.

The z/OS distribution (that is provided with Unit Test only for test purposes) contains many of the products and subsystems you expect in a z/OS system. Most of these run without any customization and some need some customization or initialization to make them work.

As with all z/OS systems, the enabled subsystems are mainly defined by configuration files and procedures that are read or started when an IPL of the system is done. The supplied z/OS distribution contains several such configurations for various combinations of DB2®, CICS®, IMS™ and other subsystems. This guide concentrates on the configuration that starts DB2 and CICS version 4.1.

Much of the z/OS customization shown in this guide is done to isolate the base configuration files that come with the supplied z/OS distribution from your configuration. By placing your data on separate disk volumes and making minimal changes to the volumes supplied by the z/OS distribution, migration to a new version of the z/OS distribution should be relatively easy.

Additional information related to the supplied z/OS distribution, including disk contents and configurations, can be found in Appendix A, "z/OS distribution notes," on page 31.

Information on z/OS systems programming can be found in the series of Redbooks titled *ABCs of z/OS System Programming*, and additional useful information is contained in the Redbook titled *S/390® Partners in Development: OS/390® (and z/OS) New Users Cookbook*(SG24-6204).

Redbooks are available at `http://www.redbooks.ibm.com`.

Information configuring individual products can be found in other Redbooks and in the manuals available at `http://www.ibm.com`.

The Unit Test feature provides a platform for running z/OS on a personal computer or workstation based on Intel compatible 64 bit architecture. The z/OS software that runs on the Unit Test platform is based on a prepackaged installation called the Application Developer Controlled Distribution.

If you follow this guide, you will have a z/OS 1.11 test system with the following characteristics:

- At IPL, the system will start DB2 version 9, CICS version 4.1, and the Rational Developer for System z version 8 servers.
- You will have most of your customization files (PARMLIB, PROCLIB, and so forth) stored on disk volumes or data sets that are independent of the disks distributed with the supplied z/OS distribution.
- Your user data will be on a separate disk and future user data will be written to that disk.
- The z/OS UNIX subsystem will have new file systems mounted for the Developer for System z installation as well as for /tmp and /u.
- You will have a work volume specifically for temporary files.
- You will be able to communicate with your z/OS system via TCP/IP and have all TCP/IP settings in one data set.
- Your z/OS system will have better security than the security provided with the supplied z/OS distribution. This document will suggest some settings in RACF® to prevent users from damaging critical system files.

This guide will also introduce you to how you can create and customize new user IDs, and how you can make some minor but common changes to z/OS.

Installation of the Unit Test feature

This section of the guide is intended as an addendum to the installation instructions in the zPDT Redbooks. You can find the instructions for configuring your Linux system and installing the basic Unit Test system in the second Redbook volume, *System z Personal Development Tool: Installation and Basic Use* (SG24-7722).

Important: The instructions in the zPDT Redbook refer to both a 32 bit and a 64 bit installation program. Rational Developer for System z Unit Test feature contains only the 64 bit installation program named RDZ_z1090-1.2.41.23.x86_64. The name of the program may vary slightly.

In order to run Unit Test, you need to attach a supplied USB hardware key that has been updated to enable one or more virtual System z CPUs. This guide assumes that you have an enabled USB hardware key. The process of updating the USB hardware key using `SecureUpdateUtility` is the same as the process described in the zPDT Redbook.

Important: The method for obtaining the necessary update file for the USB hardware key is different than the method described in the Redbook. Instructions are available in the materials you receive with the Unit Test feature.

The system described in this document has a slightly different setup for the Linux machine. A separate partition was not created for the virtual disks. The system used for this guide was an OpenSUSE 11.2 system that was installed with all of the defaults. The hardware clock was set to use GMT because that is required by the USB hardware key. The Linux user ID that was used is `ibmsys1`. All of the Unit Test related files mentioned are stored within the `ibmsys1` home directory. The scripts are in `/home/ibmsys1/z`, and the virtual 3390 volumes are stored in `/home/ibmsys1/z1090/disks`. The directory structure `home/ibmsys1/z1090/disks` was used because that directory structure complies with the structure created when you start the Unit Test feature.

Defining the Unit Test machine characteristics

Create volumes USER00, SYSUT1, and PUBLIC

The system configuration in this guide is intended to isolate most of the changes you make to disk volumes that are separate from the original disk volumes shipped as part of the supplied z/OS distribution. This makes future upgrades of the z/OS system software easier to implement.

Most of the customizations in this guide are stored on two new volumes you create named **USER00** and **SYSUT1**. These names are arbitrary but are used in all of the descriptions and examples. A volume named **PUBLIC** is also created and referenced. PUBLIC is used for all temporary data sets.

- **USER00** contains user data, including z/OS UNIX file systems.
- **SYSUT1** contains the Developer for System z installation, a new RACF database, and some system customization data.
- **PUBLIC** contains temporary data sets.

Note: Most data sets that start with high-level qualifier of USER, in which most of the system changes will be made, are stored on the SBSYS1 volume which is part of the supplied z/OS distribution. The USER data sets cannot be moved to a new volume without changing references in a large number of PARMLIB members. Before you move to a new version of the supplied z/OS distribution, you should copy up all of the USER data sets to USER00, mount USER00 in the new installation, and then merge your changes into the USER libraries of the new supplied z/OS distribution installation.

To create the three disks that must be added, use the following commands in Linux:

```
mkdir –p /home/ibmsys1/z1090/disks
alcckd /home/ibmsys1/z1090/disks/USER00 –d3390-9
alcckd /home/ibmsys1/z1090/disks/SYSUT1 –d3390-3
alcckd /home/ibmsys1/z1090/disks/PUBLIC –d3390-3
```

You may want to use different device types or sizes.

Define the device map

The Unit Test feature allows the customization of the System z services available within the environment. The services can be defined in a device map or "devmap". The devmap shown here is based on the ones in the "1090 Control Files" section of the zPDT Redbook. Several additional disk definitions and a custom networking section have been added. When you decide to use a newer version of the z/OS distribution in the future, you can mount your custom disks along with the newer distribution disks and restore your user data and customizations with minimal effort.

The disks provided by the supplied z/OS distribution are mounted at the same addresses as those in the zPDT Redbooks. Some changes were made.

```
[system]
memory 4096m
processors 1
3270port 3270               # port number for non-SNA (coax) 3270

[manager]
name aws3274 0001           # define non-SNA (coax) 3270 terminals
device 0700 3279 3274 mstcon
device 0701 3279 3274 tso
```

For the network adapter definitions, the following was added. Your setup may differ, and you are encouraged to consult the sections on connectivity in the Redbooks.

```
[manager]          # define network adapter (OSA) for communication with Linux
name awsosa 0024 --path=A0 --pathtype=OSD --tunnel_intf=y   # QDIO mode
device 400 osa osa
device 401 osa osa
device 402 osa osa

[manager]          # define network adapter (OSA) for communication with network
name awsosa 22 --path=F0 --pathtype=OSD    # QDIO mode
device 404 osa osa
device 405 osa osa
device 406 osa osa
```

Since the disks are located in the /home/ibmsys1/z1090/disks/ directory, and because the three disks mentioned above have been added, the DASD definitions look like this:

```
[manager]
name awsckd 0001
device 0a80 3390 3990 /home/ibmsys1/z1090/disks/sbres1
device 0a81 3390 3990 /home/ibmsys1/z1090/disks/sbres2
device 0a82 3390 3990 /home/ibmsys1/z1090/disks/sbsys1
device 0a83 3390 3990 /home/ibmsys1/z1090/disks/sbuss1
device 0a84 3390 3990 /home/ibmsys1/z1090/disks/sbprd1
device 0a85 3390 3990 /home/ibmsys1/z1090/disks/sbprd2
device 0a86 3390 3990 /home/ibmsys1/z1090/disks/sbprd3
#device 0a87 3390 3990 /home/ibmsys1/z1090/disks/sbdis1     # if desired
#device 0a88 3390 3990 /home/ibmsys1/z1090/disks/sbdis2     # if desired
#device 0a89 3390 3990 /home/ibmsys1/z1090/disks/sbdis3     # if desired
#device 0a8a 3390 3990 /home/ibmsys1/z1090/disks/sbdis4     # if desired
#device 0a8b 3390 3990 /home/ibmsys1/z1090/disks/sbdis5     # if desired
#device 0a8c 3390 3990 /home/ibmsys1/z1090/disks/sbdis6     # if desired
device 0a8d 3390 3990 /home/ibmsys1/z1090/disks/sbdb91
device 0a8e 3390 3990 /home/ibmsys1/z1090/disks/sbdb92
device 0a8f 3390 3990 /home/ibmsys1/z1090/disks/sbdb93
device 0a90 3390 3990 /home/ibmsys1/z1090/disks/sbcic1
#device 0a91 3390 3990 /home/ibmsys1/z1090/disks/sbims    # if
# available and needed
#device 0aa0 3390 3990 /home/ibmsys1/z1090/disks/sares1
# standalone IPL
# Additional user volumes

device 0ab0 3390 3990 /home/ibmsys1/z1090/disks/PUBLIC  # PUBLIC for temp
                                                        # data sets
device 0ab1 3390 3990 /home/ibmsys1/z1090/disks/USER00  # USER00 for user data
device 0ab2 3390 3990 /home/ibmsys1/z1090/disks/SYSUT1  # SYSUT1 Developer
                                                        # for System z installs
                                                        # RACF DB
```

These additional disks listed at the end must exist before verifying the devmap with the `awsckmap` command. The `devmap` file was created in this directory:

`/home/ibmsys1/z/aprof11s`

The devmap was verified with the command:

`awsckmap /home/ibmsys1/z/aprof11s`

Starting the Unit Test environment

If you use the same directory structure used in this guide, you can start your Unit Test system with a script similar to the one below. The script example used in this document is named:

`/home/ibmsys/z/runzpdt`

After you create the script, you must ensure that it is executable by issuing the following command:

`chmod 755 /home/ibmsys/z/runzpdt`

The sample script is shown here:

```
#!/bin/sh
cd /home/ibmsys1/z  # the directory from which you want to run

echo stopping previous instance
awsstop
ps -f | grep x3270 | grep mstcon | awk '{print $2}' | xargs -n 1 kill
sleep 5
# start Unit Test.  --clean is optional
echo awsstart aprof11s --clean
awsstart aprof11s --clean
sleep 5
echo Rational Unit Test started
```

```
echo $(token)
# start x3270 for the console and one local user terminal

echo  nohup x3270 -model 4 mstcon@localhost:3270 \&
nohup x3270 -model 4 mstcon@localhost:3270 > /dev/null &

echo  nohup x3270 -model 4 tso@localhost:3270 \&
nohup x3270 -model 4 tso@localhost:3270 > /dev/null &

# IPL z/OS -- Change the parm string to the loadparm you need.
# note that the guide assumes: a80 parm 0a824k
#   To do an IPL of the standalone system use: ipl aa0 parm 0aa0sa

echo  ipl a80 parm 0a82DC
ipl a80 parm 0a82DC
```

The *ipl* statement contains three pieces of information. The a80 is the device address of the sysres volume, which is a bootable z/OS volume. The string 0A82DC indicates that the (4 digit) device address of the IODF volume (which holds IPL configuration files) is 0A82 and that the LOADxx member that will be used is LOADDC.

The member LOADDC was chosen because this one is already configured to do a cold start and start CICS 4.1 and DB2.

When you first IPL the system, you will see messages similar to:

```
IEA311I UNLABELED DASD ON 0AB0. UNIT PUT OFFLINE.
```

These messages will stop after you complete the section titled “Initialize new disks” later in this guide.

Once you get the DC configuration running, you can use DB to perform a warm start (which preserves the JES job spool).

A brief introduction to z/OS system configuration

In the simplest case, z/OS is configured by changing partitioned data set (PDS) members in the SYS1.PROCLIB, SYS1.PARMLIB, and a few other important data sets including site-specific partitioned data sets.

Most configuration (PARMLIB) member names consist of a predefined name with a two character suffix added. A common convention is to refer to the suffix as *xx*, so you will often see references to LOADxx, IEASYSxx, and so on. Configuration files refer to other members by a keyword and suffix number. For example, a member called IEASYSDC might contain a line OMVS=DB, which means that z/OS UNIX will find the member containing its configuration parameters by starting with a predetermined name, BPXPRM, and add the suffix DB (resulting in member name of BPXPRMDB). The keyword in the configuration files *does not* usually match the member name prefix.

Data sets such as SYS1.PARMLIB and SYS1.PROCLIB are usually not updated directly. Often, there are installation-specific libraries that are searched before the SYS1 data sets, leaving the SYS1 libraries with IBM supplied defaults.

The supplied z/OS distribution defines two sets of alternate libraries. The configurations for the distribution itself are stored in a set of libraries starting with the qualifiers ADCD.Z111S. The supplied z/OS distribution also provides a set of libraries for you to use, which start with the high-level qualifier USER, and which

are located first in the related data set concatenations. The supplied z/OS distribution is already set up to read from most of the USER libraries.

The USER.* libraries are initially empty but almost all changes mentioned here are made in the USER.* libraries so that the original values can be referenced and so that changes can be easily merged into a new z/OS distribution later. You should avoid updating the ADCD.* and SYS1.* libraries whenever possible.

The IPL process

LOADxx and IEASYSxx members

When z/OS is started, it looks in some predetermined locations for a member called LOADxx. The xx value is specified in the IPL statement of the startup script that was just created (DC in the example on page 5). The supplied z/OS distribution provides various LOADxx members in SYS1.IPLPARM. A LOADxx member defines various settings to start the system, such as the PARMLIB concatenation which indicates which data sets, and their order, are to be searched for other configuration members. The LOADxx member also defines which IEASYSxx member is to be used. IEASYSxx is considered the starting point for system configuration, because it contains pointers to other PARMLIB members that are used during the IPL process.

Tip: If you make a mistake which causes z/OS to not start, you can try to do an IPL of the system with the `CS` or `00` as the last two digits of the loadparm. This will start the system with a simpler configuration. `CS` does a cold start (this clears the JES spool), `00` does a warm start. For example, to use a loadparm of `CS`, change the IPL command in your startup script to `ipl 0a80 parm 0a82cs`. Note that CS and 00 share most configuration members with DC and DB, which are used in this guide, so they may also fail to start the system.

Alternatively, you can do an IPL of the standalone system which provides a very basic system, but lets you change your configuration data sets, because it does not share anything with the normal z/OS setup. When using the standalone system, you MUST specify the volume name of data sets you want to edit. Do not edit the standalone system configuration. You can do an IPL of the standalone system by replacing the IPL statement with `ipl 0aa0 parm 0aa0sa`.

PROCLIB: System procedure library

PARMLIB members only contain configuration information. The procedures that actually start the various subsystems and servers are found in a different concatenation called PROCLIB. Like the PARMLIB data sets, the supplied z/OS distribution contains SYS1.PROCLIB, ADCD.Z111S.PROCLIB and USER.PROCLIB. However, the supplied z/OS distribution does not automatically use USER.PROCLIB. That is changed in a later step by modifying the MSTJCLxx member of PARMLIB.

Configuring a working z/OS system

Note: In order to reduce complexity, several shortcuts have been taken in this document. Copies of existing PARMLIB members have been made and the copies have been altered. In larger systems, it is common to create entirely new IPL scenarios by creating a new LOADxx member, which points to a new IEASYSxx member, which points to one or more new PARMLIB members. This process ensures that you can always do an IPL with an old configuration, but it leads to a proliferation of members and a complex web of relationships. By altering existing members where possible, that complexity can be reduced. Frequent system IPLs are done rather than dynamically activating changes to the running system.

Perform these steps to configure the z/OS system so that you isolate your data from the supplied z/OS distribution volumes, start CICS, DB2 and Developer for System z, and establish TCP/IP communications:

- Initialize the USER00, SYSUT1, and PUBLIC disks and related catalog entries.
- Change system defaults to write new files to the USER00.
- Create some USER libraries that the supplied z/OS distribution does not supply.
- Enable use of USER libraries by TCPIP, shutdown commands and a few other things.
- Create new z/OS UNIX file systems for `/tmp` and `/u`.
- Customize z/OS UNIX startup to allow a single location for TCP/IP settings.
- Customize TCP/IP settings to establish network connectivity.
- Define some basic security.
- Define a new TSO logon procedure.
- Create user ids.
- Create some data sets and log streams needed by for CICS.
- Make changes to the CICSA startup procedure to fix some problems.
- Install the RSE, Job Monitor, and Lock daemon of Developer for System z.

You may also want to make some additional changes that are commonly made.

- Customize ISPF defaults and the ISPF main panel
- Change console defaults
- Streamline startup and shutdown scripts
- Create an NJE connection to existing z/OS systems

When you do an IPL of the supplied z/OS distribution system for the first time, you will see some errors during the IPL process. For example, CICS 4.1 will not start.

Logon to TSO

Use your favorite 3270 emulator software, such as IBM Personal Communications Manager (PCOMM), to connect to the non-SNA (coax) 3270 device emulator provided by the Unit Test feature.

When you connect from outside the Linux system hosting Unit Test, use the Linux TCPIP address and 3270 as port. (The actual port number is defined in the devmap.)

When using the x3270 emulator on the host Linux, you can start a session with the following command:

```
x3270 -port 3270 tso@localhost &
```

An alternate format of the x3270 command, which produces a larger screen size, is

```
x3270  -model 4 tso@localhost:3270
```

Logon to TSO using the IBMUSER account. The initial password for IBMUSER is either SYS1 or IBMUSER. You may want to make a few minor changes to your session before you start working. If you are accustomed to using TSO naming conventions in ISPF, then to ensure that you do not write out files with unexpected high-level qualifiers, issue the command

```
TSO PROFILE PREFIX(IBMUSER)
```

Initialize new disks

The supplied z/OS distribution is designed to place user data on disk SBSYS1 and z/OS UNIX data on SBUSS1, but these disks also contain important system data, so new disks must be created to hold the customizations and users' data. This will help with future migrations to new versions of supplied z/OS distribution.

Volume names change with each release of the supplied z/OS distribution, and it is possible to mount the old xxSYS1 and xxUSS1 disks on a new system using a newer installation, and then copy the files back to the newer volumes. Placing customizations and users' data on separate volumes is a somewhat cleaner approach.

Create a data set called IBMUSER.CNTL to contain JCL. Store the samples given in this guide in the IBMUSER.CNTL data set so that you can find them or reuse them as necessary. You should be aware that, because you have not customized the system yet, IBMUSER.CNTL will be placed on SBSYS1 and will be lost once you upgrade your system to a newer z/OS distribution. To preserve the content, copy the data set to a user data set once the system customization is complete.

Verify the addresses of disks PUBLIC, USER00 and SYSUT1 in your devmap. If they are not AB0, AB1 and AB2 respectively, change the following commands and JCL accordingly.

Vary these disks offline at the z/OS console:

```
V AB0,OFFLINE
V AB1,OFFLINE
V AB2,OFFLINE
```

Create and submit a member containing the following:

```
//IBMUSERA JOB (ACCT),MSGCLASS=H,NOTIFY=&SYSUID.
//*-------------------------------------------------------
//*-------------------------------------------------------
//* MOD-3: VTOC(0,1,974)  INDEX(65,0,50)
//* MOD-9: VTOC(0,1,2939) INDEX(196,0,150)
//FORMAT   EXEC PGM=ICKDSF
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
 INIT UNIT(AB0) NOVALIDATE NVFY VOLID(PUBLIC)  PURGE -
   VTOC(0,1,974) INDEX(65,0,50)
 INIT UNIT(AB1) NOVALIDATE NVFY VOLID(USER00)  PURGE -
   VTOC(0,1,2939) INDEX(196,0,150)
 INIT UNIT(AB2) NOVALIDATE NVFY VOLID(SYSUT1)  PURGE -
   VTOC(0,1,974) INDEX(65,0,50)
/*
```

You will see messages on the console that you need to respond to (console command R xx,U). This job should end with return code 0.

Now vary the volumes online.

```
V AB0,ONLINE
V AB1,ONLINE
V AB2,ONLINE
```

Once the initialization is done, you need to create some user catalogs, as shown in the following JCL:

```
//IBMUSERB JOB (ACCT),MSGCLASS=H,NOTIFY=&SYSUID.
//*-------------------------------------------------
//DEFCAT   EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
  DEFINE USERCATALOG (NAME(USERCAT.VUSER00) ICFCATALOG -
   CYLINDERS(10 5) VOLUME(USER00) )

 DEFINE USERCATALOG (NAME(USERCAT.VSYSUT1) ICFCATALOG -
   CYLINDERS(5 5) VOLUME(SYSUT1) )
/*
```

If no user catalog is used, all (cataloged) data sets will be cataloged in the master catalog, which is replaced when you upgrade your z/OS system. To avoid placing entries in the master catalog, you can create an alias for the (new) high-level qualifier of a data set that you want to catalog in the USER00 user catalog. During migration, you then only have to replicate the alias definitions so that all user catalog entries are restored.

In a following section of this document, z/OS UNIX file systems will be created on the USER00 volume with a high-level qualifier of CUST, so start by creating an alias for CUST. When creating new users, you should also create an alias for the user ID in the USER00 catalog.

```
//IBMUSERC JOB (ACCT),MSGCLASS=H,NOTIFY=&SYSUID.
//*-------------------------------------------------
//DEFALIAS   EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
 DEFINE ALIAS (NAME(CUST) RELATE(USERCAT.VUSER00))
//*
```

Ensure all new data sets are written to USER00

Finally it is necessary to create a PARMLIB member that will ensure that new data sets will be written to volume USER00, unless specified otherwise.

Create USER.PARMLIB(VATLST00). The contents of this member must be in specific columns, so copy ADCD.Z111S.PARMLIB(VATLST00) as a model.

```
VATDEF IPLUSE(PRIVATE),SYSUSE(PRIVATE)
PUBLIC,0,1,3390    ,N
USER* ,0,0,3390    ,N
```

This will ensure that new data sets are written to USER00 and that temporary data sets are written to PUBLIC.

USER.PARMLIB already exists, and the supplied z/OS distribution is already set up to read it.

Tip: Be sure that all of your PARMLIB changes are in USER.PARMLIB. The easiest way to edit existing PARMLIB members and ensure that your changes are saved in USER.PARMLIB is to use the DDLIST command from any ISPF command line, and then type PARMLIB. Place an E next to the pseudo-ddname called PARMLIB. When you edit a member from the resulting member list, all changes will be saved in the first data set in this concatenation which, by default, is USER.PARMLIB. While in the editor, you can also compare your changes to members that already exist in ADCD.Z111S.PARMLIB or SYS1.PARMLIB, with the COMPARE NEXT command.

You should also copy ADCD.Z111S.CLIST(ISPFCL) to USER.CLIST(ISPFCL) and change the first line from

```
PROC 0 VOL(SBSYS1)
```

to

```
PROC 0 VOL(USER00)
```

To reduce accidental updates to the master catalog, you might also want to set the TSO prefix for all users when they log on by adding `PREFIX(&SYSUID.)` to the line that begins with `PROFILE` (Line 3).

```
PROFILE NOMODE MSGID PROMPT INTERCOM WTPMSG PREFIX(&SYSUID.)
```

You must do an IPL of the system to pick up the VATLST00 changes. See the next section for instructions.

Enable use of USER.PROCLIB

Create USER.PARMLIB(MSTJCL00) based on ADCD.Z111S.PARMLIB(MSTJCL00) and include USER.PROCLIB in the IEFPDSI DD name:

```
//MSTJCL00 JOB MSGLEVEL=(1,1),TIME=1440
//         EXEC PGM=IEEMB860,DPRTY=(15,15)
//STCINRDR DD SYSOUT=(A,INTRDR)
//TSOINRDR DD SYSOUT=(A,INTRDR)
//IEFPDSI  DD DSN=USER.PROCLIB,DISP=SHR       << Modified
//         DD DSN=ADCD.&UNIXVER..PROCLIB,DISP=SHR  << Modified
//         DD DSN=SYS1.PROCLIB,DISP=SHR
//SYSUADS  DD DSN=SYS1.UADS,DISP=SHR
//SYSLBC   DD DSN=SYS1.BRODCAST,DISP=SHR
```

Tip: The supplied z/OS distribution has a version number in the system data set names (ADCD.version.*). This is troublesome during upgrades, as you have to adjust all your references to these data sets. By defining the version number as a system symbol in the IEASYMxx PARMLIB member, you can use this system symbol instead for most references from within PARMLIB members and started tasks (servers).

The supplied z/OS distribution already assigns the current version number to system symbol &UNIXVER. in PARMLIB member IEASYM00. When using this variable, you only have to verify during an upgrade that the variable still exists in the new release to be sure that all references to ADCD.&UNIXVER..* are valid after the upgrade.

The next time that you do an IPL of the system, USER.PROCLIB will also be used to find startup procedures for started tasks (servers).

To perform an IPL, enter this command on the system console:

```
S SHUTDB
```

Wait for VTAM® to end and messages to stop. You will probably need to respond to messages to shut down TSO and z/OS UNIX. You can see what programs are

still running by pressing F11 on the console. (F11 in this case corresponds to the `D J,L` console command.) When only DLF and JES are running, switch to a Linux console and enter `awsstop` to stop Unit Test. Then restart Unit Test.

Create new HFS files for /tmp and /u

Management of z/OS UNIX file systems in z/OS is a complex area. This procedure provides a simplistic design for new file systems. Considerations such as space requirements, alternate mount points, and so forth may require a more in-depth plan.

The supplied z/OS distribution provides fairly small file systems for the `/tmp` and `/u` directories. This can cause problems, particularly when installing software such as Developer for System z, or when programs create large dumps on the z/OS UNIX file system. Note that the startup script described in "Starting the Unit Test environment" on page 5 already issues the `awsstop` command before restarting Unit Test. These file systems can be replaced. Create two new HFS files using ISPF option 3.2.

```
CUST.HFS.U
CUST.HFS.TMP
```

You may want to change the space parameters, but the ISPF 3.2 screen should be something like the following:

```
------------------------------------------------------------------------------
                           Allocate New Data Set
Command ===>

Data Set Name  . . . : CUST.HFS.TMP

Management class . . .                (Blank for default management class)
Storage class  . . . .                (Blank for default storage class)
 Volume serial . . . .                (Blank for system default volume) **
 Device type . . . . .                (Generic unit or device address) **
Data class . . . . . .                (Blank for default data class)
 Space units . . . . . CYLINDER       (BLKS, TRKS, CYLS, KB, MB, BYTES
                                       or RECORDS)
 Average record unit                  (M, K, or U)
 Primary quantity  . . 100            (In above units)
 Secondary quantity    20             (In above units)
 Directory blocks  . . 0              (Zero for sequential data set) *
 Record format . . . . U
 Record length . . . . 0
 Block size  . . . . . 0
 Data set name type    HFS            (LIBRARY, HFS, PDS, LARGE, BASIC, *
                                       EXTREQ, EXTPREF or blank)
 Extended Attributes                  (NO, OPT or blank)
 Expiration date . . .                (YY/MM/DD, YYYY/MM/DD
Enter "/" to select option             YY.DDD, YYYY.DDD in Julian form
   Allocate Multiple Volumes           DDDD for retention period in days
                                       or blank)
```

If your changes to VATLST00 above worked correctly, and you left the Volume serial field blank, the data sets should be created on volume USER00.

After you have created CUST.HFS.TMP and CUST.HFS.U, change PARMLIB member BPXPRMDB as follows.

Comment out the existing mount of HFS.&SYSNAME..TMP to '/&SYSNAME./tmp', and replace it with a mount of CUST.HFS.TMP

```
/* MOUNT    FILESYSTEM('HFS.&SYSNAME..TMP') */
/*          TYPE(HFS)                       */
/*         MODE(RDWR) NOAUTOMOVE            */
/*         MOUNTPOINT('/&SYSNAME./tmp')     */

MOUNT    FILESYSTEM('CUST.HFS.TMP')
         TYPE(HFS)
         MODE(RDWR) NOAUTOMOVE
         MOUNTPOINT('/&SYSNAME./tmp')
```

Make a similar change for the mount of /u:

```
/* MOUNT    FILESYSTEM('HFS.USERS') */
/*         TYPE(HFS)                */
/*         MODE(RDWR)               */
/*          MOUNTPOINT('/u')        */

MOUNT    FILESYSTEM('CUST.HFS.U')
         TYPE(HFS)
         MODE(RDWR)
         MOUNTPOINT('/u')
```

Finally, you need to copy the existing /u directories to contain the new /u file system, and ensure that everyone has proper access permission to the new file systems. The z/OS UNIX command line can be accessed with the TSO OMVS command, and then you can enter the following commands. (Be sure to verify that each command works correctly before proceeding to the next command.) [1]

```
cd /
mkdir /tempmnt
/usr/sbin/mount -f cust.hfs.u /tempmnt
/samples/copytree /u /tempmnt
ls /tempmnt                 # to verify that the copy worked
chmod 777 /tempmnt
/usr/sbin/unmount /tempmnt
/usr/sbin/mount –f cust.hfs.tmp /tempmnt
chmod 777 /tempmnt
/usr/sbin/unmount /tempmnt
rm -r /tempmnt
```

At this point, you should do an IPL of the system to ensure that the new file systems are used and that your user directory changes are working. Once the system restarts, you can enter the z/OS UNIX command prompt and issue the `df -k` command to ensure that your file systems are mounted correctly.

Setting up TCP/IP

z/OS running on Unit Test can communicate with your network via TCP/IP. This allows you to use standard 3270 terminal emulators, FTP, Developer for System z, and other services to move data to and from your z/OS system.

TCP/IP and LAN configuration is very site-dependent. The exact steps outlined here may not work at your site because of local network configuration, firewalls, Linux dependencies, or hardware restrictions. You may need the services of a network administrator to get communications to work with your network.

Set up Linux routing

Since mainframes are generally confined to data centers, TCP/IP on z/OS does not act as a DHCP client. It does not automatically configure itself to a TCP/IP address

1. At this point, /u has a mounted file system /u/db9g and that will be copied, too. Upon IPL, the original contents will be mounted at that mount point. You can delete the contents of /tempmnt/db9g before the IPL if you want to.

supplied by the network. Therefore it is necessary to configure a few settings to get TCP/IP to communicate with the network. Several methods of configuring TCP/IP are described in the PDT Redbook.

This guide shows an example of setting up the method referred to as scenario 4 in the book. This method allows the z/OS system to communicate with your network and allows your Linux machine to connect to the z/OS virtual machine as well.

Before configuring TCP/IP, you should obtain a static IP address for z/OS. The z/OS IP address must be within the same subnet as your Linux machine. It does not matter if the Linux machine has a DHCP or static IP address as long as both the z/OS and Linux address are in the same subnet. The z/OS environment will be configured to use both the static IP address you obtained and an address of 10.1.1.2. The address of 10.1.1.2 is used to communicate with the Linux machine and cannot be seen by other machines on your network. The following examples show how to configure z/OS so that the external network connects to the machine by address 9.12.200.20, and Linux connects to z/OS by address 10.1.1.2. z/OS can connect to the Linux machine using the address 10.1.1.1.

Modify TCPPARMS files

The supplied z/OS distribution supplies several USER.* libraries. However, there is no PDS to contain user modifications to TCP/IP parameters. You can create a PDS on volume SYSUT1 called USER.TCPPARMS, modeled after ADCD.Z111S.TCPPARMS.

PROFILE.TCPIP

Copy ADCD.Z111S.TCPPARMS(PROF2) to USER.TCPPARMS(PROFILE). Note the member name change. The name is being changed simply to clarify its use.

Modify the HOME, ROUTE and ROUTE DEFAULT lines to include addresses and netmask information that is correct for your network. For example, given a z/OS IP address of 9.12.200.20, and a netmask of 255.255.255.0, a TCP/IP PROFILE member might look like the following example. Note that comments and defined PORTs were removed for brevity. Note also that the IP address of the Linux system is not needed here. The gateway address usually ends in .1 or .0, but that may be different on your network. This example also includes the definitions for the 10.1.1.2 address, which is used when you want to communicate with z/OS from the Linux machine.

```
ARPAGE 5
DATASETPREFIX TCPIP
AUTOLOG 5
    FTPD JOBNAME FTPD1   ; FTP Server
    PORTMAP              ; Portmap Server
ENDAUTOLOG
PORT
     7 UDP MISCSERV            ; Miscellaneous Server
     7 TCP MISCSERV
     9 UDP MISCSERV
 ((( additional ports removed for brevity )))
SACONFIG DISABLED
  DEVICE PORTA MPCIPA
  LINK ETH1  IPAQENET PORTA
  HOME 10.1.1.2 ETH1

  DEVICE PORTB MPCIPA
  LINK ETH2  IPAQENET PORTB
  HOME 9.12.200.20 ETH2

 BEGINROUTES
```

```
 ROUTE 10.0.0.0       255.0.0.0       =         ETH1   MTU 1492
 ROUTE 9.12.200.0     255.255.255.0   =         ETH2   MTU 1492
 ROUTE DEFAULT        9.12.200.1                ETH2   MTU 1492
 ENDROUTES
ITRACE OFF
IPCONFIG NODATAGRAMFWD
UDPCONFIG RESTRICTLOWPORTS
TCPCONFIG RESTRICTLOWPORTS
START PORTA
START PORTB
```

Copy ADCD.Z111S.VTAMLST(OSATRL2) to USER.VTAMLST(OSATRL2) and remove any comments within it so that it looks like the next example.

The device name in the TCP/IP profile member must match the port names that are specified in USER.VTAMLST(OSATRL2). In this example, these are PORTA and PORTB. Also, verify that your devmap (see "Define the device map" on page 4) correctly defines the device addresses in the READ, WRITE, and DATAPATH statements of USER.VTAMLST(OSATRL2).

```
OSATRL1 VBUILD TYPE=TRL
OSATRL1E TRLE LNCTL=MPC,READ=(0400),WRITE=(0401),DATAPATH=(0402),      X
               PORTNAME=PORTA,                                         X
               MPCLEVEL=QDIO
OSATRL2E TRLE LNCTL=MPC,READ=(0404),WRITE=(0405),DATAPATH=(0406),      X
               PORTNAME=PORTB,                                         X
               MPCLEVEL=QDIO
```

To activate this configuration, copy ADCD.Z111S.VTAMLST(ATCCON00) to USER.VTAMLST(ATCCON00) and change the word OSATRL1 to OSATRL2.

TCPIP.DATA

Copy ADCD.Z111S.TCPPARMS(TCPDATA) to USER.TCPPARMS(TCPDATA) and set the HOSTNAME, DOMAINORIGIN and NSINTERADDR values. A sample, without comments, looks like this:

```
TCPIPJOBNAME TCPIP
HOSTNAME RDZUT0
DOMAINORIGIN  RTP.IBM.COM
DATASETPREFIX TCPIP
 NSINTERADDR  9.0.0.1
 NSINTERADDR  9.0.0.11
RESOLVEVIA UDP
LOOKUP DNS LOCAL
RESOLVERTIMEOUT 10
RESOLVERUDPRETRIES 1
ALWAYSWTO NO
```

If you choose a HOSTNAME or DOMAINORIGIN arbitrarily, be sure that the DOMAINORIGIN is not a real domain name or that the combination of the HOSTNAME and DOMAINORIGIN does not constitute an existing DNS name. Use the Linux `ping` or `nslookup` commands to ensure that your choice of names is not found by your DNS server. Identifying your computer as another computer, or as a member of an existing but incorrect domain, may cause problems that are unusual and difficult to diagnose, like timeouts, pauses, and connection failures in many areas, including 3270 connections and Developer for System z. Some systems, including components of Developer for System z, require that z/OS can locate itself by name.

If you cannot use a Domain Name Server (DNS) to resolve IP addresses of other systems or of the z/OS system, you can create a local hosts file and refer to it with a GLOBALIPNODES statement. The setup of this file is described in detail in the Redbook *TCP/IP implementation volume 1: Base functions* (SG24-7798). You may also need to change the LOOKUP statement in the TCPDATA member to `LOOKUP LOCAL DNS` to force z/OS to look in the local hosts file before calling DNS services.

TN3270

Copy ADCD.Z111S.TCPPARMS(TN3270) to USER.TCPPARMS(TN3270). No changes are needed to this member. It is copied solely for consistency because the next thing to do is change the procedures which refer to TCP/IP configuration files to point to the USER.TCPPARMS data set.

Modify TCP/IP Procedures to point to USER.TCPPARMS

Tip: An easy way to find Procedures that reference the TCPPARMS data sets using ISPF is to display a member list of ADCD.Z111S.PROCLIB, and then type the commands:

```
SRCHFOR TCPPARMS
SORT PROMPT
```

Tip: You can ensure that changes are saved in USER.PROCLIB using the same technique previously shown for PARMLIBs. Allocate a DDNAME to the PROCLIB concatenation found in the MSTJCLxx member, issue `DDLIST` and use the `E` line command with your DDNAME. For example, issue the TSO command:

```
TSO ALLOC F(APROCLIB)
  SHR DA('USER.PROCLIB' 'ADCD.Z111S.PROCLIB' 'SYS1.PROCLIB') REUSE
```

and use `DDLIST` to edit DDNAME APROCLIB. This same technique can be used for CLIST, TCPPARMS, and other libraries.

Copy the following members from ADCD.Z111S.PROCLIB to USER.PROCLIB. Be careful not to replace any members you may have already changed in USER.PROCLIB.

```
FTPD
PORTMAP
TCPIP
TN3270
```

If you will be configuring and using NFS, copy NFSC and NFSS also. NFS setup is not described in this document.

Modify each member to change references from ADCD.Z111S.TCPPARMS to USER.TCPPARMS for any members that you have duplicated in USER.TCPPARMS. Do not change the member names except in the TCPIP procedure as noted below.

For example, change the line in FTPD from

```
//SYSTCPD DD DISP=SHR,DSN=ADCD.Z111S.TCPPARMS(TCPDATA)
```

to

```
//*SYSTCPD DD DISP=SHR,DSN=ADCD.Z111S.TCPPARMS(TCPDATA)
//SYSTCPD DD DISP=SHR,DSN=USER.TCPPARMS(TCPDATA)
```

Remember that the name of the TCP/IP profile member name was changed from PROF1 to PROFILE, so the PROFILE DD statement should be

```
//PROFILE  DD DISP=SHR,DSN=USER.TCPPARMS(PROFILE)
```

Also note that some procedures reference ADCD.Z111S.VTAMLIB. This reference should be kept during upgrades. Replace the distribution version number by the system symbol as described in "Enable use of USER.PROCLIB" on page 11.

```
//STEPLIB  DD DISP=SHR,DSN=ADCD.&UNIXVER..VTAMLIB
```

Create a RESOLVER procedure

There are several changes that need to be made in z/OS to support TCP/IP communication. By default, z/OS TCP/IP uses different configuration files and search orders for the z/OS UNIX and MVS™ programs. The first thing to do is have both environments use the same (MVS) configuration.

To provide a common search path for MVS and z/OS UNIX to find TCP/IP configuration data, and to specify the location of the configuration data, create member USER.TCPPARMS(RESOLVER), containing these two statements:

```
GLOBALTCPIPDATA('USER.TCPPARMS(TCPDATA)')
COMMONSEARCH
```

Next, create a resolver procedure by creating USER.PROCLIB(RESOLVER), containing the following JCL:

```
//*
//* TCPIP RESOLVER
//*
//RESOLVER PROC PARMS='CTRACE(CTIRES00)'
//*
//EZBREINI EXEC PGM=EZBREINI,REGION=0M,TIME=1440,
//            PARM=&PARMS
//SETUP    DD DISP=SHR,DSN=USER.TCPPARMS(RESOLVER),FREE=CLOSE
//*
```

Once that is created, change USER.PARMLIB(BPXPRMDB) to include the line RESOLVER_PROC(RESOLVER).

Adding the RESOLVER_PROC statement will cause the RESOLVER procedure to run when your system starts.

Do an IPL of the system to verify your changes have been made.

Adding RACF security

The supplied z/OS distribution is shipped with very few security rules defined, so you may want to add several restrictions. In addition, the RACF database exists on a system volume, which makes migration to a newer z/OS distribution cumbersome. These issues warrant a quick but major overhaul of your security setup. Most importantly, making the changes listed below establishes the ability to automatically assign z/OS UNIX UID and GID values to new users and groups. The installation of Developer for System z will take advantage of this ability.

The examples presented here will copy the RACF database to the SYSUT1 disk, upgrade the database to allow functions introduced in z/OS, and then add some security rules.

These additional rules are as follows:

- Define a new TSO logon procedure called TSOLOGON.
- Allow jobs and users to use any accounting ID.
- Activate RACF Enhanced Generic Naming to allow ** in the DATASET class.
- Allow automatic UID/GID assignment when creating new groups and users (requires AIM stage 3).
- Create a new group called RDZUSERS for new users. The group includes an automatically assigned z/OS UNIX GID. By creating a new group, the existing permission structure used by started tasks and existing subsystems can be left in place, and proper permits can be assigned to the Developer for System z clients.
- Restrict updates to SYS1, ADCD, and USER data sets to users in group SYS1.
- Restrict updates to all catalogs to users in group SYS1.
- Allow users in the RDZUSERS group to update only the USERxx user catalogs to allow them to create data sets on USER00, while protecting other data sets from deletion.
- Provide TSO users with the abilities to allow users to view any job output in SDSF and to submit jobs.
- Provide users broad access to operator commands through the OPERCMDS class. (You might want to make this more restrictive.)
- Add some missing abilities for members of the SYS1 group, such as the ability to be z/OS UNIX superusers.

Copy the RACF data base to SYSUT1

Create and submit the following JCL that copies the current RACF database to the new SYSUT1 volume. The proposed size of 20 cylinders will result in about 2% usage after completion of the steps in this guide.

```
//IBMUSERE JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1),NOTIFY=&SYSUID
//*
//* CLONE ADCD RACF DATABASE AND USE CLONE AFTER IPL
//*
//* MUST RUN WHEN THERE IS NO RACF ACTIVITY
//* UPON FAILURE, ENSURE THAT ICHRDSNT IS NOT IN USER.LINKLIB
//*
//         SET VOLSER=SYSUT1               * SHOULD BE ON NON-ADCD DISK
//         SET SIZE=20
//*
//* LOGICAL COPY OF RACF DATABASE
//*
//COPY     EXEC PGM=IRRUT400,PARM='LOCKINPUT,FREESPACE(20)'
//SYSPRINT DD SYSOUT=*
//INDD1    DD DISP=SHR,DSN=SYS1.RACFDS
//OUTDD1   DD DISP=(NEW,CATLG),DSN=USER.RACF,       * MUST BE IN MCAT
//            UNIT=SYSALLDA,VOL=SER=&VOLSER,
//            SPACE=(CYL,(&SIZE),,CONTIG),DCB=DSORG=PSU
//*
//* RE-ALLOW UPDATES TO ADCD RACF DATABASE (FOR BACKOUT PURPOSES)
//*
//UNLOCK   EXEC PGM=IRRUT400,PARM='UNLOCKINPUT',COND=EVEN
//SYSPRINT DD SYSOUT=*
//INDD1    DD DISP=SHR,DSN=SYS1.RACFDS
//*
//* CREATE A RACF DATA BASE NAME TABLE (ICHRDSNT)
//* BASED UPON SYS1.SAMPLIB(RACTABLE)
//*
```

```
//TABLE    EXEC HLASMCL,COND=(0,NE)
//C.SYSIN  DD *
ICHRDSNT CSECT
         DC    AL1(1)             INDICATES ONE RACF DATA SET
         DC    CL44'USER.RACF'    PRIMARY RACF DS NAME
         DC    CL44''             BACKUP RACF DS NAME
         DC    AL1(255)           NUMBER OF RESIDENT DATA BLOCKS
         DC    X'00'              NO UPDATES DUPLICATED ON BACKUP DS
         END
//L.SYSLMOD DD DISP=SHR,DSN=USER.LINKLIB
//L.SYSIN   DD *
      NAME ICHRDSNT(R)
//*
```

After you submit this job and ensure that it ran successfully, shut down and then do an IPL of the system to force the system to use the new RACF database.

Upgrade the RACF data base to AIM3

Create and submit the following job to enable z/OS UNIX functions such as automatic GID and UID generation.

```
//IBMUSERF  JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1),NOTIFY=&SYSUID
//*
//* CONVERT USS INFO IN RACF DATABASE FROM OS/390 TO Z/OS FORMAT (AIM)
//*
//STAGE1   EXEC PGM=IRRIRA00,PARM=STAGE(1)
//SYSPRINT DD SYSOUT=*
//STAGE2   EXEC PGM=IRRIRA00,PARM=STAGE(2)
//SYSPRINT DD SYSOUT=*
//STAGE3   EXEC PGM=IRRIRA00,PARM=STAGE(3)
//SYSPRINT DD SYSOUT=*
//*
```

Establish new RACF security rules

The following job contains the series of RACF commands that establish the rules mentioned previously in this document. This is not a comprehensive security plan. Because the Unit Test feature is intended to be used as a unit test platform, these rules allow regular users extensive abilities that would not be available in a larger development, test, or production system.

In addition, with the exception of changes to the SYS1 group, no alterations are made to existing distribution user IDs such as ADCDA through ADCDZ. You may want to set new passwords, revoke or even delete some of the existing distribution user IDs.

Once these new RACF rules are in place, new TSO user IDs should be created in group RDZUSERS.

The rules here do not protect data sets owned by existing users such as IBMUSER, but you can protect them after running this job by the using ADDSD and PERMIT commands similar to the following:

```
ADDSD 'IBMUSER.**' UACC(NONE)
PERMIT 'IBMUSER.**' CLASS(DATASET) ACCESS(ALTER) ID(SYS1)
```

Review the rules in this job, submit it, and review the output using SDSF (option M.5 from the ISPF primary menu). Note that comments starting with /* must not start in column 1.

```
//IBMUSERG  JOB MSGLEVEL=(1,1),MSGCLASS=A,CLASS=A,NOTIFY=&SYSUID
//*
//* COMMANDS FOR BASIC SECURITY SETUP
//*
```

```
//CMD      EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 /* define logon procedure TSOLOGON, allow everyone to use it   */
RDEFINE TSOPROC TSOLOGON UACC(READ)
SETROPTS RACLIST(TSOPROC) REFRESH
 /* allow any acounting id                                      */
RDEFINE ACCTNUM ** UACC(READ)
RALTER ACCTNUM ACCT# UACC(READ)

 /* activate Enhanced Generic Naming (allow ** in DATASET class)*/
SETROPTS EGN

 /* allow automatic uid/gid assigment (requires AIM stage 3)    */
RDEFINE FACILITY BPX.NEXT.USER APPLDATA('5000/500')
SETROPTS RACLIST(FACILITY) REFRESH

RDEFINE UNIXPRIV SHARED.IDS UACC(NONE)
SETROPTS CLASSACT(UNIXPRIV) RACLIST(UNIXPRIV)

 /* create default group for new users                          */
ADDGROUP RDZUSERS OMVS(AUTOGID)

 /* define catalog protection                                   */
ADDGROUP CATALOG OWNER(IBMUSER) SUPGROUP(SYS1) DATA('HLQ STUB')
ADDSD 'CATALOG.**' UACC(READ)
PERMIT 'CATALOG.**' CLASS(DATASET) ACCESS(ALTER) ID(SYS1)

ADDGROUP USERCAT OWNER(IBMUSER) SUPGROUP(SYS1) DATA('HLQ STUB')
ADDSD 'USERCAT.**' UACC(READ)
PERMIT 'USERCAT.**' CLASS(DATASET) ACCESS(ALTER) ID(SYS1)
ADDSD 'USERCAT.VUSER*' UACC(UPDATE)
PERMIT 'USERCAT.VUSER*' CLASS(DATASET) ACCESS(ALTER) ID(SYS1)

 /* protect system data sets - allow group SYS1 alter           */
ADDSD 'SYS1.**' UACC(READ)
PERMIT 'SYS1.**' CLASS(DATASET) ACCESS(ALTER) ID(SYS1)

ADDGROUP ADCD OWNER(IBMUSER) SUPGROUP(SYS1) DATA('HLQ STUB')
ADDSD 'ADCD.**' UACC(READ)
PERMIT 'ADCD.**' CLASS(DATASET) ACCESS(ALTER) ID(SYS1)

ADDGROUP USER OWNER(IBMUSER) SUPGROUP(SYS1) DATA('HLQ STUB')
ADDSD 'USER.**' UACC(READ)
PERMIT 'USER.**' CLASS(DATASET) ACCESS(ALTER) ID(SYS1)

SETROPTS GENERIC(DATASET) REFRESH

 /* open up common utilities                                    */
RDEFINE SDSF ** UACC(READ)
SETROPTS CLASSACT(SDSF)

RALTER TSOAUTH ACCT UACC(READ)
RALTER TSOAUTH JCL UACC(READ)
RALTER TSOAUTH OPER UACC(READ)
SETROPTS CLASSACT(TSOAUTH)

 /* define profiles for special utilities                       */
 /* UACC(READ) : everyone can do display commands               */
 /* PERMIT UPDATE : allow any operator command to the user/group*/
RDEFINE OPERCMDS ** UACC(READ)
PERMIT ** CLASS(OPERCMDS) ACCESS(ALTER) ID(SYS1)
PERMIT ** CLASS(OPERCMDS) ACCESS(UPDATE) ID(OMVSGRP)
PERMIT ** CLASS(OPERCMDS) ACCESS(UPDATE) ID(RDZUSERS)
SETROPTS CLASSACT(OPERCMDS) RACLIST(OPERCMDS)

 /* give group SYS1 missing sysprog authorities                 */
```

```
PERMIT BPX.SUPERUSER CLASS(FACILITY) ACCESS(READ) ID(SYS1)
PERMIT BPX.FILEATTR.PROGCTL CLASS(FACILITY) ACCESS(READ) ID(SYS1)
PERMIT BPX.FILEATTR.APF CLASS(FACILITY) ACCESS(READ) ID(SYS1)
SETROPTS RACLIST(FACILITY) REFRESH

//*
```

In addition to these protections, you may want to revoke most of the user IDs that are distributed with the original distribution. Users can be listed with this command:

```
TSO SEARCH CLASS(USER)
```

You can determine which users have TSO or OMVS segments with this command:

```
TSO LISTUSER * NORACF OMVS TSO
```

Be careful not to revoke IDs that are used by started tasks. You can use the SDSF DA view to see all active started tasks. (SDSF is option M.5 on the primary ISPF menu.) For administrator IDs, you might just want to change the passwords.

Configuring CICS 4.1

CICS 4.1 automatically starts when you start the system with LOADDC (or LOADDB). There are several optional configuration changes you may want to make for CICS.

(Optional) Allow CICS commands to be entered from the console

The basic CICS 4.1 installation allows users to enter CICS 4.1 MODIFY commands in SDSF, but does not provide the ability to enter those commands from shutdown scripts or the MVS console.

To allow CICS commands to be entered from the MVS console and from shutdown scripts, you can change the definition for terminal L700 to automatically install. To do this, log on to CICS (`L CICS` from the 3270 logon screen), clear the screen, and enter the command:

```
CEDA ALTER G(USERCONS) TE(L700) AUTINSTM(Y) CONSNAME(L700)
```

Once this is complete, you may want to alter the shutdown scripts (which are described in “Altering system startup and shutdown scripts” on page 28) to close CICS using the MODIFY command instead of a CANCEL command. An example of the command used to shut down CICS with MODIFY is:

```
F CICSA,CEMT P SHUT IMM
```

(Optional) Enable CICS Management Interface

If you plan on using the CICS Explorer™ within Developer for System z to view or modify CICS resources, you can define connections to the CICS management client interface (CMCI). Once these steps are completed, a Developer for System z user can create a CMCI connection to port 1490 using the Connections selection in the Preferences. To isolate your changes from the distributed volumes, copy the contents of DFH410.SYSIN to a new data set named USER.DFH410.SYSIN. Copy ADCD.Z111S.PROCLIB(CICSA) to a USER.PROCLIB(CICSA).

1. In member USER.DFH410.SYSIN(DFH$SIP1), add the following line somewhere before the .END statement:

   ```
   TCPIP=YES,
   ```

Change GRPLIST line (line 6) to:

```
GRPLIST=(XYZLIST,WULIST),
```

2. Change the CICS startup procedure, USER.PROCILIB(CICSA) :

   To the first //STEPLIB (which is in the CICS STEP near line 59), add:

   ```
   //        DD DSN=DFH410.CPSM.SEYUAUTH,DISP=SHR
   ```

   To // DFHRPL (also in the CICS step, near line 70), add:

   ```
   //        DD DSN=DFH410.CPSM.SEYULOAD,DISP=SHR
   ```

   Log into CICS (L CICSTS41 from the VTAM screen) and enter these CEDA commands:

   ```
   CEDA ADD LIST(WULIST) G(DFHWU)
   CEDA ADD LIST(WULIST) G(DFH$WU)
   CEDA INSTALL   LIST(WULIST)
   ```

   Cancel CICS and restart from the MVS console or SDSF.

   ```
   C CICS41
   S CICS41
   ```

3. Because the SYSIN data set no longer has the same name as the CICS runtime data sets, change the allocation for the SYSIN data set (near line 52) from

   ```
   // DSN=&INDEX1..SYSIN(DFH$SIP&SIP)
   ```

   to

   ```
   // DSN=USER.DFH410.SYSIN(DFH$SIP&SIP)
   ```

Log into CICS (L CICS from the VTAM screen) and enter these CEDA commands:

```
CEDA ADD LIST(WULIST) G(DFHWU)
CEDA ADD LIST(WULIST) G(DFH$WU)
CEDA INSTALL   LIST(WULIST)
```

Cancel CICS and restart from the MVS console or SDSF.

```
C CICSA
S CICSA
```

Installing Rational Developer for System z host components

The supplied z/OS distribution does not ship with Rational Developer for System z host components installed. Installation materials are available with your copy of Developer for System z, and installation and customizing instructions are available in the Program Directory and the Host Configuration Guide.

This section provides some tips for installing Developer for System z into the z/OS system you have customized with this guide. Refer to this section as you perform the tasks in the Program Directory and the Host Configuration Guide.

All installation materials and installed files will be placed on the SYSUT1 volume. Developer for System z will be installed into its own CSI.

Use the high-level qualifier RDZUT

For the purposes of this guide, the high-level qualifier of RDZUT is used for Developer for System z installation files. Create an alias to the SYSUT1 user catalog for the RDZUT qualifier. (RACF security for the data sets is explained later in this document.) Enter the TSO command:

```
DEFINE ALIAS (NAME('RDZUT') RELATE('USERCAT.VSYSUT1'))
```

Move binary installation files to the SYSUT1 volume

As described in the program directory, the easiest way to move the installation materials to your z/OS system is through FTP.

With the supplied z/OS distribution, it is not necessary to preallocate the files that will contain your uploads. Assuming that your installation materials are in a location on a Windows based workstation called D:\HHOP800, the following commands, issued from a Windows command prompt, will copy the installation materials to the SYSUT1 volume, using RDZUT as a high-level qualifier:

```
ftp <address-of-your-z/OS-system>
<when prompted for userid enter IBMUSER>
<when prompted for a password, enter the IBMUSER password>
quote site lrecl=80 recfm=fb vol=SYSUT1 track pri=1500 sec=300
lcd D:\HHOP800
cd 'RDZUT.'
prompt off
bin
mput *.bin
mput *.smpmcs
quit
```

Receive files to SYSUT1

When you use the RECEIVE command, as described in the program directory, you should respond to the prompts using a high-level qualifier of RDZUT, and you should add VOL(SYSUT1). For example:

```
RECEIVE INDA('RDZUT.IBM.HHOP800.F1.BIN')
 INMR901I Dataset IBM.HHOP800.F1 from IBM## on IBM###
 INMR154I The incoming data set is a 'DATA LIBRARY'.
 INMR906A Enter restore parameters or 'DELETE' or 'END' +
DA('RDZUT.IBM.HHOP800.F1') VOL(SYSUT1)
```

This process can be automated with a small REXX program:

```
/* REXX  - place receive HHOP800 files on SYSUT1 */
/*         using high level qualifier RDZUT      */

Do queued();  Pull; End

Queue "RECEIVE INDA('RDZUT.IBM.HHOP800.F1.BIN')"
Queue "DA('RDZUT.IBM.HHOP800.F1' ) VOL(SYSUT1)  SYSOUT(X)"
Queue "RECEIVE INDA('RDZUT.IBM.HHOP800.F2.BIN')"
Queue "DA('RDZUT.IBM.HHOP800.F2' ) VOL(SYSUT1)  SYSOUT(X)"
Queue "RECEIVE INDA('RDZUT.IBM.HHOP800.F3.BIN')"
Queue "DA('RDZUT.IBM.HHOP800.F3' ) VOL(SYSUT1)  SYSOUT(X)"
Queue "RECEIVE INDA('RDZUT.IBM.HHOP800.F4.BIN')"
Queue "DA('RDZUT.IBM.HHOP800.F4' ) VOL(SYSUT1)  SYSOUT(X)"
Queue "RECEIVE INDA('RDZUT.IBM.HHOP800.F5.BIN')"
Queue "DA('RDZUT.IBM.HHOP800.F5' ) VOL(SYSUT1)  SYSOUT(X)"
Queue "RECEIVE INDA('RDZUT.IBM.HHOP800.F6.BIN')"
Queue "DA('RDZUT.IBM.HHOP800.F6' ) VOL(SYSUT1)  SYSOUT(X)"
Queue "RECEIVE INDA('RDZUT.IBM.HHOP800.F7.BIN')"
Queue "DA('RDZUT.IBM.HHOP800.F7' ) VOL(SYSUT1)  SYSOUT(X)"
Queue "RECEIVE INDA('RDZUT.IBM.HHOP800.JCL.BIN')"
Queue "DA('RDZUT.IBM.HHOP800.JCL') VOL(SYSUT1)  SYSOUT(X)"
```

When you have processed all of the BIN files, you can delete them. Volume SYSUT1 should contain the following files:

```
RDZUT.IBM.HHOP800.F1
RDZUT.IBM.HHOP800.F2
RDZUT.IBM.HHOP800.F3
RDZUT.IBM.HHOP800.F4
RDZUT.IBM.HHOP800.F5
RDZUT.IBM.HHOP800.F6
RDZUT.IBM.HHOP800.F7
RDZUT.IBM.HHOP800.JCL
RDZUT.IBM.HHOP800.SMPMCS
```

Tips for SMP/E steps

These steps install Developer for System z into its own CSI. The z/OS UNIX files are created in a new HFS file called RDZUT.OMVS.V80. When installing Developer for System z, modify and submit the following members in RDZUT.IBM.HHOP800.JCL.

```
FEK1SMPE
FEK2RCVE
FEK3ALOC
FEK4MNT
FEK5MKD
FEK6DDEF
FEK7APLY
```

Use the following commands in ISPF to edit these members, and add a valid jobcard before submitting the jobs. In job FEK4MNT, the size of the file system is increased to allow configuration and log files to be placed in the same file system.

```
FEK1SMPE:
  C ALL #csihlq RDZUT
  C ALL vvvvvv SYSUT1
  C ALL #tzone SMPTZN
  C ALL #dzone SMPDZN

FEK2RCVE:
  C ALL #globalcsi RDZUT.CSI
  C ALL #hlq       RDZUT

FEK3ALOC:
  C ALL =FEK   =RDZUT
  C ALL dddddd SYSUT1
  C ALL tttttt SYSUT1

FEK4MNT:
  C ALL =FEK         =RDZUT
  C ALL #dsn         RDZUT.OMVS.V80
  C ALL #volser      SYSUT1
  C ALL -PathPrefix- /
  C ALL #dsprefix    RDZUT
  C ALL tttttt       SYSUT1
  C ALL SYS1.SIOELMOD IOE.SIOELMOD
  C ALL (80 (150

FEK5MKD:
  C ALL #dsprefix    RDZUT
  C ALL -PathPrefix- /
  C ALL tttttt       SYSUT1
  C ALL =FEK         =RDZUT

FEK6DDEF:
  C ALL #globalcsi RDZUT.CSI
  C ALL #tzone     SMPTZN
  C ALL #dzone     SMPDZN
  C ALL FEK.       RDZUT.
```

```
  C ALL ttttt     SYSUT1
  C ALL dddddd    SYSUT1
  C ALL -PathPrefix- /

FEK7APLY:
  C ALL #globalcsi RDZUT.CSI
  C ALL #tzone     SMPTZN
```

Developer for System z customization

Important: Throughout the installation and customization procedure, be aware that all changes to SYS1.* data sets should be made to the USER.* data sets instead. For example, if the general instructions tell you to update a member in SYS1.PARMLIB, you should make the changes in USER.PARMLIB.

This section describes the minimum number of changes needed to customize Rational Developer for System z. If you do not make customization changes in addition to those listed here, your Developer for System z installation will use default values for other customizable items, such as TCP/IP ports.

Changes to BPXPRMxx

There are many possible changes you may make to the BPXPRMxx member during the installation and customization process. At a minimum, the file system containing Developer for System z needs to be mounted at IPL time, and the maximum address space size needs to be increased. Make at least the following changes to BPXPRMDB in USER.PARMLIB:

Change

```
MAXASSIZE(1073741824)
```

to

```
MAXASSIZE(2G)
```

Add a mount of the file system containing Developer for System z.

```
MOUNT    FILESYSTEM('RDZUT.OMVS.V80')
          TYPE(ZFS)
          MODE(RDWR)
          MOUNTPOINT('/usr/lpp/rdz')
```

Customizing the FEKSETUP job

The beginning of Developer for System z customization involves submitting a job called FEKSETUP. Customize this job as shown below. Do not forget to uncomment the two lines containing the VOLSER keyword. It occurs in the block of SET statements and in the COPY procedure.

The FEKSETUP job must be customized to place all /etc/ configuration files, /var/ project definitions, and var/log files within the RDZUT.OMVS.V80 file system.

```
//         SET HLQ=RDZUT
//         SET CUST=RDZUT.#CUST
//         SET DISP=NEW
//         SET VOLSER=SYSUT1
//         SET BASE='/usr/lpp/rdz'
//*
//* z/OS UNIX ACTIONS
//*
//USS      EXEC PGM=BPXBATCH,REGION=0M,TIME=NOLIMIT
//STDENV   DD *
BASE=/usr/lpp/rdz
```

```
CNFG=/usr/lpp/rdz/etc/rdz
SCLM=/usr/lpp/rdz/var/rdz/sclmdt
WORK=/usr/lpp/rdz/var/rdz
LOGS=/usr/lpp/rdz/var/rdz
```

FEKSETUP should return RC=0.

COMMNDxx updates

For Load parameter DC, COMMNDDC should be modified. Loadparm DB uses COMMNDDB.

Alternatively, you can add the start statements to member VTAMDB, so that the Developer for System z procedures get started slightly later in the IPL cycle, after JES2 has been initialized. This startup method is shared by DC and DB.

LPALSTxx updates

Add RDZUT.SFEKLPA to LPALSTCI.

PROGxx updates

For Load parameter DC and DB, PROGDB is used. Add the following APF authorization:

```
APF ADD
    DSNAME(RDZUT.SFEKAUTH)           VOLUME(SYSUT1)
```

Add the following LNKLST entries:

```
LNKLST ADD NAME(LNKLST00) DSN(RDZUT.SFEKAUTH) VOLUME(SYSUT1)
LNKLST ADD NAME(LNKLST00) DSN(RDZUT.SFEKLOAD) VOLUME(SYSUT1)
```

JMON Procedure update

Replace high-level qualifier FEK with RDZUT when making updates to the JMON procedure.

RSED Procedure update

Replace configuration location `/etc/rdz` with `/usr/lpp/rdz/etc/rdz` when making updates to the RSED procedure.

LOCKD Procedure update

Replace configuration location `/etc/rdz` with `/usr/lpp/rdz/etc/rdz` when making updates to the LOCKD procedure.

Ensure use of Java 6.0

In `/usr/lpp/rdz/etc/rdz/rsed.envvars`, change `JAVA_HOME` to:

```
JAVA_HOME=/usr/lpp/java/J6.0
```

ISPF Client Gateway configuration

In `/usr/lpp/rdz/etc/rdz/ISPF.conf` change the allocation for SYSPROC to:

```
sysproc=ISP.SISPCLIB,RDZUT.SFEKPROC
```

FEKRACF changes

Change all whole word instances of FEK to RDZUT. Do not modify the word FEKAPPL. Change all references to GID(*n*) to AUTOGID and change references to UID(*n*) to AUTOUID. Changes as of Developer for System z version 8.0.1. are shown below:

Change `GID(1)` to `AUTOGID`.

Change `UID(7)` to `AUTOUID`.

Change `UID(8)` to `AUTOUID`.

Change `UID(9)` to `AUTOUID`.

Set the passticket encryption key by changing:

```
SSIGNON(KEYMASKED(key16           ))
```

to

```
 SSIGNON(KEYMASKED(0123456789ABCDEF))
```

where `0123456789ABCDEF` is a random 16 digit hex string of your choice.

FEKRACF defines update permissions for people developing CARMA RAMs, managing Application Deployment Manager systems, and performing other administrative tasks. For now you can assign those to group SYS1. You can grant permission to others (using PERMIT) at a later time.

Change all occurrences of `#sysprog` to `SYS1`.

Change all occurrences of `#cicsadmin` to `SYS1`.

Change all occurrences of `#ram-developer` to `SYS1`.

Uncomment the following commands in the `RACFINIT` step:

```
SETROPTS GENERIC(CONSOLE)
SETROPTS CLASSACT(CONSOLE) RACLIST(CONSOLE)
SETROPTS GENERIC(APPL)
SETROPTS CLASSACT(APPL) RACLIST(APPL)
RDEFINE PROGRAM ** ADDMEM('SYS1.CMDLIB'//NOPADCHK) UACC(READ)
SETROPTS WHEN(PROGRAM)
```

Note: If you did not define the OPERCMDS security during the generic RACF setup in "Establish new RACF security rules" on page 19, then you must also uncomment the following:

```
SETROPTS GENERIC(OPERCMDS)
SETROPTS CLASSACT(OPERCMDS) RACLIST(OPERCMDS)
```

This guide does not include customization of Developer for System z beyond getting a connection to the Remote Systems Explorer (RSE) and Job Monitor. Customization of procedures for z/OS projects, configuration of additional components, tuning, and other setup tasks are additional tasks you may want to perform. These tasks are documented in the *Host Configuration Guide* (SC23-7658).

Optional tasks

Creating new TSO user IDs

TSO user IDs are created through a series of commands. The supplied z/OS distribution provides a group named TEST that users IDs ADCDA through ADCDZ already belong to. These user IDs do not have OMVS segments, so they cannot be used with Developer for System z unless you alter them. The example commands shown here create a user ID in the RDZUSERS group. Replace `#userid`, `#name`, and `#password` with appropriate values, and do not remove the quotes in the commands.

From a CLIST, REXX exec, or TSO command line, enter the following commands. The commands will create the user ID, provide an OMVS segment, and assign an account number, default logon procedure, and region size. They will also protect data sets with a high-level qualifier belonging to the user from being accessed by other users. Finally, the commands will create an alias in the master catalog to indicate that the user's data sets are cataloged in the user catalog on volume USER00. Be sure that the substitution in the HOME( ) and PROGRAM() parameters are in lower case.

```
ADDUSER #userid DFLTGRP(RDZUSERS) NAME('#name') PASSWORD(#password)
ALTUSER #userid OMVS(HOME(/u/#userid) PROGRAM(/bin/sh) AUTOUID)
ALTUSER #userid TSO(ACCTNUM(ACCT#) PROC(TSOLOGON) SIZE(4096))
ADDSD '#userid.**' UACC(NONE)
DEFINE ALIAS (NAME('#userid') RELATE('USERCAT.VUSER00'))
```

The new user's z/OS UNIX directory must be created. From a z/OS UNIX command line, type the following commands. You can access z/OS UNIX by typing `TSO OMVS` on the command line of any ISPF screen. You exit z/OS UNIX with the `exit` command. Again, replace `#userid` with the name of the new user ID in lower case.

```
mkdir /u/#userid
chown #userid:RDZUSERS /u/#userid
```

Define a new logon procedure

A common customization in z/OS systems is to alter the logon procedure that TSO users use. You should not alter the ISPFPROC logon procedure, because errors might prevent you from being able to fix problems later on.

Create USER.PROCLIB(TSOLOGON) based on ISPFPROC, and make modifications to TSOLOGON instead of ISPFPROC. Be sure to change the identifier on the EXEC from ISPFPROC to TSOLOGON and to override the default volser used by the ISPFCL CLIST.

```
//TSOLOGON EXEC PGM=IKJEFT01,REGION=0M,DYNAMNBR=175,
//             PARM='%ISPFCL VOL(USER00)'
```

To allow all users to use the TSOLOGON procedure, issue the following TSO commands. If you executed the RACF commands in "Establish new RACF security rules" on page 19, this is already done.

```
RDEFINE TSOPROC TSOLOGON UACC(READ)
SETROPTS RACLIST(TSOPROC) REFRESH
```

Altering system startup and shutdown scripts

When you shut down the system by typing `S SHUTDB`, or `S SHUTDOWN`, or similar commands, the system runs a series of commands through a script. You can customize the script:

1. Copy the SHUTDB procedure from ADCD.Z111S.PROCLIB to USER.PROCLIB.
2. Change the references to ADCD.Z111S.PARMLIB to point to USER.PARMLIB.
3. Copy the referenced members from ADCD.Z111S.PARMLIB to USER.PARMLIB, and make your changes to the commands there.

Here are some changes you probably will want to make.

To allow DLF to stop without errors, change

```
MODIFY DLF,MODE=NORMAL
```

to

```
MODIFY DLF,MODE=Q
```

To allow ZFS to stop without an operator prompt, change

```
F OMVS,STOPPFS=ZFS
```

to

```
F OMVS,SHUTDOWN
```

To allow CICS to shut down properly, change

```
C CICSA
```

to

```
F CICSA,CEMT P SHUT IMM
```

You can change the startup scripts the same way. Changes might include *not* starting particular subsystems, changing pause times, and so forth. For loadparm DC, look in members COMMNDDC and VTAMDB. The procedure that runs the VTAMDB script is also called VTAMDB.

Define JES NJE connectivity

The supplied z/OS distribution is a standalone system with no connection to other z/OS systems. However, you might want to connect it to one or more of your z/OS systems to transfer data to customize and utilize the Unit Test system.

Since z/OS 1.7, JES supports NJE over TCP/IP, which makes setting up a connection between two systems an easy task. The following operator commands, to be executed on the Unit Test system, name the local system RUT0 and define a connection to M168.

```
$TNODE(N1),NAME=RUT0
$TLINE1,UNIT=TCP
$SLINE1
$ADDNETSRV1,SOCKET=LOCAL
$SNETSERV1
$TNODE2,NAME=M168
$ADDSOCKET(REMOTE),NETSRV=1,LINE=1,NODE=2,IPADDR=M168.RTP.IBM.COM
$SN,SOCKET=REMOTE
```

Similar operator commands should be given on the M168 system to complete the setup. Since this is an existing system, the command to define the local node name was skipped. The commands below also assume M168 does not have spare line or node definitions, so new ones (line 5 and node 20) are created.

```
$ADDLINE5,UNIT=TCP
$SLINE5
$ADDNETSRV1,SOCKET=LOCAL
$SNETSERV1
$TNJEDEF,NODENUM=20
$TNODE20,NAME=RUT0
$ADDSOCKET(REMOTE),NETSRV=1,LINE=5,NODE=20,IPADDR=RDZUT0.RTP.IBM.COM
$SN,SOCKET=REMOTE
```

The Unit Test system can use the existing NJE definitions on the M168 system to connect to other NJE nodes in your network. Issue the following operator commands on the Unit Test system to connect to the IPO1 system (node 3) through the previously defined M168 system (node 2).

```
$TNODE3,NAME=IPO1
$ADDCONNECT,NODEA=2,NODEB=3
```

TRANSMIT (XMIT) and RECEIVE

The TSO TRANSMIT (XMIT) and RECEIVE commands do not use the JES node name by default. In order to do so, copy ADCD.Z111S.PARMLIB(IKJTSO00) to USER.PARMLIB and update the NODESMF definition in the TRANSREC section. Change

```
TRANSREC                  /*                    */ +
   NODESMF((NODENAME,SMF)) /*                    */ +
```

to

```
TRANSREC                  /*                    */ +
   NODESMF((*,*))         /*                    */ +
```

Appendix A. z/OS distribution notes

The following information describes the contents of the z/OS distribution disks and was originally created to describe the z/OS distribution from which the supplied z/OS distribution was derived.

Customization help and location of product program directories (PGMDIRs)

A great deal of effort has gone into the z/OS distribution to customize the products. However, you may find some products that are not fully customized or are customized with options that you need to change.

Included in this release are files in the format of *p.SVSC.l*, where *p* = the product's high-level qualifier and *l* = library name. The purpose of these files is to provide PGMDIRs, README, and INSTALL instructions. This z/OS distribution is built from a driver system that uses different volume and library names. Therefore, you will notice that the names referenced in the SVSC files are different than in the z/OS distribution. You may need to make translations on names to implement in the z/OS distribution. Included on the SBRES1 volume are data sets with high-level qualifiers of MVS.ZOSRxx that document MVS. The MVS program directories are in these data sets.

Contents of DVDs

The following DVDs are contained in the z/OS 1.11 distribution:

DVD Disk 1	
File	**Description**
sbres1.gz	RES Volume 1 - Required for IPL
sbres2.gz	RES Volume 2 - Required for IPL
sbsys1.gz	System volume 1 - Required for IPL
sbuss1.gz	z/OS UNIX System Services Volume 1 - Required for IPL
sbprd1.gz	z/OS Products - Not required for basic IPL but required to run individual products
sbprd2.gz	z/OS Products - Not required for basic IPL but required to run individual products
sbprd3.gz	z/OS Products - Not required for basic IPL but required to run individual products

DVD Disk 2	
File	**Description**
sbdis1.gz	Distribution Volume 1
sbdis2.gz	Distribution Volume 2
sbdis3.gz	Distribution Volume 3
sbdis4.gz	Distribution Volume 4
sbdis5.gz	Distribution Volume 5

DVD Disk 2	
sbdis6.gz	Distribution Volume 6

DVD Disk 3	
File	**Description**
sbcic1.gz	CICS 4.1 Target, Distribution, customized data sets
sbdb91.gz	DB2 9.1 Target Libraries
sbdb92.gz	DB2 9.1 Distribution libraries, DB2 product libraries
sbdb93.gz	DB2 9.1 Databases, DB2 Catalogs, customized files

DVD Disk 4	
File	**Description**
sbwas1.gz	WAS 7.0 target libraries
sbwas2.gz	WAS 7.0 Distribution libraries

DVD Disk 5	
File	**Description**
sbbbn1.gz	z/OS System Management Facility Target Libraries
sbbbn2.gz	z/OS System Management Facility Distribution Libraries
sbwas3.gz	WAS 7.0 zFS data set

Naming conventions

- Catalog structure name changes:
  - CATALOG.Z111S.MASTER - Master Catalog found on sbsys1
  - USERCAT.Z111S.PRODS - z/OS Product Catalog found on sbres2
  - USERCAT.Z111S.CICS - CICS Catalog found on sbcic1
  - USERCAT.Z111S.IMS - IMS Catalog found on sbims1

    Note: IMS is only available via download in this release. However, IMS customization has been included in this documentation. Reference: Subsystem Downloads
  - USERCAT.Z111S.DB2V9 - DB2V9 Catalog found on sbdb91
  - USERCAT.Z111S.WAS - Websphere catalog found on sbwas1
- ADCD Library names found in this release - all of the following reside on sbres1.
  - ADCD.Z111S.PARMLIB
  - ADCD.Z111S.PROCLIB
  - ADCD.Z111S.TCPPARMS
  - ADCD.Z111S.VTAMLIB
  - ADCD.Z111S.VTAMLIST
  - ADCD.Z111S.VTAM.SOURCE
  - ADCD.DYNISPF.ISPPLIB
  - ADCD.Z111S.CLIST
  - ADCD.Z111S.DBS.ISPPLIB

- ADCD.Z111S.DBS1.ISPPLIB
- ADCD.Z111S.ISPPLIB
- ADCD.Z111S.LINKLIB
- ADCD.Z111S.LPALIB
- ADCD.Z111S.WLM

Build Structure

SBRES1 and SBRES2

SBRES1 is a 3390-3 volume that contains the base MVS system software target, or run-time libraries, and other system data sets required to do an IPL of the system and use TSO/E and ISPF. This is the MVS IPL volume.

SBRES2 is a 3390-3 volume that is an extension of the SYSRES volume SBRES1. The volume is also required for doing an IPL. SBRES2 contains the user catalog USERCAT.Z111S.PRODS, containing entries for z/OS related products.

SBSYS1

This 3390-3 volume contains data sets that you might change, either through normal use of the system or user customization. All USER.xxxxxx data sets are on this volume. System control data sets, such as SYS1.IPLPARM, the RACF database, IODF (I/O Definition File), and the system's master catalog (CATALOG.Z111S.MASTER) reside on this volume. This volume is required to do an IPL.

SBUSS1

SBUSS1 is a 3390-3 volume that contains all HFS and ZFS files for z/OS UNIX System Services. The volume contains the ROOT and Version HFS files. This volume is required to do an IPL.

SBPRD1, SBPRD2, and SBPRD3

SBPRD1 and SBPRD2 are 3390-3 volumes that contain all z/OS products that are not required to do an IPL of basic z/OS. If the products are customized at some future date, for example parmlib members are updated, then these volumes may be required for doing an IPL. SBPRD3 contains the language target libraries. SBPRD3 contains the HFS files for Java 5 and Java 6.

SBDIS1, SBDIS2, SBDIS3, SBDIS4, SBDIS5, and SBDIS6

These volumes contain the base MVS system software distribution libraries. They contain the DLIB (Distribution Library) and zone CSIs. These volumes do not need to be present to do an IPL, but they are needed to install service or products on the MVS system using SMP/E. These volumes are all 3390-3.

SBDB91, SBDB92, and SBDB93

SBDB91 is a 3390-3 volume that contains DB2 V9 related data. SBDB91 contains the DB2 user catalog, USERCAT.Z111S.DB2V9. SBDB91 also contains all DB2 target libraries. This volume is not required to do an IPL, but it is required to bring up DB2 V9.

SBDB92 is a 3390-3 volume that contains DB2 V9 DLIB and CSIs. SBDB92 also contains target and DLIB for various DB2 V9 utility products. The volume is not

required for doing an IPL, but it is needed to apply DB2 maintenance and run DB2 utilities and products (for example, QMF™, Administration tool).

SBDB93 is a 3390-3 volume that contains DB2 catalog and databases. The volume contains basic customization for bringing up DB2 on the z/OS distribution. The volume is not required for an IPL but is needed to bring up DB2.

SBCIC1

SBCIC1 is a 3390-3 volume that contains CICS Target, DLIB and all related data for CICS. USERCAT.Z111S.CICS resides on this volume. This volume is not required to do an IPL, but it is required to bring up CICS and apply CICS maintenance.

SBWAS1, SBWAS2, and SBWAS3

SBWAS1 is a 3390-3 volume that contains target libraries for Websphere Application Services.

SBWAS2 is a 3390-3 volume that contains distribution libraries for Websphere Application Services. SBWAS2 is required to perform maintenance on Websphere Application Services.

SBWAS3 is a 3390-3 volume that contains one Websphere Application Services target ZFS file that is very large. These volumes are not required to do an IPL, but SBWAS3 is required to bring up Websphere Application Services.

SBIMS1, SBIMS2, SBIMS3

SBIMS1 is a 3390-3 volume that contains IMS Target, DLIB, and all related data for IMS. USERCAT.Z111S.IMS resides on this volume. This volume is not required to do an IPL, but it is required to bring up IMS and perform maintenance on IMS.

SBIMS2 and SBIMS3 are 3390-3 volumes that contain IMS Enterprise Suite. This volume is not required to do an IPL, but it is required to bring up IMS and perform maintenance on IMS Enterprise Suite.

SBIMS is available as a separate file that can be downloaded from the IBM web site.

SBBN1 and SBBN2

SBBBN1 is a 3390-3 volume that contains target libraries for z/OS Management Facility, zosmf. This volume is not required to do an IPL, but it is required to bring up zosmf.

SBBBN2 is a 3390-3 volume that contains distributions libraries for z/OS Management Facility, zosmf. This volume is not required to do an IPL, but it is required to perform maintenance on zosmf.

SARES1

This 3390-3 volume contains a single volume stand-alone system. This volume can be used to do an IPL and logon to a TSO/ISPF session.

The disk volume can assist in building LPAR environments and correct errors that prevent system IPL.

Recommended use of this volume is to install the volume and have it accessible. The volume should not be altered. The volume should stay in a background mode and be available for emergency situations. If other z/OS, OS/390, or LPAR partitions contain errors, the SARES1 volume could be used to solve the problem.

The stand-alone system does not contain TCPIP or z/OS UNIX System Services support. The system cannot be used to install products or apply maintenance via SMP/E.

To do an IPL of the standalone res volume, do the following:

Load Address: 0Axx

Load Parms: 0AxxSA

Migration guidelines

The following guidelines will make it easier for you to replace this level of system software with new levels built the same way.

Only IBM-supplied system software should reside on SBxxxx volumes, except for SBSYS1. SBSYS1 contains RACF, IPL, and catalog data sets that are user-dependent.

The z/OS distribution does not contain a generalized migration utility or process. Each user has unique requirements. You should build a migration plan based on your unique needs. The z/OS distribution does have some assistance. The z/OS distribution uses a system of concatenated libraries (see below). The highest level of concatenation is USER.xxxxxxxx. These libraries are empty in the z/OS distribution. You should place changes or overrides in these libraries. You should back up the USER.xxxxxxxx libraries regularly. When a new z/OS distribution release is installed, the USER.xxxxxxxx libraries can be copied from backups to the new USER.xxxxxxxx libraries to provide instance migration of programs and parameters.

The z/OS distribution is configured with a standard library concatenation for the following:

- LINKLST
- CLIST
- ISPPLIB (TSO panels)
- LPALIB
- PARMLIB
- VTAMLST
- VTAMLIB
- VTAM source

PROCLIB must be manually configured by changing MSTJCLxx.

The configured order of concatenation is user, distribution developers, and z/OS system data sets. For example, the LINKLST concatenation would be as follows:

- USER.LINKLIB
- ADCD.Z111S.LINKLIB
- SYS1.LINKLIB

USER.xxxxx libraries have all been built on SBSYS1, which is the only volume that should contain user data.

USER.xxxxx will not be changed by the z/OS distribution or system processes; user updates in USER.xxxxx will be retained between release levels. ADCD.Z111S.xxxxxx libraries are allocated on SBRES1 and are the libraries that are used by distribution developers. No RACF rules exist on these libraries; however, changes to these libraries could destroy customization necessary to bring up many products. SYS1.xxxxxxx libraries should NEVER be updated by users or distribution developers. System libraries should ONLY be updated through SMP/E install, maintenance processes, or similar processes.

PLEASE NOTE: Changes to System libraries or z/OS distribution pre-customization may corrupt your system and prevent or delay IBM support activities.

The correct procedure for a user update to system data is to do the following:

- Copy the system data from SYS1.xxxxx or ADCD.xxxx to USER.xxxxx.
- Edit, compile, or run user program against the USER.xxxx item.
- Do another IPL, recycle system service, or logon to TSO again.

Removing a user update would be the reverse of the above procedure.

LOADPARMS options

Note: JES2 should be cold started the first time you bring up the system.

LoadParm is preset to 0A82CS.

Table 1. LOADPARMS options

Table Listing of available pre-configured distributed LOADPARMS	
LOADPARM	**Description**
CS	CLPA and cold start of JES2. Base z/OS system functions. No CICS, DB2, IMS, WAS, and so forth.
00	Warm start of JES2. Base z/OS system functions. No CICS, DB2, IMS, WAS, and so forth.
WS	Warm start of JES2. Base z/OS system functions. No CICS, DB2, IMS, WAS, and so forth.
DC	CLPA, brings in CICS LPA modules, cold start of JES2, starts up DB2 and CICS.
DB	Warm start of JES2 and starts the DB2 and CICS.
DI	CLPA and cold start of JES2 and loads the IMS Libraries. IMS must be manually started.
CC	CLPA and cold start of JES2, loads the CICS Libraries, starts up CICS. No DB2.
CW	Warm start of JES2 and starts up CICS.
9C	CLPA, cold start of JES2, starts up DB2 V9 only.
9W	Warm start of JES2, starts up DB2 V9 only.
IC	CLPA and cold start of JES2 and load the IMS Libraries, start IMS, no DB2 or CICS.

Table 1. LOADPARMS options (continued)

Table Listing of available pre-configured distributed LOADPARMS	
IW	Warm start of JES2 start IMS, no DB2 or CICS.
AC	CLPA and cold start of JES2 load IMS and CICS libraries, start IMS, DB/2 V9, and CICS 3.2.
AW	Warm start of JES2. Start IMS, DB/2 V9, and CICS 3.2.
BC	CLPA and cold start of JES2, load WAS libraries, WAS is manually started.
BW	Warm start of JES2. WAS is manually started.

Console PF key settings

As the system is running, you may wish to do some of the following from the Master Console:

Table 2. Console PF key settings

Current® PF Key settings	
PF Key	**Action/Command**
PF1	Display Devices
PF2	Display 3270 Devices
PF3	Clear Top of Screen
PF4	Create a 10-line Display Area
PF5	Display Address Space Information
PF6	Display Outstanding Reply Requests and Error Messages
PF7	Display PF Keys
PF8	Scroll Display Area
PF9	Display TSO Users
PF10	Display Active Address Spaces
PF11	Display Active Jobs
PF12	Clear Bottom Screen

User IDs

The following TSO Userids and Passwords have already been set up on your system:

Table 3. Table of pre-defined user IDs

Table of pre-defined user IDs	
User ID	**Password**
ADCDMST (RACF special authority)	SYS1 or ADCDMST
IBMUSER (RACF special authority)	SYS1 or IBMUSER
SYSADM (DB2 and RACF special authority)	SYS1 or SYSADM
SYSOPR (DB2 and RACF special authority)	SYS1 or SYSOPR
ADCDA - ADCDZ	TEST
WEBADM	WEBADM

Table 3. Table of pre-defined user IDs (continued)

Table of pre-defined user IDs	
OPEN1 thru OPEN3	SYS1

Maintenance service levels

Current levels

All the products on the z/OS distribution consist of maintenance that is in a closed status. A PTF that is still in open status or has other than a document hold at the time the z/OS distribution was built would not be added to the z/OS distribution. The following PUT levels may be minus PTFs that were open at build time. Also, some functions may be better than the PUT level due to the addition of RSUs and individual PTFs. In general, the following is valid:

- All functions of the base Z/OS 1.11 product are at PUT level 1003.
- All functions of the CICS TS 4.1 are at PUT level 1003.
- The functions of the DB2 9.1.0 base are at PUT level 1003.
- All functions of the IMS 11.1.0 are at PUT level 1003.
- WAS 7.0.0 is at Service Level 7.0.0.8 - Build Level CF080948.14.

Appendix B. Starting IMS

The z/OS distribution contains an installed and configured IMS DB/TM system.

Starting IMS

These instructions to start IMS require two minor changes to the configuration of the supplied z/OS distribution. The steps provided in this appendix do an IPL of the system with the IMS configuration loadparms of IC and IW. This is different than the loadparm used in the rest of this guide. IEASYSIC (and IW) references BPXPRMxx, LPALSTxx, COMMNDxx, PROGxx and other members that are different than those referenced by IEASYSDC (and DB). To be consistent with the changes made in the rest of this guide, including changes to the z/OS UNIX file system, locate the parmlib members for the IMS configuration and merge the changes made in the rest of the guide into these members. Alternatively, if you will not be using CICS 4.1, you can follow the guide, altering the parmlib members appropriate for the IC and IW loadparms.

This system was built using the IMS 'Installation Verification Program' (IVP) dialog.

- Copy `ADCD.Z111S.VTAMLST(ATCCON00)` to `USER.VTAMLST(ATCCON00)` and change the word `IMS10APL` to `IMS11APL`.
- Copy `ADCD.Z111S.VTAMLST(IMS11APL)` to `USER.VTAMLST(IMS11APL)` and change all instances of `IMS11TAB` to `IMS91TAB`.
- Start your system:
  1. The first time you do an IPL of the system to run IMS, you need to do a cold start. To do a cold start, set the loadparm value to IC:

     ```
     ipl a80 parm 0a82ic
     ```

     After you start the system once with a cold start, you can start the system with a warm start by setting the loadparm to IW:

     ```
     ipl a80 parm 0a82iw
     ```
- Start IMS:
  1. Start IRLM.

     Issue the z/OS command:

     ```
     S IMS11RL1
     ```
  2. Start the IMS control region.

     Issue the z/OS command:

     ```
     S IMS11CR1
     ```
  3. Issue the appropriate IMS start command using the outstanding IMS WTO.

     The following is an example of this WTO:

     ```
     *nn DFS810A IMS READY         10286/1207444 IMS11CR1.IMS11CR1   IVP1
     ```

     The reply is the following. (Make sure to include the dot (.) at the end of the command.)

     a. Cold start

        ```
        R nn,/NRE CHKPT 0 FORMAT ALL .
        ```

     b. Normal IMS warm start

        ```
        R nn,/NRE .
        ```

     c. Emergency IMS warm start

```
R nn,/ERE .
```

At this point, IMS is up and running.

(Optional) Running IMS sample applications

A basic set of applications has been set up. The applications are ready to run. Refer to the *IMS Installation Guide* for information on the IMS IVP and sample applications provided.

The following is an example of running the 'IVTNO' transaction:

1. Logon to the IMS User terminal:
   a. Acquire a session to your z/OS system.
   b. On this session enter `L IMS3270` –- this will bring up the IMS Sign-on screen
   c. Enter an existing user ID and its valid password. For example:
      ```
      USER ID: ADCDA
      PASSWORD: TEST
      ```
2. Obtain the input screen:
   a. Issue the /FORMAT command:
      ```
      /FOR IVTNO
      ```
      The screen that is displayed should look like the following example:
      ```
      ************************************************
      *     IMS INSTALLATION VERIFICATION PROCEDURE    *
      ************************************************


                                      TRANSACTION TYPE : NON-CONV (OSAM DB)
                                      DATE             : 10/13/2010

        PROCESS  CODE  (*1) :
                                                          (*1) PROCESS CODE
        LAST  NAME          :                                    ADD
                                                                 DELETE
        FIRST  NAME         :                                    UPDATE
                                                                 DISPLAY
        EXTENSION  NUMBER   :                                    TADD

        INTERNAL  ZIP CODE  :


                                                        SEGMENT# :
      ```
   b. Display a record by filling in the following values and pressing Enter:
      ```
      Field              Value
      -------------  -------------------
      Process Code:    DISPLAY
      LAST NAME:        Last1
      ```
   c. Information for LAST1 should be displayed on the screen.

Notices

Note: This Program is licensed only for development and test of applications that run on IBM z/OS. The Program may not be used to run production workloads of any kind, nor more robust development workloads including without limitation production module builds, pre-production testing, stress testing, or performance testing.

Programming interfaces: Intended programming interfaces allow the customer to write programs to obtain the services of Rational Developer for System z.

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

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

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

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

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

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED

WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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

Intellectual Property Dept. for Rational Software
IBM Corporation
3039 Cornwallis Road, PO Box 12195
Research Triangle Park, NC 27709
U.S.A.

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

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

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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

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

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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

Copyright license

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

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

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 2005, 2010.

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

Trademarks and service marks

See www.ibm.com/legal/copytrade.shtml.

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.

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

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

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

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

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

Index

V

W

Z

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

Developer for System z
IBM Rational Developer for System z Unit Test: Configuration Guide
Version 8.0.1

Publication No. SC14-7281-01

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

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

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

Comments:

Thank you for your support.

Submit your comments using one of these channels:

- Send your comments to the address on the reverse side of this form.
- Send a fax to the following number: 1-800-227-5088(US and Canada)

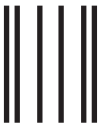
If you would like a response from IBM, please fill in the following information:

Name	Address
Company or Organization	
Phone No.	Email address

Readers' Comments — We'd Like to Hear from You
SC14-7281-01

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

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

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Information Development
Department G7IA / Bldg. 503
P.O. Box 12195
Research Triangle Park, NC
27709-2195

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line

Program Number: 5724-T07

Printed in USA

SC14-7281-01